IMPLEMENTASI *OBJECT TRACKING* PADA KAMERA PENGAWAS SEMI-OTOMATIS DENGAN *IMAGE PROCESSING* METODE *MEAN-SHIFT*

SKRIPSI

Oleh

Faiq Aprilian Romzi

NIM 141910201077

PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS JEMBER

2018

IMPLEMENTASI *OBJECT TRACKING* PADA KAMERA PENGAWAS SEMI-OTOMATIS DENGAN *IMAGE PROCESSING* METODE *MEAN-SHIFT*

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Teknik Elektro (S1)
dan mencapai gelar Sarjana Teknik

Oleh

Faiq Aprilian Romzi

NIM 141910201077

PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS JEMBER

2018

PERSEMBAHAN

Alhamdulillah, puji syukur atas kehadiran Allah SWT karena atas limpahan rahmat, karunia, hidayah, dan kasih sayang-Nya saya dapat menyelesaikan penelitian ini dengan baik. Akhirnya saya persembahkan penelitian ini kepada :

1. Allah SWT yang Maha Segalanya.
2. Junjungan kita Nabi Muhammad SAW sebagai penutan kita di segala hal
3. Orang Tua, Mas Robby, Mas Nizar, dan Mbak Leyla, terimakasih telah memberikan kesempatan, dukungan serta semangat dalam kehidupan ini.
4. Budhe Ie yang telah memberikan pelajaran moral kepada saya sehingga menjadi pribadi yang tangguh.
5. Dosen Teknik Elektro Universitas Jember yang telah memberikan ilmu yang sangat bermanfaat.
6. Seluruh saudara serta kerabat saya di Blitar yang telah mendorong saya menjadi pribadi yang lebih baik.
7. Dulur Alumni Teknik Elektronika Industri 1, terimakasih telah menyambut dengan hangat kehadiran saya di Kota Blitar dan terimakasih atas pengalaman berharga yang diberikan.
8. Dulur Ketek UJ, dulur yang memberikan pelajaran lain dalam hidup, menumbuhkan kesadaran sosial, serta terimakasih telah merangkul saya dengan cara yang berbeda.
9. Keluarga Besar UKM Robotika Teknik, terimakasih telah mendidik dan mengajarkan tentang kesetiaan, loyalitas serta tanggung jawab.
10. Anggota Komunitas Elco, terimakasih telah menyumbangkan ilmu, telah rela datang untuk meramaikan komunitas, telah menghargai keberadaan komunitas, dan terimakasih telah mendukung kegiatan-kegiatan Elco.
11. Pak Masroni, Pak Kurniawan, serta guru lain, yang telah mengenalkan saya pada dunia elektro, membimbing mendukung, dan mengapresiasi saya sehingga dapat terus berkembang.
12. Qonita Ayu Wulandari

MOTTO

***Aal izz Well* (Punchuk Wangdu)**

Permudahlah, jangan mempersulit. Gembirakanlah, jangan menakut-nakuti
(Muttafaq 'Alaih)

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Faiq Aprilian Romzi

NIM : 141910201077

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul ”Implementasi *Object Tracking* Pada Kamera Pengawas Semi-Otomatis Dengan *Image Processing* Metode *Mean-Shift*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab penuh atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 8 Juni 2018

Yang menyatakan,

Faiq Aprilian Romzi

NIM 141910201077

SKRIPSI

IMPLEMENTASI *OBJECT TRACKING* PADA KAMERA PENGAWAS SEMI-OTOMATIS DENGAN *IMAGE PROCESSING* METODE *MEAN-SHIFT*

Oleh

Faiq Aprilian Romzi

NIM 141910201077

Pembimbing :

Dosen Pembimbing Utama : Catur Suko Sarwono, ST, MSi

Dosen Pembimbing Anggota : Khairul Anam, S.T., M.T., Ph.D

PENGESAHAN

Skripsi berjudul "**Implementasi Object Tracking Pada Kamera Pengawas Semi-Otomatis Dengan Image Processing Metode Mean-Shift**" karya Faiq Aprilian Romzi telah diuji dan disahkan pada :

Hari : Jum'at

Tanggal : 8 Juni 2018

Tempat : Fakultas Teknik Universitas Jember

Ketua,	Anggota I,
Catur Suko Sarwono, ST, Msi	Khairul Anam, S.T., M.T., Ph.D
NIP 196801191997021001	NIP 197804052005011002
Anggota II,	Anggota III,
Sumardi ST., MT	Ike Fibriani, S.T., M.T.
NIP 196701131998021001	NIP 198002072015042001

Mengesahkan

Dekan,

Dr. Ir. Entin Hidayah, M.U.M.

NIP 196612151995032001

RINGKASAN

Implementasi *Object Tracking* Pada Kamera Pengawas Semi Otomatis Dengan *Image Processing* Metode *Mean-Shift*; Faiq Aprilian Romzi; 141910201077; 2018; 77 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Closed Circuit Television (CCTV) merupakan sebuah sistem tertutup yang menggunakan piranti kamera sebagai pengambil dan penyimpan gambar pada waktu tertentu dimana perangkat ini terpasang. Seiring dengan perkembangan zaman, kebutuhan akan pengawasan semakin tinggi. Pada tempat dengan pengawasan khusus sudah terpasang beberapa kamera. Namun pemasangan kamera yang *fixed* belum efektif karena jangkauan pantau masih terbatasi oleh *frame***.**

Untuk mengatasi masalah tersebut, tentunya dibutuhkan suatu metode agar jangkauan pantau lebih luas, *object tracking* merupakan solusi yang dapat dipilih. *Object tracking* merupakan proses mencari posisi objek dari sebuah video untuk setiap *frame* pada video tersebut. Object tracking merupakan proses yang dilakukan oleh *computer vision* sehingga diperlukan algoritma yang dapat menganalisis setiap perubahan *frame* dalam video dengan tujuan dapat diketahui adanya pergerakan objek pada lokasi tersebut. Dari beberapa penelitian sebelumnya, diketahui metode *mean-shift* dapat diterapkan pada penelitian ini. Metode *mean-shift* dipilih karena tingkat keakuratan penjejakan yang tinggi sehingga nilai error untuk menggerakkan kamera kecil.

Penelitian ini memiliki tujuan yaitu: (1) Merancang algoritma *image processing* dengan metode *mean-shift* sebagai penjejak objek. (2) Mengimplementasikan sistem *object tracking* pada kamera pengawas semi-otomatis. Kamera pengawas dirancang menggunakan kamera *webcam* yang terhubung dengan komputer. Personal komputer sebagai pengolah gambar mendapatkan masukan dari kamera *webcam* berupa gambar *real time*. Hasil

pengolahan gambar yang berupa data-data koordinat dikirimkan ke mikrokontrol sebagai penggerak motor DC dan motor servo.

Pengujian pada alat mencakup pengaruh variabel yang terkait dengan metode yang digunakan serta ketahanan alat saat mendapatkan gangguan. Variabel yang diuji adalah jumlah iterasi, ukuran target model dan *nilai increment area*. Setelah melakukan pengujian, dapat diketahui metode mean-shift dapat diimplementasikan pada kamera pengawas semi-otomatis dengan mempertimbangkan nilai variabel jumlah iterasi, ukuran *template* target model, dan nilai *increment area*.

PRAKATA

Puji syukur kehadirat Allah SWT yang Maha Kuasa, karena dengan ridho, hidayah dan petunjukNya, penulis dapat menyelesaikan skripsi yang berjudul "*Implementasi Obejct Tracking Pada Kamera Pengawas Semi-Otomatis Dengan Image Processing Metode Mean-Shift*". Selama penyusunan skripsi ini penulis mendapat bantuan berbagai pihak sehingga skripsi ini dapat diselesaikan.

Untuk itu penulis mengucapkan terimakasih kepada.

1. Allah SWT yang telah melimpahkan rezeki, rahmat, hidayah dan karunia serta kasih sayang-Nya sehingga penulis mampu menyelesaikan skripsi ini.
2. Nabi besar Muhammad SAW, yang telah menjadi suri tauladan seluruh umat.
3. Ibu Dr. Ir. Entin Hidayah, M.U.M., selaku Dekan Fakultas Teknik Universitas Jember
4. Bapak Dr. Bambang Sri Kaloko, S.T., M.T., Selaku Ketua Jurusan Teknik Elektro Universitas Jember;
5. Bapak Catur Suko Sarwono, ST. Msi dan Khairul Anam S.T., M.T., Ph.D. selaku dosen pembimbing yang telah membimbing menyelesaikan skripsi ini;
6. Bapak Sumardi ST., MT dan Ike Fibriani, S.T., M.T. selaku dosen penguji yang sudah memberikan saran untuk memperbaiki tugas akhir ini;
7. Keluarga besar Civitas Akademia Jurusan Teknik Elektro Universitas Jember. Serta semua pihak yang yang telah mendukung dalam penyelesaian skripsi ini.

Semoga skripsi ini dapat bermanfaat dalam mengembangkan ilmu pengetahuan khususnya untuk disiplin ilmu teknik elektro. Kritik dan saran yang mambangun diharapkan terus mengalir untuk lebih menyempurnakan skripsi ini dan dapat dikembangkan untuk penelitian selanjutnya;

Jember, 08 Juni 2018

Penulis

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR LAMPIRAN

Halaman

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Closed Circuit Television (CCTV) merupakan sistem tertutup yang menggunakan kamera sebagai pengambil dan penyimpan gambar pada waktu dan tempat tertentu dimana perangkat ini terpasang. Berbeda dengan televisi yang bersifat membagikan gambar kamera secara menyeluruh, CCTV bersifat tertutup yang digunakan untuk pengawasan. CCTV biasanya terpasang pada tempat umum atau tempat yang memerlukan pengawasan khusus seperti kantor, toko, bank, bandara, dan pusat keramaian lain.

Seiring dengan perkembangan zaman, kebutuhan akan pengawasan keamanan semakin tinggi. Bahkan pada tempat tertentu yang memerlukan pengwasan lebih, telah terpasang beberapa kamera. Namun pemasangan kamera yang *fixed* mengakibatkan kinerja pengawasan kurang optimal dikarenakan jangkauan pantau kamera terbatasi oleh luas maksimal *frame* kamera tersebut. Untuk itu perlu dikembangkan teknologi yang dapat menambah atau memaksimalkan luas *frame* kamera sehingga keamanan dapat meningkat.

Penjejakan Objek atau *Object tracking* merupakan proses mencari letak objek yang akan diamati dari sebuah data video untuk setiap *frame* pada video tersebut. Fungsi utama dari penjejakan objek adalah mengikuti sebuah objek untuk tujuan tertentu. Penjejakan objek merupakan proses yang dilakukan oleh komputer vision. Untuk itu diperlukan algoritma yang dapat menganalisis setiap perubahan *frame* dalam video sehingga dapat diketahui adanya pergerakan objek beserta lokasi objek tersebut.

Beberapa metode telah dikembangkan sebagai cara untuk melakukan penjejakan objek. Diantaranya penelitian yang dilakukan oleh Hendy Mulyawan dkk (2011) tentang *object tracking* dengan metode *Template Matching* dan penelitian oleh Agustina dkk (2012) yang menggunakan metode *Mean-Shift* untuk *object tracking*. Dari hasil penelitian, metode *template matching* kesuksesan *tracking* pada kecerahan lingkungan yang stabil sebesar 59,94 % dengan proses komputerisasi yang cepat. Metode *mean-shift* dapat menjejaki gambar dengan

tingkat kesuksesan yang tinggi namun lebih lambat apabila objek mengalami perubahan skala dan orientasi.

Dengan meninjau keadaan yang terjadi diatas, maka penulis melakukan penelitian tentang Implementasi *Object Tracking* Pada Kamera Pengawas Semi-Otomatis Dengan *Image Processing* Metode *Mean-Shift*. Metode *mean-shift* dipilih karena tingkat keakuratan penjejakan yang tinggi sehingga nilai *error* untuk menggerakkan kamera kecil.

1.2 Rumusan Masalah

Dari latar belakang yang telah ditulis, didapatkan beberapa rumusan masalah yaitu :

a. Bagaimana cara merancang metode *Mean-Shift* pada *image processing* untuk menjejak objek ?
b. Bagaimana cara mengimplementasikan sistem *object tracking* pada kamera pengawas semi-otomatis ?

1.3 Batasan Masalah

Penelitian yang akan dilakukan dibatasi agar tidak terlalu luas, batasan-batasan masalah dari penelitian ini adalah :

a. Alat bekerja secara semi-otomatis dimana masih dibutuhkan operator sebagai penentu objek yang akan dijejaki.
b. Menggunakan laptop sebagai pengolah gambar.
c. Pergerakan alat hanya keatas dan kebawah (*pan* dan *tilt*).
d. Tidak ada cermin dalam ruang lingkup pengawasan alat.

1.4 Tujuan

Penelitian ini memiliki beberapa tujuan yaitu :

a. Merancang algoritma *image processing* dengan menggunakan metode *Mean-Shift* sebagai penjejakan objek.
b. Mengimplementasikan sistem *object tracking* pada kamera pengawas semi-otomatis.

1.5 Manfaat

Manfaat dari penelitian ini yaitu :

a. Dapat merancang algoritma *image processing* menggunakan metode *mean-shift* sebagai penjajak objek.
b. Dapat mengimplementasikan sistem *object tracking* pada kamera pengawas semi otomatis.

1.6 Sistematika Penelitian

Penyusunan laporan penelitian ini disusun sebagai berikut :

a. BAB 1. PENDAHULUAN
   Berisi latar belakang penelitian, rumusan masalah, batasan masalah, manfaat penelitian, dan sistematika penelitian.
b. BAB 2. TINJAUAN PUSTAKA
   Berisi landasan tori penelitian, teori-teori terkait, dan bahan penelitian.
c. BAB 3. METODOLOGI PENELITIAN
   Mejelaskan tentang metode yang digunakan untuk menyelesaikan penelitian.
d. BAB 4. HASIL DAN PEMBAHASAN
   Berisi data hasil penelitan serta analisa terkait tujuan dan hasil penelitian.
e. BAB 5. PENUTUP
   Pemaparan kesimpulan dan saran penulis.

BAB 2. TINJAUAN PUSTAKA

2.1 Penelitian Terkait

a. Aplikasi *Webcam* Untuk Penjejak Pergerakan Manusia Dalam Ruangan

Penelitian ini membahas implementasi webcam untuk menjejak pergerakan manusia pada suatu ruangan. Dalam permbacaan citra melalui webcam, tahapan yang dilakukan adalah mengubah citra RGB ke *grayscale* yang berfungsi untuk menyederhanaan model citra, kemudian menerapkan metode *frame differencing* sebagai perhitungan untuk mengurangi selisih nilai perbedaan gambar pada setiap *frame*. Kemudian dilakukan *filtering* dengan median *filtering* untuk menghilangkan noise berupa bintik-bintik. *Threshold*ing digunakan untuk mengkonversi citra *grayscale* ke citra biner sehingga diperoleh gambar dengan nilai 1 dan 0. Kemudian dilakukan segmentasi gambar untuk mendapatkan objek yang dijejaki. Setelah posisi objek diketahui, dikirimkan perintah ke mikrokontrol untuk melakukan pergerakan *tracking*.

b. Pelacakan Benda Bergerak Menggunakan Metode *Mean Shift* dengan Perubahan Skala dan Orientasi

Pada penelitian dikembangkan metode *mean-shift* untuk mengatasi perubahan orientasi dan skala yang baru. Penelitian ini menggunakan proses akuisisi citra dengan memilih file video dengan format AVI. Akusisi citra bertujuan untuk mendapatkan parameter jumlah *frame*, ukuran video, dan *frame rate*. Terdapat perubahan algoritma dari algoritma asli dari algoritma *mean-shift*. Perubahan algoritma dengan menambah algoritma untuk penentuan target kandidat yang berasal dari wilayah target yang dilakukan perluasan. Selain itu perkiraan skala dan orientasi dari obyek target yang ditemukan disisipkan setelah obyek target ditemukan. (Mahali dkk. 2014)

Perhitungan posisi baru pada penelitian dilakukan dengan cara mengenali perubahan *frame* yang terjadi. Apabila pergeseran *frame* melibihi nilai ambang, maka pergeseran tersebut dapat dicari posisi baru. Selisih titik pusat objek baru

dengan objek sebelumnya dapat digunakan untuk menentuka lebar wilayah yang kemudian dapat ditentukan perkiraan wilayah target kandidat selanjutnya.

2.2 Pengolahan Citra

Gambar yang diperoleh dari kamera pada dasarnya adalah suatu nilai cahaya dua dimensi yang diambil pda waktu tertentu. Untuk melakukan proses pengolahan oleh komputer diperlukan 3 buah tahapan yaitu :

1. Nilai cahaya dipetakan menjadi sebuah matriks 2 dimensi
2. Matriks yang didapat kemudian di *sampling* agar didapatkan fungsi matematika yang mempresentasikan suatu gambar bergantung pada waktu.
3. Hasil dari matriks dilakukan kuantitasi untuk mendapatkan nilai bulat yang akan diolah komputer.

a. Jenis Gambar

Untuk dapat diproses agar mendapatkan data tertentu dari sebuah gambar, diperlukan jenis gambar yang sesuai. Biasanya digunakan beberapa jenis gambar yaitu : gambar RGB, gambar *grayscale*, gambar *binary*, dan *special image*.

Gambar RGB merupakan gambar berwarna dengan 3 warna dasar yaitu merah, hijau, biru (*Red, Green, Blue*). Masing-masing intensitas warna umumnya ditampung pada *channel* 8 bit sehingga secara keseluruhan membutuhkan minimal 24 bit.

Gambar 2.1 Tampilan *image* RGB

Grayscale merupakan jenis warna yang mempresentasikan intensitas warna pada kanal dengan jumlah bit tertentu. Pada gambar *grayscale* 8 bit, nilai 0 merupakan nilai dengan nilai intensitas paling rendah yaitu warna hitam dan 255 merupakan nilai *grayscale* dengan intensitas tertinggi yaitu warna putih yang merupakan campuran dari semua warna.

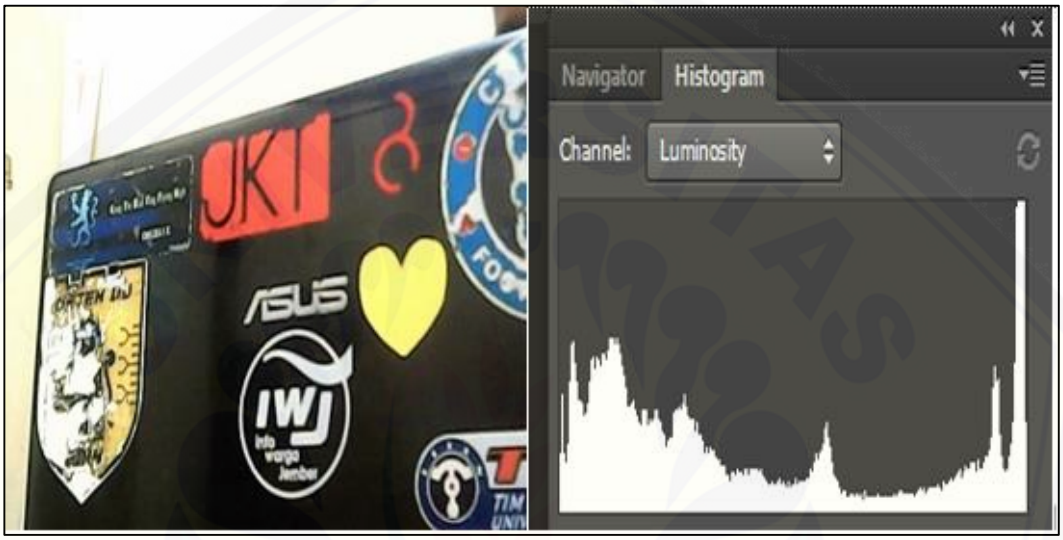
Gambar 2.2 Tampilan *image grayscale*

Gambar *binary* adalah gambar yang termasuk gambar *grayscale* yang mempresentasikan gambar dengan dua buah keadaan (1 bit) yaitu hitam dan putih. Gambar *binary* merupakan jenis gambar yang sederhana untuk diproses karena hanya memiliki 1 bit setiap kanalnya.

Gambar 2.3 Tampilan *image* binary

b. Peningkatan Kualitas Citra

Histogram adalah sebuah grafik yang menunjukkan frekuensi kemunculan suatu warna pada gambar. Pada grafik dengan koordinat kartesian, sumbu X merupakan nilai warna dari gambar dan sumbu Y merupakan jumlah kemunculan nilai warna tersebut. Gambar dibawah merupakan contoh gambar dengan histogramnya.



Gambar 2.4 Citra dan histogramnya

Histogram berfungsi sebagai indikator visual agar dapat ditentukan nilai keabuan yang sesuai dengan kebutuhan sehingga dapat diperoleh citra yang dibutuhkan. Dengan diketahuinya nilai skala keabuan, dapat diubah tingkat kecerahan, kontras, dan lain-lain. Histogram juga dapat dimanfaatkan untuk menentukan nilai ambang (*threshold*) pada suatu citra dengan tujuan pemisahan batas-batas objek terpilih dari latar belakang.

Filtering merupakan teknik yang digunakan untuk menekan gangguan atau derau (*noise*) pada gambar. Gangguan pada gambar dapat merupakan perbedaan suatu piksel dengan piksel sekitarnya. Teknik *filtering* memiliki beberapa metode diantaranya *mean filtering*, *median filtering*, *modus filtering*, dan gaussian *filtering*.

Mean filtering adalah metode *filter* yang mengambil nilai rata-rata dari suatu piksel dengan piksel disekitarnya. Piksel yang akan di-*filter* dimasukkan pada matirx N x N dimana N harus bernilai ganjil.

1	2	3
4	T	5
6	7	8

Gambar 2.5 Matriks *mean filtering*

Gambar 2.5 merupakan *mean filtering* dengan N bernilai 3. T merupakan piksel yang akan di-*filter*, sedangkan 1, 2, 3, 4, 5, 6, 7, dan 8 merupakan nilai piksel disekitar piksel T. Nilai T yang baru merupakan nilai rata-rata dari semua piksel.

Modus filtering merupakan pengambilan nilai terbanyak dari suatu kumpulan nilai pada suatu matriks piksel. Piksel utama berada ditengah matriks, sedangkan nilai disekitar matriks utama merupakan nilai dari piksel yang berada disekitar piksel utama. Dari matriks yang telah ditentukan, diambil nilai dengan frekuensi kemunculan terbanyak yang akan menggantikan nilai piksel utama. Karena piksel utama harus diitari oleh piksel lain, maka nilai *modus filtering* harus bernilai ganjil.

2	3	5
8	8	4
8	3	9

Gambar 2.6 Matriks *modus filtering* dengan hasil 8

Median filtering adalah penggantian nilai suatu piksel dengan nilai tengah dari kumpulan data disekitar piksel utama. Untuk *median filtering* ini, data yang digunakan untuk menghitung *median* terdiri dari kumpulan data yang ganjil. Hal ini dikarenakan dengan jumlah data yang ganjil maka piksel yang akan diproses akan berada ditengah.

2	3	3	4	5	8	8	8	9

Gambar 2.7 Urutan *median* dari matriks

Gambar 2.7 merupakan urutan yang dari matriks pada gambar 5 sehingga dapat diketahui nilai piksel barunya adalah 5.

c. Segmentasi

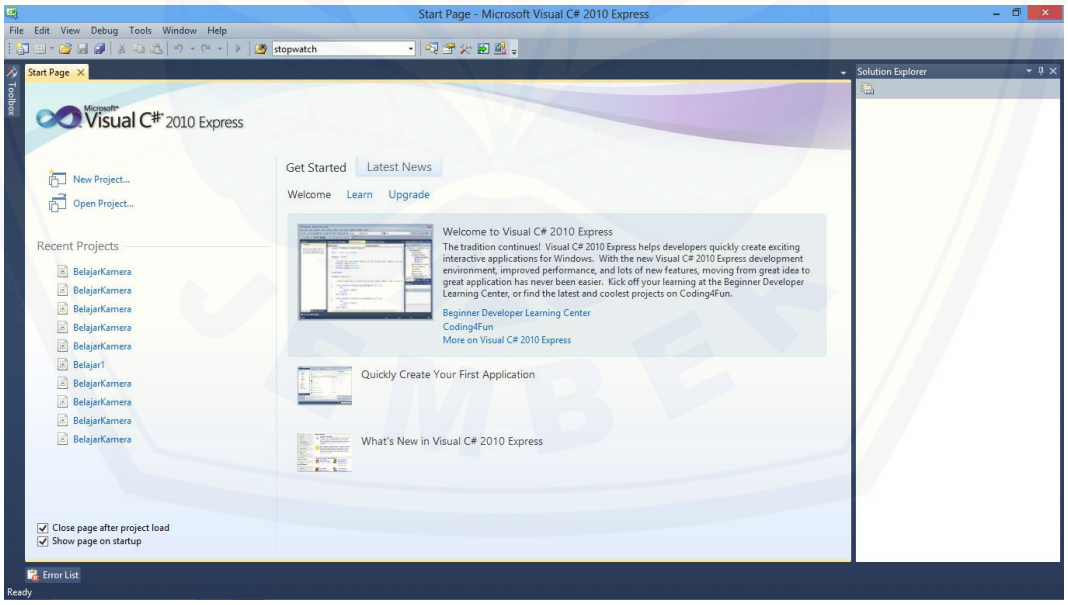
Segmentasi adalah proses pemilihan suatu objek dari keseluruhan gambar. Segmentasi bertujuan mengklasifikasi gambar sehingga objek dapat dibedakan. Dalam proses segmentasi, terdapat beberapa metode yang dapat dilakukan antara lain segmentasi berdasarkan klasifikasi, segmentasi berdasar daerah tepi (*edge*), dan segmentasi berdasarkan daerah (*region*).

Segmentasi citra berdasarkan klasifikasi adalah mencari kesamaan-kesamaan pada piksel, dapat berupa warna atau ukuran. Segmentasi berdasar tepi adalah mendeteksi suatu garis dengan anggapan setiap garis adalah batas dari tiap objek dengan objek lain atau objek dengan latar belakang. Sedangkan segmentasi daerah adalah segmentasi yang menentukan suatu daerah tertentu pada gambar yang diyakini merupakan objek yang akan dicari berdasarkan kesamaan tekstur.

2.3 Microsoft Visual Studio

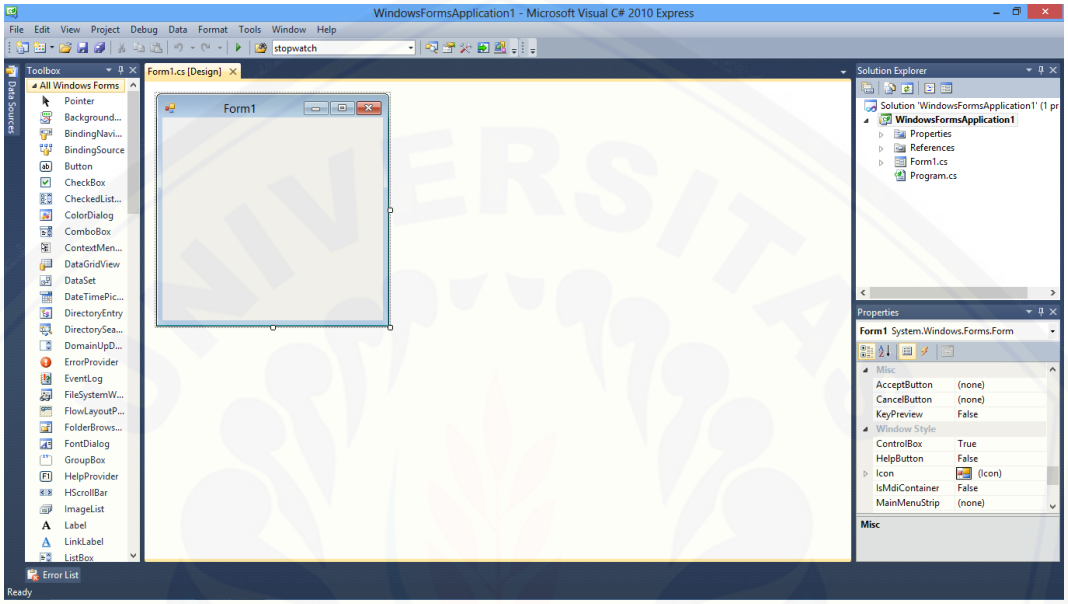
Microsoft Visual Studio adalah sebuah aplikasi bantu terpadu dalam hal pemrograman yang dibuat dan dikembangkan oleh Microsoft Corporation dan berguna untuk mendesain sebuah aplikasi grafis yang berbasis bahasa C++, C#, BASIC, dan bahasa serupa lainnya. Microsoft Visual Studio dapat digunakan untuk mendesain berbagai aplikasi seperti aplikasi bisnis, aplikasi *console*, aplikasi personal dan juga pengembangan aplikasi lainnya. Untuk menjalankan aplikasi buatan Visual Studio, dibutuhkan sebuah *platform* yaitu Net Framework.

Net Framework merupakan sebuah *platform* buatan Microsoft yang berfungsi sebagai pembangun dan digunakan untuk memulai sebuah aplikasi. Net framework dibutuhkan untuk menjalankan aplikasi dengan menggunakan antarmuka dengan layanan *Graphic User Interface* (GUI). Net framework merupakan *platform* yang diciptakan untuk menghubungkan sistem dalam aplikasi dengan informasi yang dimuat serta dengan hubungan dengan *device* yang terkait, sehingga manusia sebagai *user* dapat melakukan proses antarmuka dengan aplikasi secara lebih mudah dan *user friendly*.



Gambar 2.8 Halaman Start Page pada Visual Studio 2010

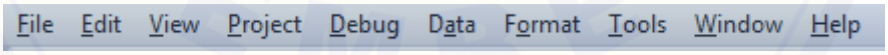
Integrated Development Environtmen (IDE) merupakan halaman pada visual studio dimana kita dapat mengembangkan sistem aplikasi pada *compiler*, pembuatan desain, dan penggunaan komponen lainnya. IDE berperan penting dalam proses pengembangan aplikasi, karena pada halaman ini, programmer akan menerapkan pemikiran yang telah ada menjadi aplikasi nyata yang akan dibuat.



Gambar 2.9 Tampilan Halaman *Integrated Development Environtment*

Pada halaman IDE terdapat beberapa komponen item yang memiliki fungsi masing-masing diantaranya adalah menu bar, toolbar standart, solution explorer, properties windows, toolbox, dan *error list*.

1. Menu Bar



Gambar 2.10 Menu Bar Visual Studio

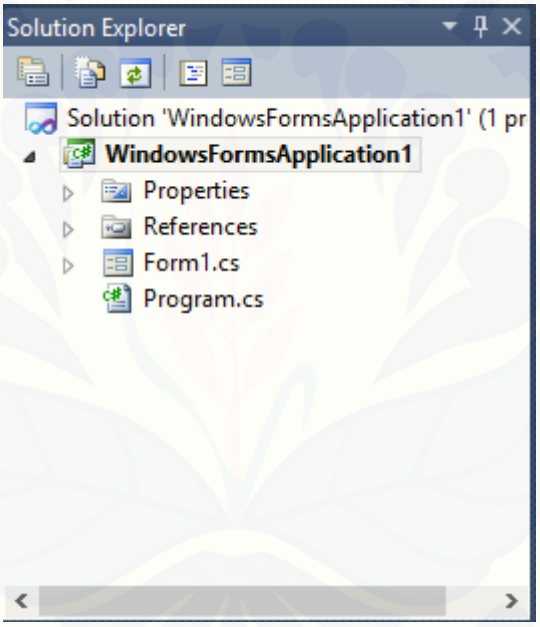
Menu bar adalah baris menu horizontal yang terdapat pada bagian atas halaman IDE. Menu bar berisi pilihan menu utama seperti menu file, edit, view, windows dan lain lain.

2. Toolbar Standart

Gambar 2.11 Tampilan toolbar standart pada visual studio

Toolbar berisi kumpulan item-item yang pasti digunakan seperti new form, open file, save file, save as, item editing, iten compiler dll. Dengan adanya toolbar, kita dimudahkan untuk mengakses menu-menu penting dalam visual studio tanpa membuka isi dari manu bar.

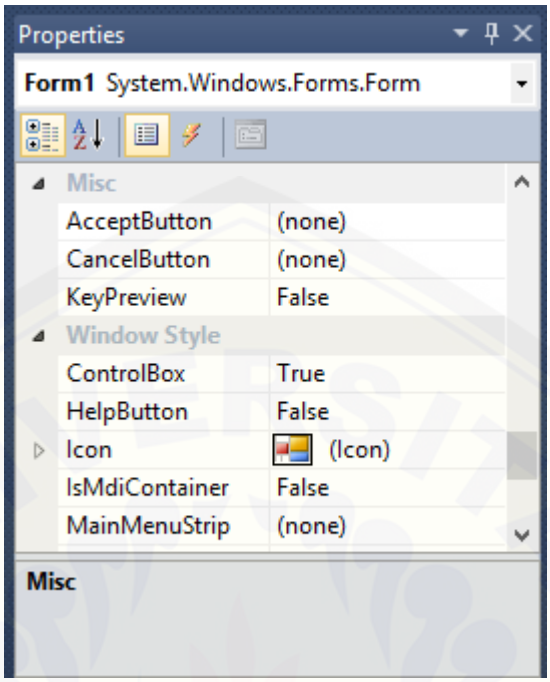
3. Solution Explorer



Gambar 2.12 Jendela Solution Explorer

Saat membuat aplikasi, pasti dibutuhkan beberapa file penunjang seperti kode program, file gambar, file suara dan file tambahan lainnya yang perlu diikut-sertakan dalam proses compiling agar dalam menjalankan program exe atau web file tersebut akan ikut dijalankan. Solution explorer menanmpung file-file tersebut sehingga memudahkan pengguna dalam mengelola project. Pada pertama kali melakukan pengerjaaan project, solution explorer terdapat pada sebelah kiri halaman. Namun dapat dipindah dengan melakukan dragging, untuk memunculkan solution explorer dapat dipilih pada menubar views.

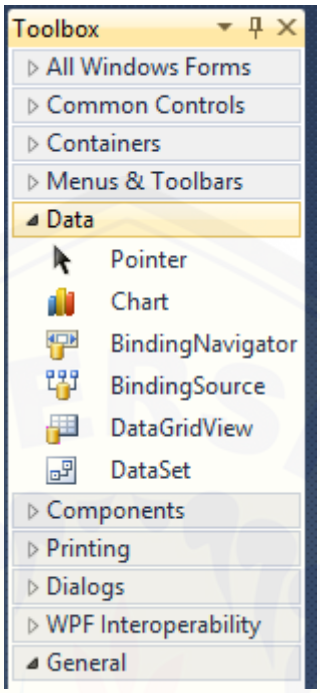
4. Properties Windows



Gambar 2.13 Tampilan jendela properties windows

Properties windows adalah tampilan daftar opsi pada setiap komponen aktif. *User* dapat mengubah opsi pada setiap komponen dengan cara memilih properti yang sesuai. Properties windows terdapat pada sebelah kanan bawah halaman IDE. Untuk menampilkan properties windows dapat dilakukan dengan cara memilih menu view pada menu bar, kemudian memilihi tab Properties. Kemudian untuk mengubah properti dari item, langkah awal yang harus dilakukan adalah memilih item kemudian sesuaikan opsi.

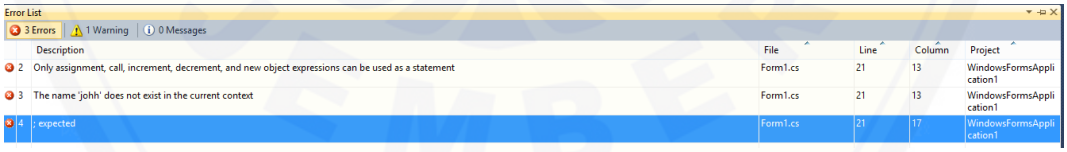
5. Toolbox



Gambar 2.14 Toolbox visual studio 2010

Toolbox adalah himpunan komponen-komponen control yang dapat ditambahkan dalam form project secara drag and drop. Item yang ditampilkan pada toolbox adalah item yang telah dijadikan referensi. *User* dapat menambah atau mengurangi item secara manual.

6. *Error List*



Gambar 2.15 Daftar *error* pada *error list*

Error list adalah jendela yang menampilkan informasi tentang pesan *error* secara spesifik. Dengan menggunakan *error list*, aplikasi dapat dikembangan dengan lebih cepat.

2.4 Metode *Tracking Mean-shift*

Untuk mendapatkan data lokasi objek yang akan dijejaki, digunakan algoritma *mean-shift* yaitu merupakan algoritma teknik analisis *non-parametric space* atau analisis ruang fitur berbasis non-parametrik. Menurut Mahali (2014), Algoritma *Mean-Shift* klasik tidak dapat mengukur objek dengan perubahan skala dan orientasi sehingga diperlukan modifikasi dari algoritma *Mean-Shift* klasik.

Sebelum menggunakan metode *mean-shift* diperlukan adanya representasi target model atau pemilihan model target dan representasi target kandidat. Representasi target model merupakan probabilitas dari histogram target yang ditentukan.

$$\hat{q}_u = C \sum_{i=1}^{n} k(\|x_1^{\cdot}\|^2)\delta(b(x_1^{\cdot}) - u)$$

$\{x_\iota\}\, \iota \doteq 1 \ldots n$ merupakan lokasi piksel yang telah dinormalisasi dari target, dan $b(x_\iota^{\cdot})$ adalah nilai RGB piksel pada lokasi $\dot{x_\iota}$. Sedangkan C adalah konstanta normalisasi yang ditunjukkan oleh persamaan :

$$C = \sum_{i=1}^{n} k(\|x_1^{\cdot}\|^2)^{-1}$$

Representasi target kandidat merupakan probabilitas histogram objek pada tiap *frame* yang memiliki panjang dan lebar sama dengan panjang dan lebar target model. Probabilitas dari target kandidat dinyatakan dalam persamaan :

$$\hat{p}_u = C_2 \sum_{i=1}^{n} k(\|x_1^{\cdot}\|^2)\delta(b(x_1^{\cdot}) - u)$$

Dimana Ch merupakan konstanta yang diperoleh dari persamaan :

$$C_h = \sum_{i=1}^{n} k(\|x_1^{\cdot}\|^2)^{-1}$$

Representasi target model dan representasi target kandidat membantu dalam perhitungan lokasi y agar nilai $\widehat{q_u}$ mendekati nilai $\widehat{p_u}$(y). Setelah lokasi saat ini diketahui ($\widehat{y_0}$) maka lokasi ($\widehat{y_1}$) dapat diketahui dengan persamaan berikut :

$$\hat{y}_1 = \frac{\sum_{i=1}^{n} x_1 \omega_1 g(\|y_0 - x_i\|)^2}{\sum_{i=1}^{n} \omega_1 g(\|y_0 - x_i\|)^2}$$

Dari persamaan-persamaan yang telah didapat, sebuah algoritma penjejakan objek dapat disusun sebagai berikut (Sulfan dkk, 2014) :

1. Menghitung target model $\widehat{q_u}$
2. k=0
3. Menghitung probabilitas target kandidat $\widehat{p_u}$
4. Menghitung bobot
5. Menghitung lokasi *update*
6. Mengatur nilai d = $\|y_1 - y_0\|, y_1 = y_0, k = k + 1$

Perhitungan kernel *density estimation* (KDE) digunakan dalam algoritma *mean-shift*. Perhitungan menggunakan rumus sebagai berikut :

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} K_H(x - xi)$$

x = perulangan dari panjang n

K = *kernel* simetrikal

H = parameter *smooting*

Fungsi dari KDE adalah untuk menghitung intensitas warna paling sering muncul pada lingkup tertentu. Pada suatu area yang didominasi warna tertentu, maka semua warna pada area tersebut akan berwarna yang mendominasi. Sehingga pada suatu citra akan terbagi menjadi area-area dengan warna yang homogen.

2.5 Arduino UNO R3

Arduino merupakan papan mikrokontroler terpadu yang merupakan pengembangan dari mikrokontroler konvensional seperti AVR. Kelebihan arduino dibandingkan mikrokontroler lain diantaranya adalah memiliki bahasa pemrograman sendiri yang merupakan modifikasi dari bahasa pemrograman C++. Arduino juga memiliki konverter USB *to Serial* yang tergabung dalam *board* sehingga komunikasi serial menjadi lebih praktis.

Gambar 2.16 Arduino UNO R3 (sumber: arduino.cc)

Arduino UNO R3 adalah jenis arduino yang menggunakan mikrokontroler AVR ATMega328P sebagai chip utama. Diantara jenis arduino lain, arduino UNO memiliki ukuran yang kecil. Berikut spesifikasi dari Arduino UNO R3 :

Tabel 2.1 Spesifikasi Arduino UNO R3

Chip	ATMega328P
Tegangan Operasi	5 V
Tegangan *Input* (Rekomendasi)	7 – 12 V
Tegangan *Input* (Limit)	6 – 20 V
Pin I/O Digital	14 (6 merupakan PWM)
Pin *Input* Analog	6
Arus DC pin I/O	20 mA
Arus DC pin 3.3 V	50 mA
Flash Memory	32 KB, 0,5 KB untuk *bootloader*
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

(sumber : arduino.cc)

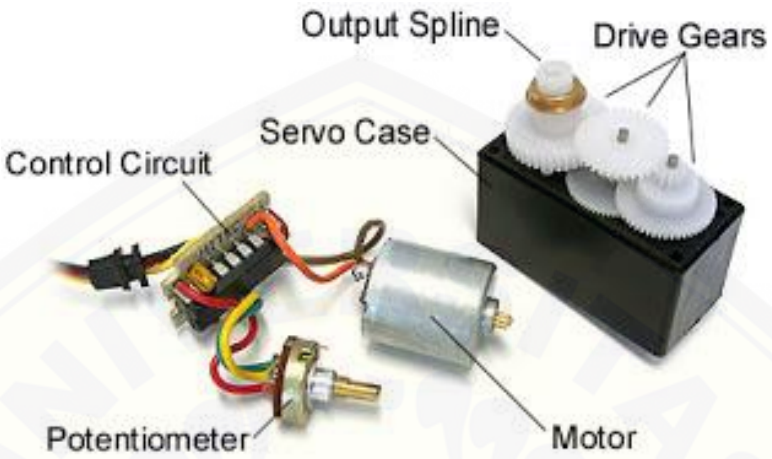
2.6 Motor Servo

Motor servo merupakan aktuator dengan torsi tinggi yang bergerak secara berputar (motor DC) dan dapat ditentukan sudut putarannya. Pada dasarnya, motor servo bekerja seperti motor DC biasa, namun servo dirancang mempunyai sensor umpan balik untuk mendapatkan posisi poros servo sehingga servo dapat dikontrol untuk menentukan posisi porosnya secara presisi. Penyesuaian posisi dilakukan dengan cara mengetahui posisi poros pada saat ini, dibandingkan dengna posisi poros yang diinginkan.

Gambar 2.17 Motor servo GWS S03N STD (sumber : pololu.com)

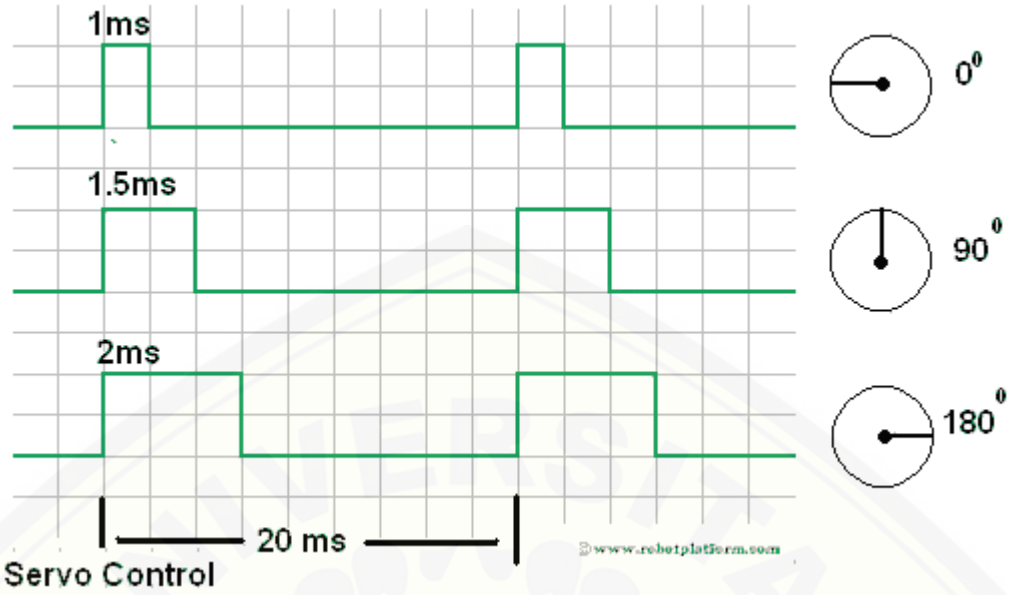
Pada umumnya, motor servo dibedakan menjadi 2 yaitu motor servo dengan *range* putaran 180º dan motor servo *continuous*. Motor servo standar hanya dapat bergerak dalam dua arah searah putaran jarum jam dan berlawanan arah dengan jarum jam, defleksi dari putaran tersebut adalah maksimal 90 derajat dari titik tengah sehingga total *range* sudut yang dapat dijangkau oleh motor servo jenis ini

adalah 180°. Sedangkan motor servo *continuous* tidak memiliki batasan sudut putaran. Dengan prinsip kerja yang demikian, motor servo cocok digunakan untuk keperluan pergerakan perangkat yang membutuhkan ketelitian pergerakannya.



Gambar 2.18 Konstruksi motor servo standar (sumber : 4.bp.blogspot.com)

Konstruksi dari motor servo adalah motor DC, potensiometer, set *gearbox*, dan rangkaian kendali elektronik. Untuk memperkuat torsi pada motor servo, digunakan redaman kecepatan menggunakan gear-gear yang sudah *build-in* pada body servo sehingga tuas akhir memiliki torsi yang tinggi. Dengan konstruksi tersebut, pengendalian motor servo dapat dilakukan dengan mudah dengan torsi yang cukup memadai. Motor servo memiliki 3 buah pin kabel keluaran dimana fungsi kabel pin tersebut adalah untuk supply (5 v dan 0 V) kemudian kabel yang merupakan pin data.



Gambar 2.19 Pemberian pulsa pada servo

Pengendalian motor servo dilakukan dengan cara mengubah lebar pulsa yang diberikan pada pin data motor servo. Motor servo akan bekerja pada frekuensi 50 Hz dan untuk mengendalikannya, perlu diubah lebar pulsa *high* antara 1 ms sampai dengan 2 ms. Pulsa *high* antara 1 ms sampai 2 ms tersebut akan menggerakkan servo dengan *range* yang telah ditentukan yaitu 0º sampai 180º.

BAB 3. METODOLOGI PENELITIAN

3.1 Alat dan Bahan

Alat dan bahan utama dan penunjang pada penelitian ini tercangkup pada tabel 3.1. Dimana telah dijelaskan secara singkat nama alat dan bahan serta spesifikasi teknis dari alat dan bahan tersebut.

Tabel 3.1 Daftar Alat dan Bahan Penelitian

No	Nama Alat	Spesifikasi Teknis dan Penjelasan
1.	Webcam	**A4Tech Webcam PK-710G**. Image Sensor 640 x 480 pixels. Frame Rate 30 fps. View Angle 60 Degrees.
2.	Mikrokontroler	**Arduino UNO R3**. CPU ATMega328P. Tegangan operasi 5V. 14 Pin I/O Digital. Clock Speed 16 Mhz.
3.	Motor Servo	**MG996 Metal Gear Servo**. Torsi 12 Kg/Cm.
4.	*Driver* Motor	**Relay Modul**. Tegangan Operasi 5V. SPDT Relay.
5.	Laptop	**Asus X45U**. Prosesor AMD E-350 APU with Radeon(tm) HD Graphics (2 CPUs), ~1.6GHz. Memory 2048. Installed Visual Studio 2010
6.	Alat Pendukung	Obeng, Solder, Timah, dll.

3.2 Ruang Lingkup Kegiatan

Dalam melakukan penelitian, ditentukan beberapa batasan-batasan agar penelitian tidak menjadi terlalu luas. Berikut merupakan ruang lingkup yang digunakan pada penelitian ini :

a. Alat bekerja secara semi-otomatis dimana masih dibutuhkan operator sebagai penentu objek yang akan dijejaki.
b. Menggunakan laptop sebagai pengolah gambar.
c. Pergerakan alat hanya keatas dan kebawah (*pan* dan *tilt*).
d. Tidak ada cermin dalam ruang lingkup pengawasan alat.

3.3 Metode Pengambilan Data

Data-data pada penelitian yang akan dilakukan diambil dengan langkah sebagai berikut :

a. Studi Literatur.

Data dari penelitian terdahulu diperlukan untuk menjadi referensi dalam melakukan penelitian ini sehingga dapat dikembangkan menjadi penelitian berlanjut dan data yang dapat dari penelitian ini mempunyai data pembanding yang sesuai.

b. Merancang pembuatan perangkat keras dan perangkat lunak.

c. Melakukan pembuatan perangkat keras dan perangkat lunak.

d. Melakukan kalibrasi dan pengujian perangkat secara mandiri dan terpadu.

Sebelum alat dirangkai dan menjadi rangkaian terpadu, setiap sensor dan aktuator yang akan digunakan diuji dan didapatkan data mandiri dari setiap komponen. Untuk komponen kamera, kalibrasi dilakukan pada intensitas cahaya lingkungan sampel agar cahaya lingkungan yang digunakan sesuai dengan kriteria yang diperlukan. Pada komponen motor servo, akan diuji langkah sudut putar motor setiap suatu sinyal diberikan. Setelah alat terangkai menjadi rangkaian terpadu, data awal yang diambil adalah kesesuaian gerak motor pada *object* yang akan dijejaki sehingga didapat nilai *error* pada sumbu x dan *error* pada sumbu y.

e. Menganalisa data dari hasil penelitian.

Data yang didapatkan setelah dilakukan pengujian diolah dan dianalisa keterkaitannya dengan tujuan penelitian sehingga tujuan penelitian dapat tercapai.

3.4 Alur Penelitian

Gambar 3.1 Alur Penelitian

Studi pustaka dilakukan dengan menganalisa jurnal terkait dengan penelitian sehingga didapatkan informasi dan mendapatkan gambaran tujuan penelitian, tahapan pengerjaan sampai penelitian selesai, dan juga pemahaman teori menganalisa data sehingga resiko dari penelitian yang dilakukan dapat dikurangi. Setelah perencanaan dilakukan, perlu dipilih alat dan bahan yang dibutuhkan kemudian diuji karakteristiknya sehingga sesuai dengan yang dibutuhkan. Setelah alat dan bahan siap, perancangan alat dilakukan hingga dapat diambil data yang diperlukan agar tujuan dari penelitian dapat tercapai.

3.5 Perancangan Alat

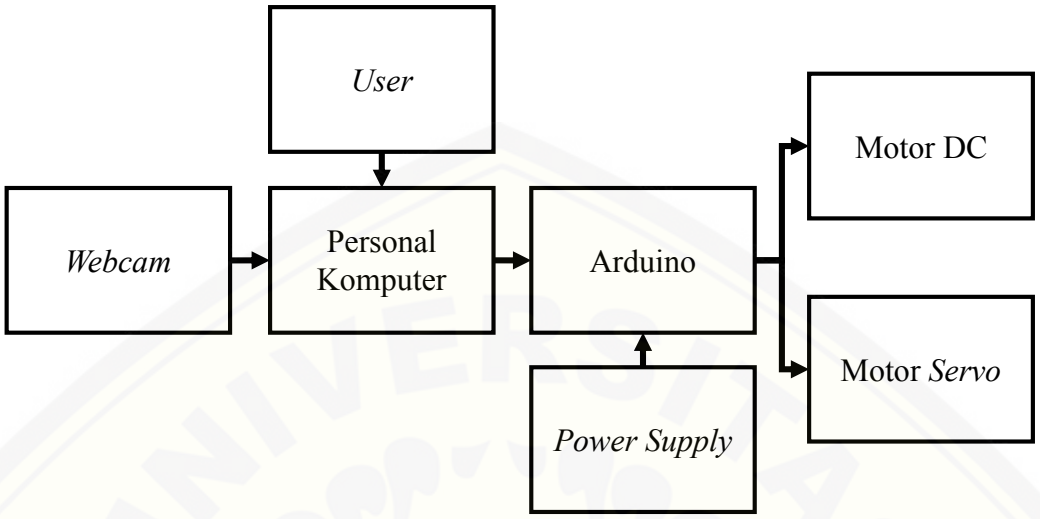
3.5.1 Perancangan Desain Alat

Gambar 3.2 Desain alat tampak depan

Gambar 3.3 Desain alat tampak samping

3.5.2 Perancangan Perangkat Keras

a. Blok Diagram

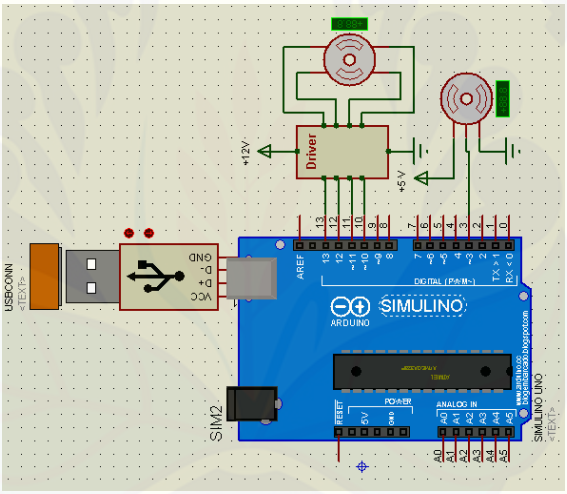


Gambar 3.4 Blok diagram perancangan perangkat keras

Personal komputer sebagai pengolah gambar mendapatkan masukan dari kamera *webcam* berupa gambar *real time* dan *user* memberikan masukan berupa objek yang akan dijejaki. Penentuan objek dilakukan oleh *user* melalui perangkat mouse dengan cari *click and drag* pada objek yang dipilih. Personal komputer melalui aplikasi Visual C# mengolah gambar dan masukan dari *user* untuk selanjutnya menghasilkan data-data. Hasil pengolahan gambar yang berupa data-data koordinat dll dijadikan masukan oleh arduino untuk menggerakkan kamera menggunakan motor DC dan motor *servo*. Motor DC digunakan untuk menggerakkan kamera kearah samping kiri dan kanan (sumbu horizontal) dengan kecepatan rendah sedangkan motor Servo berfungsi sebagai penggerak kamera keatas dan kebawah (sumbu vertikal) dengan batasan 0 – 180 derajat sesuai dengan *range* sudut putar servo. *Power supply* memberikan tegangan sebesar 5V kepada arduino dan aktuator sebagai tegangan kerja dari perangkat tersebut.

b. Rangkaian pengendali

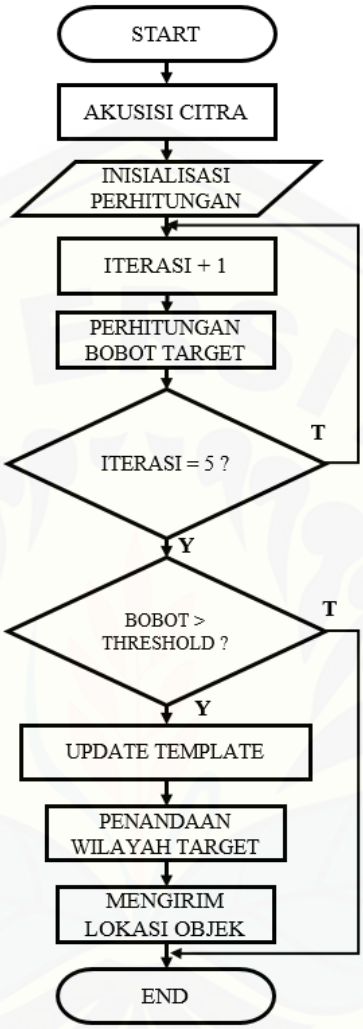
Arduino dihubungkan ke komputer melalui komunikasi serial. Untuk menggerakkan motor DC, dibutuhkan 2 pin data dan 2 pin power yang terhubung ke *driver* agar arus dan tegangan dari arduino sesuai dengan tegangan yang dibutuhkan motor DC. Pin data yang dihubungkan adalah pin 8 dan 9. *Driver* motor menggunakan rangkaian *driver* terpadu menggunakan *Opto-coupler* dan relay. Motor servo membutuhkan pwm sebagai penggerak rotor-nya, pin data pada servo dengan kabel berwarna kuning dihubungkan ke pin 3 arduino, pin sumber tegangan servo dengan kabel berwarna merah dihubungkan dengan sumber tegangan 5 volt dan pin 0 volt dengan kabel berwarna hitam dihubungkan pada pin gnd pada arduino.



Gambar 3.5 Rangkaian pengendali motor

3.5.3 Perancangan Perangkat Lunak

a. *Flowchart* utama pengolahan gambar

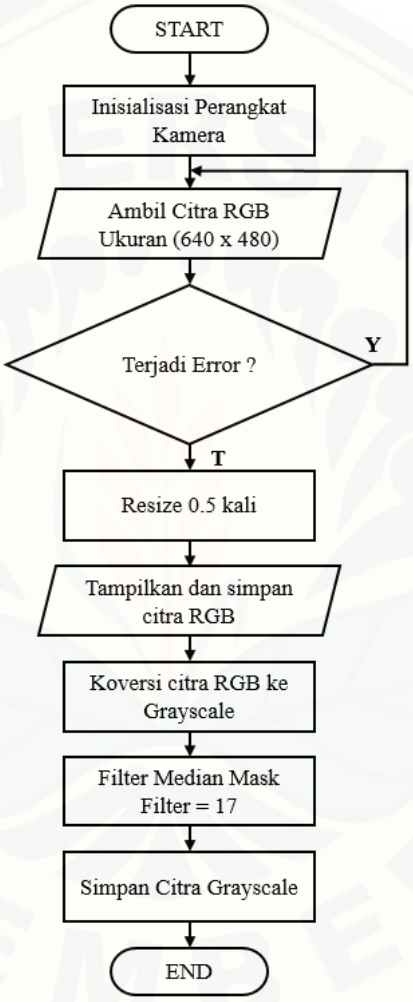


Gambar 3.6 *Flowchart* pengolahan gambar

Pada gambar 3.6, langkah awal yang dilakukan adalah akusisi citra dimana pengambilan gambar dilakukan secara *real-time*. Gambar yang diperoleh akan diproses sehingga diperoleh area yang merupakan gambar target. *Thresholding* dilakukan setelah objek yang dipilih telah ditentukan. Nilai threshold merupakan nilai variabel yang dapat diubah oleh *user* dengan range nilai 0.1 sampai 1. Dari area objek yang terpilih, akan dilakukan perhitungan untuk mengetahui bobot objek sebagai acuan. Pada iterasi pertama, bobot target dihitung. Proses perhitungan bobot target dilakukan sampai iterasi pertama kali dilakukan hingga selesai atau sampai nilai bobot target sudah tidak sesuai dengan nilai *threshold*. Setelah proses

iterasi selesai, akan dilakukan perubahan *template* agar *template* terus *update*. Target yang telah ditemukan akan ditandai pada gambar dan lokasi koordinat objek dikirim ke mikrokontroler. Setelah itu data yang merupakan koordinat dari objek yang ditemukan dikirim ke arduino dengan menggunakan komunikasi serial.

b. *Flowchart* akuisisi citra

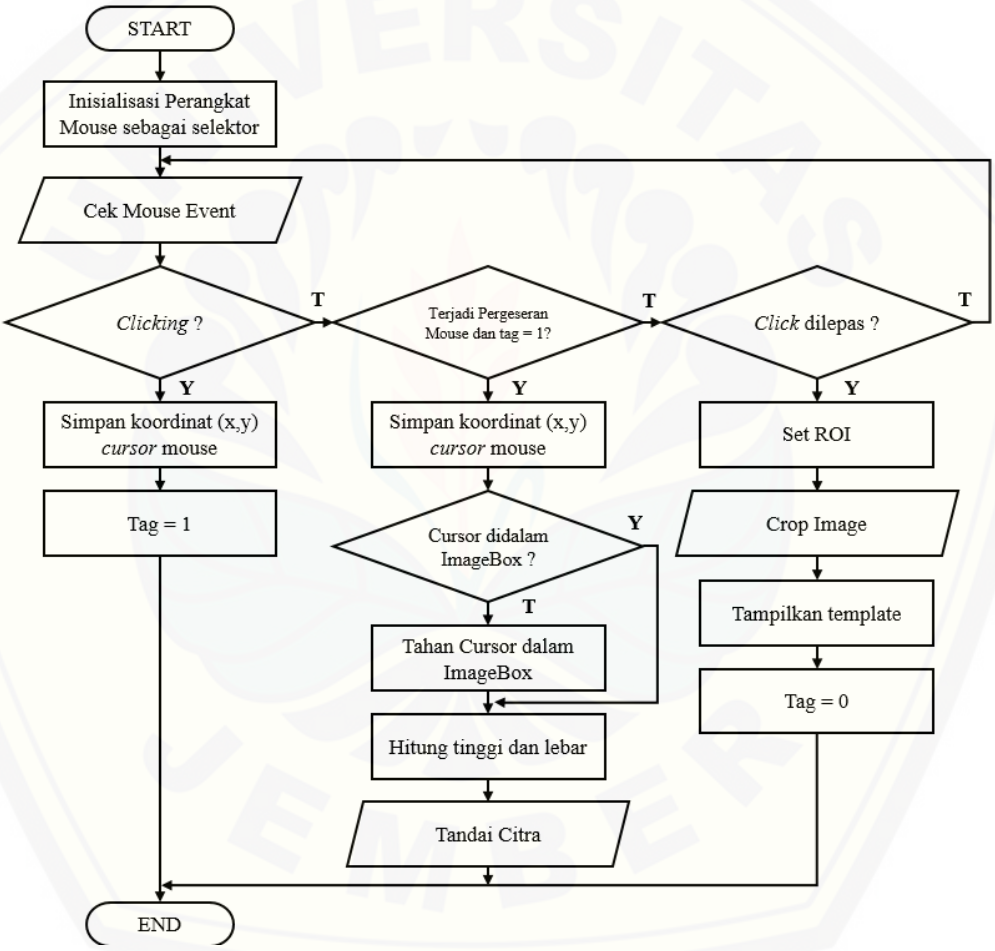


Gambar 3.7 *Flowchart* akuisisi citra

Perangkat kamera harus dapat dikenali oleh PC sehingga perlu diinisialisasi terlebih dahulu. Selanjutnya kamera mengambil gambar dari lingkungan dengan resolusi 640 x 480 piksel dan apabila terjadi *error* pada proses pengambilan gambar maka proses pengambilan gambar akan diulangi, apabila pengambilan gambar sukses, citra diubah ukuran dengan skala 2:1 atau setengah kali lebih kecil dari

ukuran awal sehingga ukuran gambar menjadi 320 x 240 piksel untuk mempersingkat waktu pemrosesan gambar. Kemudian citra ditampilkan sebagai antarmuka dengan *user*. Selanjutnya citra RGB diubah menjadi citra *Grayscale* agar lebih mudah dan cepat pengolahannya. Citra *Grayscale* selanjutnya difilter menggunakan metode filter median dengan nilai *mask* filter sebesar 17. Citra hasil *filtering* disimpan untuk diolah pada proses pemrosesan gambar.

c. *Flowchart* pemilihan *template*

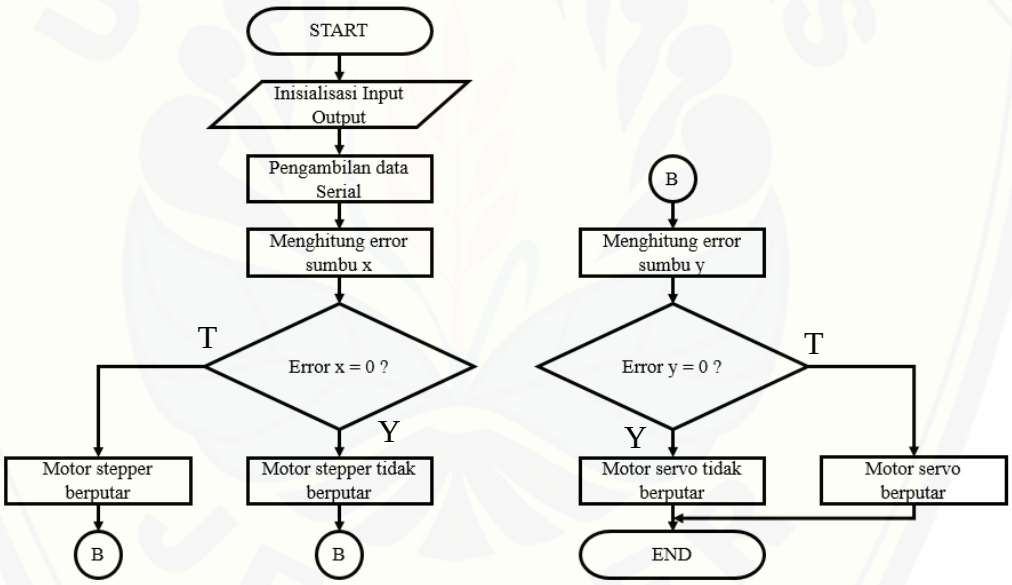


Gambar 3.8 *Flowchart* pemilihan *template*

Gambar 3.8 menunjukkan langkah dalam pemilihan *template* objek. Pemilihan objek dilakukan dengan cara klik lalu menarik mouse pada gambar yang akan dijadikan *template*. Langkah awal yang dilakukan adalah cek *event* pada mouse. Apabila ada *event clicking*, maka koordinat dari *cursor* disimpan dan label

tag diberi nilai 1. Jika tidak ada *event click* maka akan akan dicek apakah ada *event* pergeseran *cursor*, jika ada dan jika tag bernilai 1 maka koordinat *cursor* akan disimpan kemudian jika *cursor* tidak dalam area imagebox maka *cursor* akan ditahan dalam area imagebox. Selanjutnya tinggi dan lebar dari koordinat awal akan dihitung kemudian citra akan ditandai. Namun jika tidak terjadi pergesean mouse, *event* selanjutnya yang di cek adalah *event click* yang dilepas, jika ada *event* ini maka ROI akan diset dari koordinat *cursor* mouse. Citra pada imagebox dipotong (*crop*) sesuai dengan ROI setelah itu citra hasil *cropping* ditampilkan dan ditandai sebagai *template* baru.

d. Flowchart pengendalian motor



Gambar 3.9 *Flowchart* pengendalian motor

Data posisi objek diterima oleh serial arduino secara serial yang telah didapat dari kamera selanjutnya diproses oleh mikrokontroler untuk mencari *error*. Pada sumbu x, *error* akan mengakibatkan motor DC berputar untuk menjejak objek. Namun apabila tidak ada *error*, makan motor DC akan berhenti berputar. *Error* sumbu y mengakibatkan pergerakan pada motor servo untuk mengurangi *error* sumbu vertikal dan akan berhenti apabila *error* bernilai 0.

3.6 Rencana Data Pengujian

Setelah alat bekerja sesuai dengan rancangan, maka perlu direncanakan data apa saja yang harus didapatkan dari hasil pengujian agar tujuan penelitian dapat dicapai. Data yang akan dicari sebagai berikut :

1. *Object* diam pada lokasi tertentu, kemudian akan diuji kecepatan alat dalam melakukan *tracking* hingga objek berada pada tengah layar kamera.
2. Pengujian iterasi, objek bergerak dengan kecepatan dan lintasan yang sama kemudian dilakukan *tracking* dengan berbagai nilai variabel iterasi. Keluaran yang akan diuji adalah keberhasilan *tracking* dengan perbedaan nilai iterasi dan waktu CPU dalam melakukan proses *tracking*.
3. Pengujian ukuran model target dilakuakuan dengan cara mengubah-ubah ukuran *template* model target kemduain menguji pada lintasan *tracking* yang sama. Objek yang digunakan bergerak dengan kecepatan yang tetap.
4. Pengujian pengaruh *increment area* terhadap keberhasilan penjejakan objek. Pengujian ini dilakukan dengan mengubah-ubah luasan increment area kemudian di uji untuk mengikuti objek bergerak pada lintasan tertentu.
5. Pengujian perhitungan skala objek terahadap jarak nyata. Sehingga didapatkan data karakteristik perubahan jarak objek terhadap perubahan skala.

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan penelitian ini, didapatkan beberapa kesimpulan dari hasil percobaan sebagai berikut :

1. Metode *Mean-Shift* dapat diimplementasikan pada kamera pengawas semi otomatis dengan mengatur nilai variabel jumlah iterasi 2 - 4, ukuran *template* target model berbentuk persegi atau persegi panjang yang lebih condong panjang kearah horizontal, dan luasan *increment area* dengan perbesaran 2 – 4 kali.
2. Sistem object tracking dapat diterapkan pada kamera pengawas dengan menggunakan 2 buah motor penggerak. Pada penelitian ini, object tracking berhasil diterapkan dengan keberhasilan yang tinggi.

5.2 Saran

Dikarenakan pentingnya pengembangan dan penelitian lanjutan yang terkait dengan penelitian ini, penulis memberikan beberapa saran antara lain sebagai berikut :

1. Kamera webcam yang digunakan pada penelitian ini sangat mempengaruhi tingkat keberhasilan *tracking*. Pada penelitian ini masih digunakan kamera webcam dengan spesifikasi yang rendah sehingga penulis menyarankan pada penelitan selanjutnya menggunakan kamera dengan spesifikasi yang tinggi sehingga interferensi gambar dapat diminimalisir.
2. Pada penelitian terkait selanjutnya disarankan menggabungkan metode mean-shift dengan metode *tracking* lain demi mendapatkan waktu proses yang cepat, objek target yang variatif serta robust dalam segala kondisi.
3. Pada penelitan ini masih menggunakan webcam USB sebagai pengirim gambar ke komputer yang menyebabkan penempatan komputer harus disesuaikan dengan panjang kabel kamera. Pada penelitian selanjutnya dianjurkan menggunakan kamera *wireless* sehingga pemasangan kamera pengawas dapat lebih *portable*.

DAFTAR PUSTAKA

2017. *Arduino.* 3 6. https://www.arduino.cc/.

Adi, Kuncoro D, Lukas B Setyawan, dan F. Dalu Setiaji. 2013. Aplikasi Webcam Untuk Menjejak Pergerakan Manusia Didalam Ruangan. 51-56.

Agustina, Shanty Eka, dan Imam Mukhlash. 2012. Implementasi Metode Scale Invariant Feature Transform (SIFT) Dan Metode Continuosly Adaptive Mean-Shift (Camshift) Pada Penjejakan Objek Bergerak. 1-6.

Gonzalez, C. Rafael. 2002. Digital Image Processing edisi 2. Bab 10

Mahali, Muhammad Izuuddin, dan Agus Harjoko. 2014. Pelacakan Benda Bergerak Menggunakan Metode Mean Shift dengan Perubahan Skala dan Orientasi. 167-180.

Sanjaya, Mada. 2016. Panduan Praktis Membuat Robot Cerdas Menggunakan Arduino dan Matlab.

Setyawan, Sulfan Bagus, dan Djoko Purwanto. 2014. Penjejakan Objek Visual berbasis Algoritma Mean Shift dengan menggunakan kamera Pan-Tilt. 1-3.

Setyawan, Sulfan Bagus, Djoko Purwanto, dan Ronny Mardiyanto. 2015. Visual Object TrackingUsing Improved Mean Shift Algorithm.

Sutoyo, T., Edy Mulyanto, Vincent Suhartono, Oky Dwi Nurhayati, dan Wijanarto. 2009. Teori Pengolahan Citra Digital.

Watson, Karli. 2010. *Beginning Visual C# 2010.* Indianapolis: Wiley Publishing, Inc.

LAMPIRAN

LAMPIRAN A. *Listing* Program Arduino UNO R3

```
#include <Servo.h>
#include <SoftwareSerial.h>
int X = 320;
int Y = 240;
int nilaiServo=90;
Servo servoY;
double waktu;

void setup() {
  servoY.attach(3);
  pinMode(6,OUTPUT);          pinMode(7,OUTPUT);
  pinMode(4,INPUT_PULLUP);    pinMode(5,INPUT_PULLUP);
  pinMode(2,OUTPUT);          digitalWrite(2,LOW);
  Serial.begin(9600);
}

void loop() {
  if(Serial.available()){
    char terima = Serial.read();
    if      (terima=='X')X=Serial.parseInt();
    terima = Serial.read();
    if (terima=='Y'){
    Y=Serial.parseInt();Serial.flush();}
    waktu-=10;
  }
  gerakServo();
  gerakMotor();
}

void  gerakServo(){
  if(waktu+100>millis())goto akhir;
  waktu=millis();
  if(Y>290) {
```

```
    nilaiServo+=1;
    if(nilaiServo>180)nilaiServo=180;
  }
  else if (Y<190){
    nilaiServo-=1;
    if(nilaiServo<10)nilaiServo=10;
  }
  akhir:  servoY.write(nilaiServo);;
}
void  gerakMotor(){
  if(X<290){if(digitalRead(5)==LOW)goto pol;
  digitalWrite(6,HIGH);digitalWrite(7,LOW);}
  else if(X>350) { if( digitalRead(4) == LOW ) goto pol;
  digitalWrite(6,LOW);digitalWrite(7,HIGH);}
  else{ digitalWrite(6,HIGH); digitalWrite(7,HIGH);}
  goto lompat;
  pol:  digitalWrite(6,HIGH); digitalWrite(7,HIGH);
  lompat:;
}
```

LAMPIRAN B. *Listing* Program Visual C# 2010

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Structure;
using Emgu.CV.UI;
using Emgu.CV.Features2D;

using System.Diagnostics;
using System.Threading;

using System.IO.Ports;


namespace BelajarKamera
{
 public partial class Form1 : Form
 {
  Capture capturecam = null;
  Image<Bgr, byte>  imgSize = new Image<Bgr, byte>(640, 480);
  Image<Bgr, byte>  imgOrg = new Image<Bgr, byte>(320, 240);
  Image<Bgr, byte> imgROI = new Image<Bgr, byte>(320, 240);

  Image<Gray, byte> imgGray = new Image<Gray, byte>(320, 240);
  Image<Gray, byte> template = new Image<Gray, byte>(320, 240);
  Image<Gray, byte> templateBaru = new Image<Gray, byte>(320,
240);
  Image<Gray, byte> imgSmooth = new Image<Gray, byte>(320, 240);
```

```
  Image<Bgr, byte>    imgOrg1 = new Image<Bgr, byte>(320, 240);

  private Rectangle _cropRectangle,_cropRectangle1,batas ;
  private bool _isDraggring;
  private bool _adaObject;
  private bool _adaObjectFix;
  int xAwal, yAwal, xAkhir, yAkhir, lebar, tinggi;
  Point[] Max_Loc, Min_Loc;
  double[] min, max;
  Point[] Max_Loc1, Min_Loc1;
  double[] min1, max1;
  Point location0,tengah0,location1,tengah1,tengah2;
  Image<Gray, double> Result;
  Stopwatch stopwatch = new Stopwatch();
  String waktuProses,posisiTengah="X320Y240";
  double threshold = 0.7;
  int increment = 2,iterasi = 1,rWarna = 50;
  Bgr pixel; double r, g, b; Image<Bgr, byte> cnth = new
Image<Bgr, byte>(320, 240);
  Rectangle kotakAwal;
  double skala;
  int waktu,hitungan;

public Form1()
{
 InitializeComponent();
}
void loadProperty()
{
 string[] ports = SerialPort.GetPortNames();
 foreach (string port in ports)
 {
  comboBox1.Items.Add(port);
  comboBox1.SelectedIndex=0;
 }
 comboBox2.SelectedIndex = 4;
}
```

```
private void Form1_Load(object sender, EventArgs e)
{
 stopwatch.Start();
 loadProperty();
 try
 {
  capturecam = new Capture();
  capturecam.SetCaptureProperty(CAP_PROP.CV_CAP_PROP_FRAME_WIDTH,
640);
  capturecam.SetCaptureProperty(CAP_PROP.CV_CAP_PROP_FRAME_HEIGHT,
480);
 }
 catch (NullReferenceException exception)
 {
  return;
 }
 Application.Idle += new EventHandler(ProcessFunction);
}
void ProcessFunction(object sender, EventArgs e)
{
 timer1.Enabled = true;
}
private void pictureBox1_MouseDown(object sender, MouseEventArgs
e)
{
 timer1.Enabled = false;
 imgSize = capturecam.QueryFrame();
 imgOrg=imgSize.Resize(0.5, Emgu.CV.CvEnum.INTER.CV_INTER_LINEAR);
 pictureBox1.Image = imgSize.Bitmap;
 xAwal = e.X/2;
 yAwal = e.Y/2;
 _isDraggring = true;
}
private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
{
 isDraggring = false;
 cropos();
```

```
 Image<Bgr, byte> templateWarna=new Image<Bgr,byte>(lebar,tinggi);
 templateWarna.Bitmap = (Bitmap)pictureBox4.Image;
 template = templateWarna.Convert<Gray,byte>();
 cropRectangle = new Rectangle(0, 0, 0, 0);
 cropRectangle1 = new Rectangle(0, 0, 0, 0);
 updateSkala();
 timer1.Enabled = true;
}
private void pictureBox1_MouseMove(object sender, MouseEventArgs
e)
{
 if (!_isDraggring) return;
 xAkhir = e.X/2;
 yAkhir = e.Y/2;
 if (xAkhir < 1) xAkhir = 1; else if (xAkhir > 639) xAkhir = 639;
 if (yAkhir < 1) yAkhir = 1; else if (yAkhir > 479) yAkhir = 479;
 lebar = Math.Abs(xAkhir - xAwal);
 tinggi = Math.Abs(yAkhir - yAwal);
 _cropRectangle = new Rectangle(Math.Min(xAwal, xAkhir),
 Math.Min(yAwal, yAkhir), lebar, tinggi);
 _cropRectangle1 = new Rectangle(Math.Min(xAwal * 2, xAkhir * 2),
 Math.Min(yAwal * 2, yAkhir * 2), lebar * 2, tinggi * 2);
 location1 = new Point(Math.Min(xAwal, xAkhir), Math.Min(yAwal,
 yAkhir));
 pictureBox1.Invalidate();
 pictureBox5.Invalidate();
}
private void pictureBox1_Paint(object sender, PaintEventArgs e)
{
 e.Graphics.DrawRectangle(Pens.Red, _cropRectangle1);
}
private Image cropos()
{
 Bitmap gambarAsli = (Bitmap)pictureBox5.Image;
 Bitmap cropped = gambarAsli;
 try
 {
```

```
   cropped = gambarAsli.Clone(_cropRectangle,
   gambarAsli.PixelFormat);
   pictureBox4.Image = cropped;
  }
  catch
  {
  }
  return cropped;
}
private void timer1_Tick(object sender, EventArgs e)
{
  ambilGambar();
  //=======================
  if (button1.Text == "STOP")
  {
   meanShift();
  }
  //=======================
  tambahGaris();
  //=======================
  tampilkanGambar();
  //=======================
  if (serialPort1.IsOpen)
  {
   kirimSerial();
  }
  //=======================
  pengolahData();
}
private void ambilGambar()
{
  //==================AMBIL GAMBAR============================
  if (!_isDraggring)
  {
   imgSize = capturecam.QueryFrame();
   if (imgSize == null) return;
   imgOrg = imgSize.Resize(0.5,
   Emgu.CV.CvEnum.INTER.CV_INTER_LINEAR);
```

```
 }
 pictureBox5.Image = imgOrg.Bitmap;
 imgGray = imgOrg.Convert<Gray, byte>();
}
private void meanShift()
{
//=========================MEAN SHIFT==========================
 int tambahanLebar = (template.Width / 2) * (increment - 1);
 int tambahanTinggi = (template.Height / 2) * (increment - 1);
 batas = new Rectangle(location1.X - tambahanLebar - 5,
 location1.Y - tambahanTinggi - 5, template.Width * increment +
 10, template.Height * increment + 10);
 imgROI.Draw(new Rectangle(0, 0, 321, 241), new Bgr(255 - b, 255
 - g, 255 - r), -1);
 Bitmap gambarAsli = (Bitmap)pictureBox5.Image;
 imgOrg1.Bitmap = gambarAsli;
 imgROI.ROI = batas;
 imgOrg1.ROI = batas;
 imgOrg1.CopyTo(imgROI);
 imgOrg1.ROI = Rectangle.Empty;
 imgROI.ROI = Rectangle.Empty;
 pictureBox6.Image = imgROI.Bitmap;
 Image<Bgr, byte> gbr = new Image<Bgr, byte>(320, 240);
 gbr.Bitmap = (Bitmap)pictureBox6.Image;
 templateBaru = gbr.Convert<Gray, byte>();
 bool oke = false;

 Detect_Object(templateBaru, template);
 if (_adaObject)
 {
  location0 = new Point(Max_Loc[0].X, Max_Loc[0].Y);
  tengah0 = new Point(location0.X + template.Width / 2,
  location0.Y + template.Height / 2);
  {
   tengah1 = tengah0;
   location1 = location0;
   hitungan++;
   if (checkBox3.Checked == true && hitungan == 5)
```

```
    {
     updateTemplate();
     hitungan = 0;
    }
    oke = true;
   }
  }

  if (oke)
  {
   adaObjectFix = true;
  }
  else
  {
   _adaObjectFix = false;
   tengah1 = new Point(160, 120);
  }
  posisiTengah = "X" + tengah1.X * 2 + "Y" + tengah1.Y * 2;
 }
 private void Detect_Object(Image<Gray, byte> Area_Image,
 Image<Gray, byte> Image_Object)
 {
  adaObject = false;
  int i;
  for (i = 0; i < iterasi; i++)
  {
   using (Image<Gray, float> result_Matrix =
   Area_Image.MatchTemplate(Image_Object,
   TM_TYPE.CV_TM_CCOEFF_NORMED))
   {
    result_Matrix.MinMax(out min1, out max1, out Min_Loc1, out
    Max_Loc1);
    if (i == 0) max = max1;
    if (max1[0] > threshold)
    {
     if (max1[0] >= max[0])
     {
      max = max1; Max_Loc = Max_Loc1;
```

```
      _adaObject = true;
     }
    }
   }
  }
 }
private void updateTemplate()
{
 try
 {
  Bitmap gambarAsli = (Bitmap)pictureBox6.Image;
  Bitmap cropped = gambarAsli;
   _cropRectangle = new Rectangle(new Point(location1.X,
   location1.Y), new Size(template.Width, template.Height));
   cropped = gambarAsli.Clone ( _cropRectangle ,
   gambarAsli.PixelFormat);
  cropRectangle = new Rectangle(0, 0, 0, 0);
  cropRectangle1 = new Rectangle(0, 0, 0, 0);
                   //pictureBox4.Image = cropped;
  Image<Bgr, byte> templateWarna = new Image<Bgr,
byte>(template.Width,template.Height);
  templateWarna.Bitmap = cropped;//(Bitmap)pictureBox4.Image;

  cnth = templateWarna.Clone();//(Bitmap)pictureBox4.Image;
  pixel = cnth[new Point(template.Width/2,template.Height/2)];
  double R, G, B;
  R = pixel.Red;
  G = pixel.Green;
  B = pixel.Blue;

  if (R > r - rWarna && R < r + rWarna && G > g - rWarna && G < g
  + rWarna && B > b - rWarna && B < b + rWarna)
  {
   template = templateWarna.Convert<Gray, byte>();
   pictureBox4.Image = cropped;
   textBox9.Text = "UPDATE";
  }
  else
```

```
  {
   textBox9.Text = "GAK UPDATE";
  }
 }
 catch
 {
 }
}
private void tambahGaris()
{
//==================MENAMPILKAN GARIS=======================
  if (_adaObjectFix)
  {
   if (!_isDraggring)
   {
    imgOrg.Draw(new Rectangle(new Point(location1.X - 2,
    location1.Y - 2), new Size(template.Width + 4, template.Height
    + 4)), new Bgr(0, 0, 255), 1);
    imgOrg.Draw(new CircleF(tengah1, 1), new Bgr(0, 0, 255), -1);
    imgOrg.Draw(new LineSegment2D(tengah1, tengah2), new Bgr(255,
    0, 255), 2);
    tengah2 = tengah1;

    cnth.Bitmap = (Bitmap)pictureBox6.Image;
    pixel = cnth[tengah1];
    r = pixel.Red;
    g = pixel.Green;
    b = pixel.Blue;

    imgSmooth = cnth.InRange(new Bgr(b - rWarna, g - rWarna, r -
    rWarna), new Bgr(b + rWarna, g + rWarna, r + rWarna));

    imgSmooth = imgSmooth.SmoothMedian(17);
    if (checkBox1.Checked == false)
    {
     pictureBox4.Image = imgSmooth.Bitmap;
    }
```

```
    Rectangle kotak;
    for (Contour<Point> countours = imgSmooth.FindContours
    (CHAIN_APPROX_METHOD.CV_CHAIN_APPROX_SIMPLE,
    RETR_TYPE.CV_RETR_EXTERNAL); countours != null; countours =
    countours.HNext)
    {
     if (countours.InContour(tengah1) == 1)
     {
      kotak = countours.BoundingRectangle;
      imgOrg.Draw(kotak, new Bgr(0, 200, 0), 1);
      double lAwal, lAkhir;
      lAwal = kotakAwal.Width * kotakAwal.Height;
      lAkhir = kotak.Width * kotak.Height;
      skala = lAkhir / lAwal;
      textBox11.Text = Convert.ToString(lAwal) +
      Environment.NewLine + Convert.ToString(lAkhir) +
      Environment.NewLine +string.Format("{0:N2}",skala);
     }
    }
   }
  }
 }
private void tampilkanGambar()
{
 if (button1.Text == "STOP" && _isDraggring == false)
  imgOrg.Draw(batas, new Bgr(255, 0, 0), 1);
 pictureBox5.Image = imgOrg.Bitmap;
 imgSize = imgOrg.Resize(2, Emgu.CV.CvEnum.INTER.CV_INTER_LINEAR);
            pictureBox1.Image = imgSize.Bitmap;
}
private void updateSkala()
{

//============================================================
 int tambahanLebar = (template.Width / 2) * (increment - 1);
 int tambahanTinggi = (template.Height / 2) * (increment - 1);
```

```
 batas = new Rectangle(location1.X - tambahanLebar - 5,
location1.Y - tambahanTinggi - 5, template.Width * increment + 10,
template.Height * increment + 10);

 imgROI.Draw(new Rectangle(0, 0, 321, 241), new Bgr(0, 0, 0), -1);
 Bitmap gambarAsli = (Bitmap)pictureBox5.Image;
 imgOrg1.Bitmap = gambarAsli;
 imgROI.ROI = batas;
 imgOrg1.ROI = batas;
 imgOrg1.CopyTo(imgROI);
 imgOrg1.ROI = Rectangle.Empty;
 imgROI.ROI = Rectangle.Empty;
 pictureBox6.Image = imgROI.Bitmap;

 tengah1 = new Point(location1.X + template.Width / 2, location1.Y
+ template.Height / 2);
 cnth.Bitmap = (Bitmap)pictureBox6.Image;
 pixel = cnth[tengah1];
 r = pixel.Red; g = pixel.Green; b = pixel.Blue;
 imgSmooth = cnth.InRange(new Bgr(b - rWarna, g - rWarna, r -
rWarna), new Bgr(b + rWarna, g + rWarna, r + rWarna));

 imgSmooth = imgSmooth.SmoothMedian(17);
 if (checkBox1.Checked == false)
 {
  pictureBox4.Image = imgSmooth.Bitmap;
 }

 for (Contour<Point> countours = imgSmooth.FindContours
 (CHAIN_APPROX_METHOD.CV_CHAIN_APPROX_SIMPLE,
 RETR_TYPE.CV_RETR_EXTERNAL); countours != null; countours =
 countours.HNext)
 {
  if (countours.InContour(tengah1) == 1)
  {
   //imgOrg.Draw(countours, new Bgr(0, 200, 0), 1);
   kotakAwal = countours.BoundingRectangle;
   //textBox9.AppendText(Convert.ToString(kotakAwal.Size));
```

```
   }
  }

//=========================================================
}
private void kirimSerial()
{
 try
 {
  serialPort1.WriteLine(posisiTengah);
  textBox9.Text = posisiTengah;
 }
 catch (Exception Ex)
 {
 timer1.Enabled = false;
  MessageBox.Show(Ex.Message, "SERIAL BERMASALAH",
MessageBoxButtons.OK , MessageBoxIcon.Warning);
  Application.Exit();
 }
}
private void pengolahData()
{
 stopwatch.Stop();
 waktuProses = Convert.ToString(stopwatch.ElapsedMilliseconds);
 int progress; progress = (int)stopwatch.ElapsedMilliseconds;
 waktu = progress;
 if (progress > 1000) progress = 1000;
 progressBar1.Value = progress;
 if (checkBox2.Checked == true)
 {
 String[] line = { Convert.ToString(waktu) };
 System.IO.File.AppendAllLines( @Environment.GetFolderPath(
Environment.SpecialFolder.Desktop )+"\\waktuProses.txt",line);
            }
 stopwatch.Reset();
 stopwatch.Start();
 try
 {
```

```
  textBox1.Text = Convert.ToString(template.Size);
  textBox2.Text = waktuProses;
  textBox3.Text = Convert.ToString(max[0]);
  textBox4.Text = posisiTengah;
 }
 catch { }
}

//=====================PARAMETER=================================
private void hScrollBar1_Scroll(object sender, ScrollEventArgs e)
{
 iterasi = hScrollBar1.Value;
 textBox5.Text = Convert.ToString(hScrollBar1.Value);
}
private void hScrollBar2_Scroll(object sender, ScrollEventArgs e)
{
 threshold = hScrollBar2.Value;
 threshold = threshold / 10;
 textBox6.Text = Convert.ToString(threshold);
}
private void hScrollBar3_Scroll(object sender, ScrollEventArgs e)
{
 increment = hScrollBar3.Value;
 textBox7.Text = Convert.ToString(hScrollBar3.Value);
}
private void hScrollBar4_Scroll(object sender, ScrollEventArgs e)
{
 rWarna = hScrollBar4.Value;
 textBox10.Text = Convert.ToString(hScrollBar4.Value);
}
//==================TOMBOL START DAN SERIAL=================
private void button1_Click(object sender, EventArgs e)
{
 if (button1.Text == "STOP")
 {
  adaObjectFix = false;
  button1.Text = "START";
  button1.ForeColor = Color.Black;
```

```
  posisiTengah = "X320Y240";
  if (serialPort1.IsOpen) groupBox5.Enabled = true;
 }
 else
 {
  button1.Text = "STOP";
  button1.ForeColor = Color.Red;
  groupBox5.Enabled = false;
 }
}
private void button2_Click(object sender, EventArgs e)
{
 try
 {
  if (serialPort1.IsOpen)
  {
   serialPort1.Close();
   label10.Text = "TERPUTUS";
   label10.ForeColor = Color.Red;
   groupBox5.Enabled = false;
   button2.Text = "HUBUNGKAN";
  }
  else
  {
   serialPort1.BaudRate = Convert.ToInt16(comboBox2.Text);
   serialPort1.PortName = comboBox1.Text;
   serialPort1.Open();
   label10.Text = "TERHUBUNG";
   label10.ForeColor = Color.Black;
   if (button1.Text == "START") groupBox5.Enabled = true;
   button2.Text = "PUTUSKAN";
  }
 }
 catch (Exception Ex)
 {
  MessageBox.Show(Ex.Message, "SERIAL BERMASALAH",
   MessageBoxButtons.OK,MessageBoxIcon.Warning);
 }
```

```
}
//================KENDALI MANUAL==================
private void button3_MouseDown(object sender, MouseEventArgs e)
{
 posisiTengah = "X320Y0";//Target Di Atas
}
private void button3_MouseUp(object sender, MouseEventArgs e)
{
 posisiTengah = "X320Y240";//Target di Tengah
}
private void button4_MouseDown(object sender, MouseEventArgs e)
{
 posisiTengah = "X0Y240";//Target di Kiri
}
private void button4_MouseUp(object sender, MouseEventArgs e)
{
 posisiTengah = "X320Y240";//Target di Tengah
}
private void button5_MouseDown(object sender, MouseEventArgs e)
{
 posisiTengah = "X640Y240";//Target di Kanan
}
private void button5_MouseUp(object sender, MouseEventArgs e)
{
 posisiTengah = "X320Y240";//Target di Tengah
}
private void button6_MouseDown(object sender, MouseEventArgs e)
{
 posisiTengah = "X320Y480";//Target di Bawah
}
private void button6_MouseUp(object sender, MouseEventArgs e)
{
 posisiTengah = "X320Y240";//Target di Tengah
}
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
 if (checkBox1.Checked == true)
 {
```

```
  pictureBox5.Visible = false;
  pictureBox6.Visible = false;
 }
 else
 {
  pictureBox5.Visible = true;
  pictureBox6.Visible = true;
 }
}

private void checkBox2_CheckedChanged(object sender, EventArgs e)
{
 String[] line = { "======================[" +
 DateTime.Now.ToString("dd/MM/yyyy  HH:mm:ss") +
 "]======================" };

 System.IO.File.AppendAllLines(@Environment.GetFolderPath(Environ
 ment.SpecialFolder.Desktop) + "\\waktuProses.txt", line);
}
}
}
```