

BAB I PENDAHULUAN

Pada bab ini berisi latar belakang dari penulisan tugas akhir, perumusan masalah, tujuan dan manfaat penelitian, batasan masalah dan sistematika dari penulisan tugas akhir ini.

1.1 Latar belakang

Dengan kemajuan teknologi yang berkembang begitu cepat, banyak pekerjaan manusia yang harus dibantu oleh teknologi, salah satunya dalam hal aplikasi pelacakan obyek. Pelacakan obyek adalah proses menemukan obyek yang bergerak di urutan *frame* (SK Singh, 2015). Beberapa contoh kegunaan pelacakan obyek yaitu untuk digunakan bersama *CCTV* dalam pelacakan obyek mobil menggantikan alat *counter* manual untuk penghitungan mobil yang melintas di suatu kawasan, atau untuk digunakan sebagai *camera surveillance* yang mendeteksi orang yang masuk ke depan rumah menggantikan petugas keamanan yang berjaga.

Berbagai pelacakan obyek yang dimiliki oleh berbagai macam perangkat seperti kamera maupun *CCTV*, jarang yang menyertakan pergerakan kamera mengikuti obyek yang dilacak secara otomatis. Kamera yang banyak digunakan berupa kamera statis dan hanya bisa digerakan manual oleh manusia menggunakan sebuah *controller* melalui aplikasi web atau aplikasi khusus.

Untuk dapat melacak sebuah obyek menggunakan kamera, kamera perlu dilengkapi sebuah perangkat komputasi untuk mengolah citra didalam setiap *frame video*. Untuk tujuan ini, *embedded computer* banyak dipilih karena ukurannya yang jauh lebih kecil sehingga cocok untuk ditanam ke dalam alat seperti kamera. Selain itu, performa yang dapat dilakukan oleh *embedded computer* dapat diandalkan untuk memproses perhitungan yang cukup kompleks. *Embedded Computer* juga bersifat *portable* yaitu bisa dibawa kemana-mana dan menggunakan energi yang tidak besar untuk dapat beroperasi dengan baik.

Suatu *embedded computer* tetap mempunyai kemampuan komputasi dan kapasitas memori yang terbatas, sehingga algoritma yang digunakan untuk

mengolah pelacakan obyek harus disesuaikan agar tidak memberatkan *embedded computer* sehingga tidak menyebabkan *out of memory* ataupun *overheat*. Agar tidak memberatkan kinerja prosesor, maka juga diperlukan konfigurasi yang tepat sehingga akurasi pelacakan obyek tetap baik.

Terdapat beberapa metode deskripsi fitur (*Feature Descriptor*) untuk mendeteksi obyek, seperti SIFT (*Scale Invariant Feature Transformation*), SURF (*Speed Up Robust Feature*), *Color Based Object Detection*, HOG (*Histograms of Oriented Gradients*), Viola Jones, Optical Flow, dan lain-lain. Dalam kasus *camera surveillance* yang biasanya mendeteksi obyek bergerak secara *realtime*, algoritma yang banyak dipakai adalah *Histogram of Oriented Gradients*.

Histogram of Oriented Gradients adalah algoritma yang digunakan untuk melakukan ekstraksi fitur pada citra dengan menggunakan obyek manusia. Sedangkan *Support Vector Machine* adalah Teknik untuk mengklasifikasi data dengan cara memisahkan dua kelas yang berbeda berdasarkan (dalam kasus ini) tingkat gradien hasil dari deskripsi fitur HOG. Lalu garis hasil pemisah menggunakan SVM yang menentukan terdapat obyek di dalam frame video atau tidak.

Dalam penelitian ini, metode fitur ekstraksi yang digunakan adalah *Histograms of Oriented Gradients* (HOG) dengan memanfaatkan *Support Vector Machine* (SVM) sebagai klasifikasi fitur (*feature classifier*). HOG digunakan dalam penelitian ini dikarenakan HOG merupakan *feature descriptor* yang dapat mengambil tepi atau struktur gradien yang terkarakteristik dari suatu bentuk citra dengan distribusi intensitas gradien lokal yang baik. Dalam aplikasi pendeteksian obyek, kedua metode tersebut telah memiliki banyak pemanfaatan sistem seperti di bidang keamanan, *artificial intelligence*, pendataan secara otomatis, dan lain-lain (Ari Kurniawan, 2016).

1.2 Perumusan Masalah

Adapun rumusan masalah yang diangkat dalam penelitian tugas ini adalah “Bagaimana cara melakukan pelacakan obyek menggunakan *embedded computer* yang mempunyai memori yang terbatas namun dengan akurasi pelacakan yang cukup dan tidak memberatkan prosesor”.

Perumusan masalah tersebut dijabarkan dengan pertanyaan berikut.

1. Apakah algoritma pelacakan obyek yang sesuai untuk Raspberry Pi?
2. Bagaimana hasil penerapan pelacakan obyek di atas Raspberry Pi?
3. Bagaimana konfigurasi yang paling optimal untuk melakukan pelacakan obyek menggunakan Raspberry Pi?

1.3 Tujuan dan Manfaat Penelitian

Penyusunan tugas akhir ini memiliki tujuan dan manfaat untuk:

1. Menghasilkan sebuah purwarupa pelacakan obyek berbasis *Mini PC* yang mudah dipasang dan biaya terjangkau.
2. Menghasilkan sebuah aplikasi pelacakan obyek yang tidak memberatkan kerja prosesor namun dengan tingkat akurasi pelacakan obyek yang cukup baik.

1.4 Batasan Masalah

Batasan masalah pada Tugas Akhir ini yaitu:

1. Pembuatan aplikasi pelacakan obyek dibuat menggunakan OpenCV dengan Bahasa pemrograman Python.
2. Perangkat pemindai yang digunakan dalam Tugas Akhir ini adalah sebuah *webcam* dengan resolusi 1.3 mega piksel.
3. Jarak antara perangkat pemindai dengan obyek yang dilacak minimal 4 meter dan maksimal 6 meter agar obyek dapat dilacak.
4. *Servo* yang digunakan dalam Tugas Akhir ini memiliki jarak putaran dari 0° hingga 180°.
5. Obyek yang dilacak dalam Tugas Akhir ini adalah tubuh manusia.
6. Jumlah obyek manusia yang dapat dideteksi bisa lebih dari satu namun yang dapat diikuti hanya satu.
7. Percobaan yang dilakukan dalam penulisan ini menggunakan penerangan lampu didalam ruangan.

1.5 Sistematika Penulisan

Semua kegiatan yang mendukung Tugas Akhir ditulis dengan sistematika dengan cara seperti berikut.

Bab I: Pendahuluan

Bab ini berisi latar belakang dari penulisan proposal tugas akhir, perumusan masalah, tujuan dan manfaat penelitian, Batasan masalah, dan sistematika dari penulisan proposal Tugas Akhir.

Bab II: Landasan Teori

Bab ini berisikan mengenai pembahasan teori algoritma *Histogram of Oriented Gradients*, teori algoritma *Support Vector Machine*, teori pengenalan tentang *Mini PC Raspberry Pi* dan *Arduino*, serta pembahasan mengenai *OpenCV*.

Bab III: Metodologi Penelitian

Bab ini berisi tentang tahapan yang dilakukan dalam penelitian diantaranya tahap cara kerja algoritma *Histogram of Oriented Gradients* dengan bagaimana algoritma tersebut dapat melacak obyek. Tahap cara kerja perangkat pemindai mengirim citra ke *Raspberry Pi* yang lalu diolah dan hasilnya digunakan untuk mengubah sudut alat pemindai, rancangan analisis dan pengumpulan data.

Bab IV: Implementasi dan Pengujian Sistem

Bab ini menjelaskan tentang implementasi sistem pendeteksian obyek menggunakan metode *Histogram of Oriented Gradients* dan *Support Vector Machine* seperti merubah ukuran citra yang ditangkap oleh kamera *webcam*, membuat penanda garis putih untuk batas pergerakan kamera, mencari nilai *gradient* dan citra yang ditangkap, mencari obyek tubuh manusia didalam citra tersebut, memberi tanda berupa persegi panjang disekeliling obyek yang dideteksi sebagai tubuh manusia dan tanda ditengah obyek sebagai acuan pergerakan kamera jika melewati batas putih. Pengujian yang dilakukan berupa skenario pengujian ketika merekam aktifitas seseorang berjalan melewati ruangan yang

dipasang kamera *webcam* tersebut, parameter pengujian yaitu berupa jarak kamera dengan obyek dan intensitas cahaya yang ada ketika proses pengujian.

Bab V: Kesimpulan dan Saran

Bab ini berisi kesimpulan dari pengujian yang telah dilaksanakan beserta saran untuk penelitian selanjutnya dengan identifikasi kekurangan pada sistem yang dibuat agar bisa menjadi penelitiannya bisa lebih baik.



BAB II

LANDASAN TEORI

Dalam bab ini akan dijelaskan mengenai dasar teknologi yang digunakan, serta penelitian yang terkait dengan judul tugas akhir.

2.1 Tinjauan Pustaka

2.1.1 Pengertian Pelacakan Obyek

Mengidentifikasi suatu obyek bergerak dari urutan *frame* video adalah tugas paling mendasar dan penting dalam pembuatan aplikasi komputer (Vinary, 2015). Pelacakan obyek adalah proses menemukan obyek yang bergerak di urutan-urutan *frame* (R. Revathi, 2012). Tujuan dari pelacakan obyek adalah untuk mengasosiasikan obyek target dalam *frame* video berturut-turut (SK Singh, 2015). Pelacakan obyek dapat dilakukan dengan menggunakan ekstraksi fitur dari obyek dan mendeteksi benda-benda di urutan *frame*.

Proses menemukan obyek yang bergerak dalam bentuk waktu disebut juga dengan pelacakan video. Pelacakan video menggabungkan obyek dalam *frame* video berturut-turut. Pelacakan obyek digunakan untuk memperkirakan lokasi, kecepatan dan parameter jarak obyek bergerak dengan bantuan kamera statis. Sistem pelacakan obyek membutuhkan segmentasi yang akurat dari obyek latar belakang untuk pelacakan efektif.

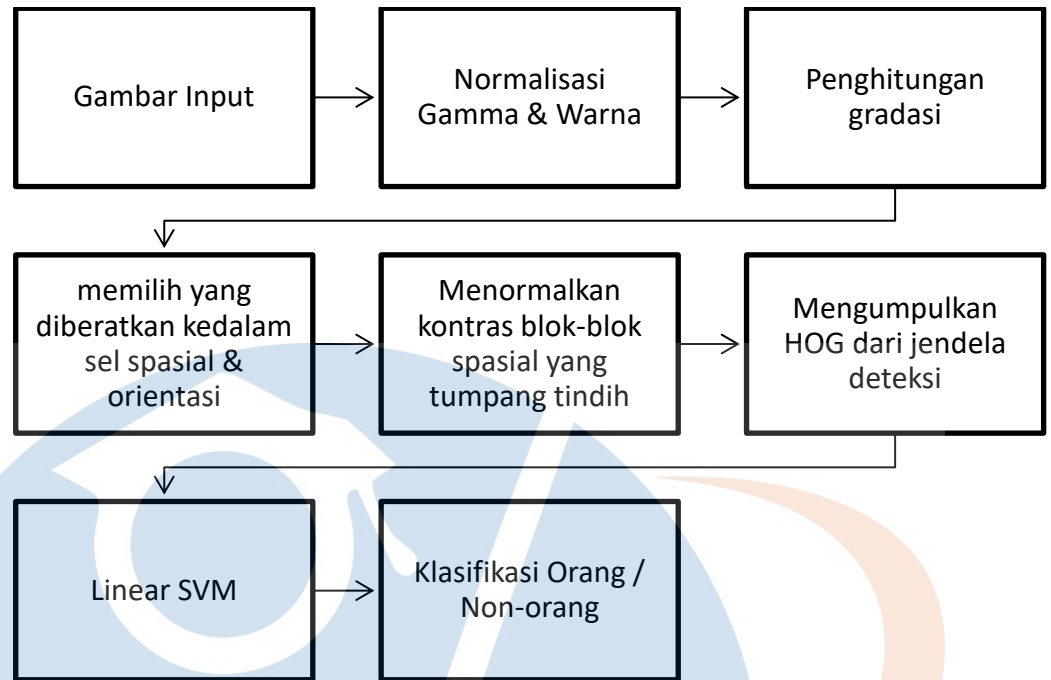
2.1.2 Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) adalah fitur deskriptif yang digunakan pada pengolahan citra untuk tujuan pendeteksian obyek. Ide dasarnya adalah penampilan dan bentuk benda lokal seringkali bisa dicirikan lebih baik dengan distribusi gradien intensitas lokal atau arah tepi, bahkan tanpa pengetahuan yang tepat tentang posisi gradien atau tepi yang sesuai. Dalam prakteknya hal ini diimplementasikan dengan membagi jendela citra menjadi spasial kecil daerah yang disebut sel. Setiap sel yang mengumpulkan histogram 1-D lokal arah gradien atau orientasi tepi di atas piksel sel.

Berdasarkan langkahnya pada Gambar 2.1, proses awal pada metode HOG adalah melakukan konversi citra RGB (*Red, Green, Blue*) menjadi *grayscale*, yang kemudian dilanjutkan dengan menghitung nilai gradien setiap piksel. Setelah mendapatkan nilai gradien, maka proses selanjutnya yaitu menentukan jumlah bin orientasi yang akan digunakan dalam pembuatan histogram. Proses ini disebut *spatial orientation binning*. Namun sebelumnya, pada proses *gradient compute* citra dibagi menjadi beberapa sel dan dikelompokkan menjadi ukuran lebih besar yang dinamakan blok. Sedangkan pada proses normalisasi, blok digunakan perhitungan geometri HOG. Proses ini dilakukan karena terdapat blok yang saling tumpang tindih. Berbeda dengan proses pembuatan histogram citra yang menggunakan nilai – nilai intensitas piksel dari suatu citra atau bagian tertentu dari citra untuk pembuatan histogramnya (Dalal, 2003).

Entri histogram gabungan membentuk data perwakilan yang berguna untuk melakukan variasi penerangan, bayangan, serta mengubah kontras lokal tanggapan sebelum menggunakannya. Hal ini bisa dilakukan dengan mengumpulkan ukuran energi histogram lokal dari daerah yang lebih luas (blok) dan menggunakan hasilnya untuk menormalkan semua sel di blok. Tugas akhir ini akan mengacu pada normalisasi blok sebagai *Histogram of Gradients* (HOG) deskriptor. Cara melakukannya adalah dengan menggeser jendela deteksi dengan HOG deskriptor dan menggunakan fitur gabungan dalam jendela SVM konvensional berbasis klasifikasi yang memberikan rantai deteksi manusia (Dalal, 2003).

STT - NF



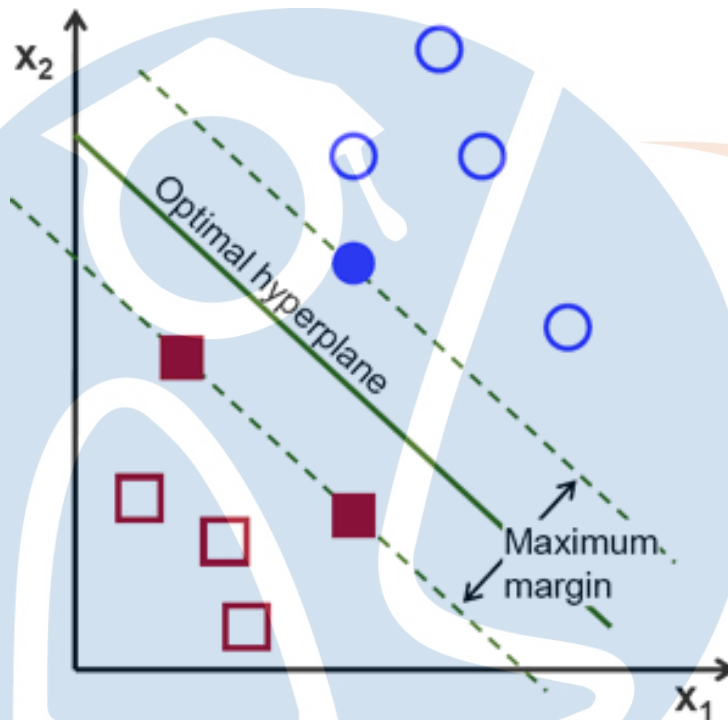
Gambar 2.1 Ekstraksi fitur dan rantai deteksi obyek (Dalal,2003)

Representasi HOG memiliki beberapa keunggulan. Metode ini akan menangkap tepi atau struktur gradien yang sangat khas bentuk lokal, dan hal itu terjadi pada representasi lokal tingkat invarian yang mudah dikendalikan ke geometris lokal dan transformasi fotometrik. Translasi atau rotasi akan membuat sedikit perbedaan jika mereka jauh lebih kecil dari ukuran lokal bin spasial atau orientasi. Untuk deteksi manusia, spasial *sampling* yang agak kasar, orientasi *sampling* halus dan kuat normalisasi fotometrik lokal dapat menjadi strategi terbaik karena memungkinkan anggota badan dan segmen tubuh untuk mengubah penampilan dan berpindah dari sisi ke sisi cukup banyak asalkan mereka mempertahankan orientasi tegak lurus (Dalal, 2003).

2.1.3 Support Vector Machines

Support Vectore Machine (SVM) adalah suatu mesin pembelajaran yang baru untuk masalah klasifikasi dua kelompok. Mesin ini secara konseptual melakukan beberapa tahapan berikut. Masukan vektor dipetakan secara non-linier ke fitur berdimensi ruang yang sangat tinggi. Di ruang fitur ini permukaan keputusan linier dibangun seperti terlihat pada Gambar 2.2. Sifat khusus dari

permukaan keputusan memastikan kemampuan generalisasi mesin pembelajaran yang tinggi. Gagasan SVM sebelumnya diimplementasikan untuk kasus terbatas dimana data pelatihan dapat dipisahkan tanpa kesalahan. Penelitian berikutnya telah memperluas hasil ke data pelatihan yang tidak dapat dipisahkan (Corina,1995).



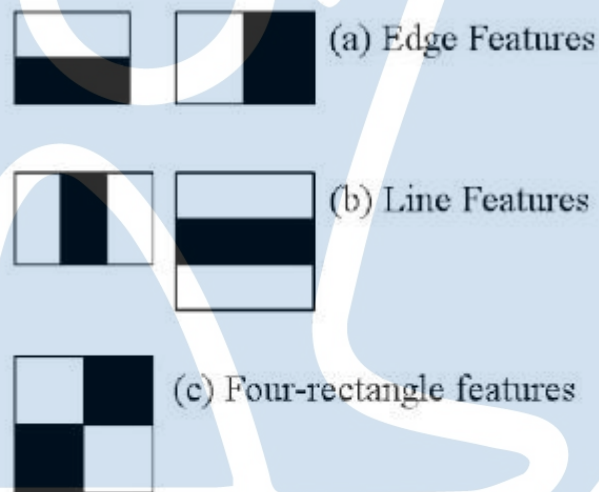
Gambar 2.2 Optimal Hyperlane (Docs OpenCV – SVM Introduction, 2018)

Kemampuan generalisasi yang tinggi dari SVM yang menggunakan transformasi masukan polinom. Hasil ini juga dapat dibandingkan kinerjanya ke berbagai algoritma pembelajaran klasik yang semuanya mengambil bagian dalam studi *benchmark Optical Character Recognition* (Corina,1995).

Prinsip dasar SVM adalah *linear classifier*, dan selanjutnya dikembangkan agar dapat bekerja pada *problem non-linear*, dengan memasukan konsep *kernel trick* pada ruang kerja berdimensi tinggi (S.R Gun, 1998). SVM juga dikenal sebagai teknik *machine learning* paling mutakhir setelah algoritma *Neural Network* (NN). Baik SVM maupun NN tersebut telah berhasil digunakan dalam aplikasi pengenalan pola (S.R Gun, 1998).

2.1.4 Haar Cascade Classifier

Haar Cascade Classifier adalah sebuah algoritma pendeteksian obyek yang menggunakan pendekatan *machine learning*. Algoritma ini paling sering digunakan untuk mendeteksi objek yang tidak terlalu banyak perubahan bentuk seperti wajah atau benda mati lainnya seperti bola, botol, dan jam tangan. *Classifier* yang digunakan oleh *Haar Cascade Classifier* adalah *Haar Feature*. *Haar Feature* ini berupa kernel untuk mengenali fitur di dalam sebuah citra seperti pada Gambar 2.3.



Gambar 2.3 Beberapa *Haar Feature* dasar (Docs OpenCV – Face Detection, 2018)

Terdapat tiga poin penting dalam algoritma *Haar Cascade*, yang pertama yaitu *Haar Cascade* menggunakan representasi citra bernama citra integral yang dapat membuat evaluasi citra lebih cepat. Citra integral dapat dihitung dari citra asal menggunakan beberapa operasi per piksel. Setelah dihitung, salah satu fitur Haar ini dapat dihitung dalam skala apa pun. Poin kedua yaitu metode untuk membangun penggolongan dengan memilih sejumlah kecil fitur penting menggunakan AdaBoost. Dalam citra apapun, *sub-window* jumlah total fitur *Haar* sangat besar, jauh lebih besar dari jumlah piksel. Untuk memastikan klasifikasi cepat, proses pembelajaran harus mengabaikan sebagian besar fitur yang tersedia, dan fokus pada sekumpulan kecil fitur penting (Viola, 2001). Poin ketiga adalah *Cascade* yaitu memberikan beberapa *Haar feature* di window berukuran 24x24

piksel. Pemberian *Haar feature* dibagi menjadi beberapa tahap. Jika di tahap awal hasil *Haar feature* tidak menunjukkan hasil yang signifikan, maka pemberian fitur akan dihentikan dan akan bergeser ke *window* berikutnya.

2.1.5 OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah perangkat lunak pustaka untuk pengolahan citra yang dirilis dengan lisensi BSD dan oleh karena itu bersifat gratis untuk penggunaan akademis dan komersial. OpenCV memiliki antarmuka C++, C, Python dan Java dan mendukung Windows, Linux, Mac OS, iOS dan Android. OpenCV dirancang untuk efisiensi komputasi dan dengan fokus yang kuat pada aplikasi *real-time*. Ditulis di C / C++ yang dioptimalkan dan dapat memanfaatkan pemrosesan *multi-core*. Diaktifkan dengan OpenCL, ia juga dapat memanfaatkan akselerasi perangkat keras dari platform penghitungan heterogen yang mendasarinya (OpenCV, 2017).

Aplikasi yang didukung OpenCV adalah sebagai berikut:

1. 2D dan 3D alat fitur.
2. Perkiraan *egomotion*.
3. Sistem pengenalan wajah.
4. Sistem pengenalan *gesture*.
5. Interaksi manusia dan komputer (HCI).
6. *Mobile robotics*.
7. *Motion understanding*.
8. Identifikasi obyek.
9. Pengenalan dan segmentasi.
10. *Stereopsis stereo vision: depth perception from 2 cameras*.
11. *Structure from motion (SFM)*.
12. *Motion Tracking*.
13. *Augmented Reality*.

Untuk mendukung aplikasi di atas, OpenCV memasukan pustaka *machine learning* statistik, yang berisi:

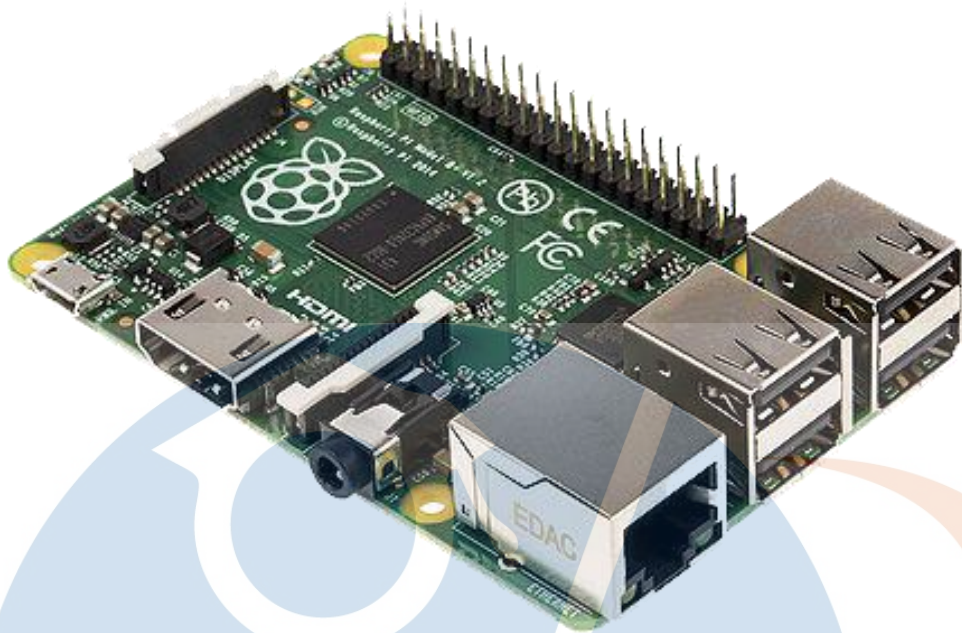
1. *Boosting.*
2. *Decision tree learning.*
3. *Gradient boosting tree.*
4. *Expectation-maximization algorithm.*
5. *K-nearest neighbor algoritm.*
6. *Naïve bayes classifier.*
7. *Artificial neural network.*
8. *Random forest.*
9. *Support vector machine(SVM).*
10. *Deep neural network(DNN).*

OpenCV juga telah didukung berbagai macam *operating sistem* seperti: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD, Android, iOS, Maemo, Blackberry 10 (OpenCV Wikipedia, 2018).

2.1.6 Raspberry Pi

Raspberry Pi adalah suatu perangkat komputer papan tunggal berukuran sebesar kartu kredit yang dikembangkan di UK oleh Raspberry Pi *Foundation* dengan tujuan mempromosikan mengajarkan komputer sains dasar. Mereka mengembangkan sumber gratis untuk membantu orang belajar tentang komputasi dan bagaimana cara membuat sesuatu dengan komputer (Steve, 2011). Bentuk papan Raspberry Pi dapat dilihat pada Gambar 2.4.

Raspberry Pi yang digunakan dalam tugas akhir ini yaitu Raspberry Pi 3 model B, yang memiliki prosesor Broadcom BCM2837 64bit Quad Core CPU 1.2Ghz, dengan RAM berukuran 1 Gigabyte dan media penyimpanan berupa SD Card dan berjalan di atas *operating system* Linux.



Gambar 2.4 Raspberry Pi 3 Model B (Raspberry Pi, 2018)



Alternate Function					Alternate Function
	3.3V PWR	1		2	5V PWR
I2C1 SDA	GPIO 2	3		4	5V PWR
I2C1 SCL	GPIO 3	5		6	GND
	GPIO 4	7		8	UART0 TX
	GND	9		10	UART0 RX
	GPIO 17	11		12	GPIO 18
	GPIO 27	13		14	GND
	GPIO 22	15		16	GPIO 23
	3.3V PWR	17		18	GPIO 24
SPI0 MOSI	GPIO 10	19		20	GND
SPI0 MISO	GPIO 9	21		22	GPIO 25
SPI0 SCLK	GPIO 11	23		24	GPIO 8
	GND	25		26	GPIO 7
	Reserved	27		28	Reserved
	GPIO 5	29		30	GND
	GPIO 6	31		32	GPIO 12
	GPIO 13	33		34	GND
SPI1 MISO	GPIO 19	35		36	GPIO 16
	GPIO 26	37		38	GPIO 20
	GND	39		40	GPIO 21
					SPI0 CS0
					SPI0 CS1
					SPI1 CS0
					SPI1 MOSI
					SPI1 SCLK

Gambar 2.5 Pinout GPIO Raspberry Pi 3 (Docs Microsoft – Raspberry Pi 2 & 3 Pin Mapping, 2018)

Raspberry Pi memiliki *General Purpose Input Output* (GPIO), yaitu pin di papan komputer yang perilakunya dapat dikendalikan termasuk dalam hal sifat pin tersebut menjadi masukan / keluaran. GPIO dikendalikan oleh pengguna secara *real-time* melalui program. Bentuk susunan fisik pin seperti terlihat pada Gambar 2.5. Beberapa ciri GPIO adalah sebagai berikut.

a. Penggunaan

Manufacture banyak menggunakan GPIO untuk hubungan ke perangkat lain, contohnya adalah *Integrated circuit* seperti *System-on-chip, embedded and custom hardware, programmable logic devices*.

b. Kapabilitas

- GPIO pin dapat dikonfigurasi sebagai masukan atau keluaran.
- GPIO pin dapat diaktifkan atau dinonaktifkan.
- Nilai masukan dapat dibaca (biasanya *High / Low*).
- Nilai keluaran dapat dibaca maupun ditulis.

c. Pengelompokan sebagai Port

Port adalah sekelompok pin GPIO (biasanya 8 pin GPIO) yang disusun dalam kelompok dan dikontrol sebagai sebuah kelompok.

2.1.7 Arduino Uno

Arduino Uno (Gambar 2.6) adalah papan *microcontroller* dengan IC (*Integrated Circuits*) ATmega328P. Papan ini memiliki 14 *digital input/output* (6 diantaranya dapat digunakan untuk *PWM outputs*), 6 *analog inputs*, kristal 16 Mhz, koneksi USB, catu daya, *ICSP Header* dan tombol reset. Papan ini telah berisi semua yang dibutuhkan untuk mendukung mikrokontroler. Pengguna cukup menghubungkan ke komputer dengan kabel USB atau menyalakan dengan adaptor AC-ke-DC atau baterai untuk memulai. Pengguna bisa bermain-main dengan Arduino Uno tanpa terlalu memikirkan melakukan sesuatu yang salah. Jika terjadi kesalahan, skenario terburuk adalah cukup mengganti chip itu seharga beberapa dolar dan memulai lagi dari awal (Arduino, 2017).

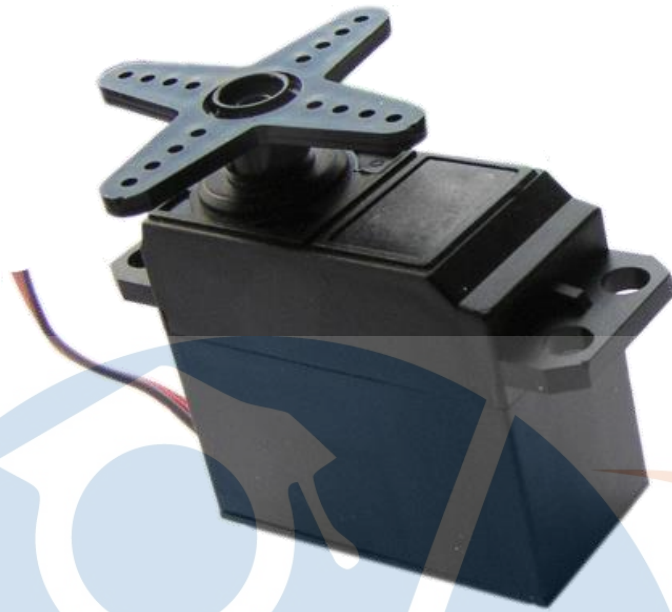


Gambar 2.6 Arduino Uno (Arduino Uno, 2017)

“Uno” berarti satu dalam Bahasa Italia dan dipilih untuk menandakan rilisnya *Arduino Software Integrated Development Environment (IDE) 1.0*. Papan Uno dan *Arduino IDE 1.0* adalah *tools* referensi utama dari Arduino, yang terus berkembang hingga beberapa rilis. Hingga saat ini, Arduino Uno menjadi model referensi utama untuk *platform* Arduino.

2.1.8 Motor Servo

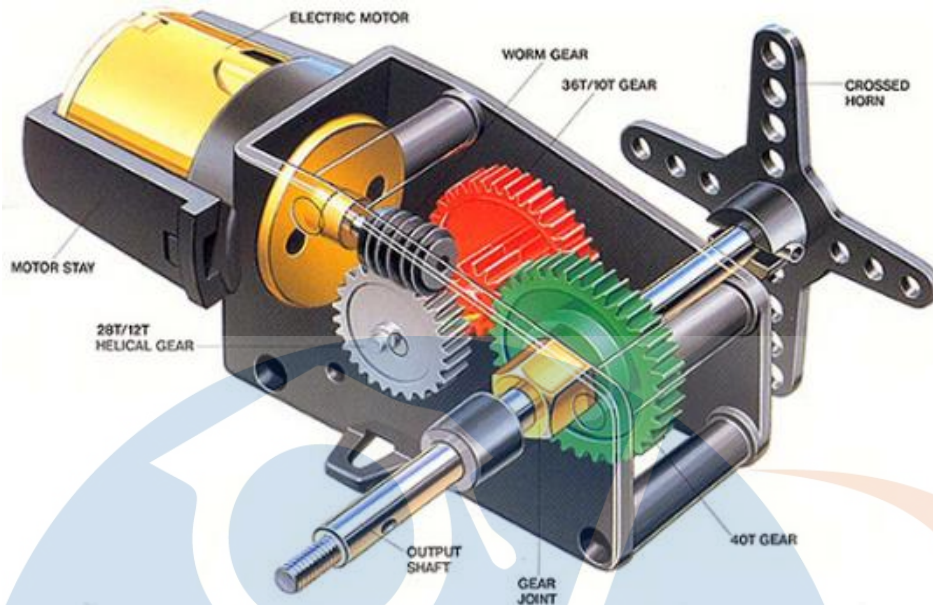
Motor servo (Gambar 2.7) adalah sebuah motor DC yang dilengkapi rangkaian kendali dengan sistem *closed feedback* yang terintegrasi dalam motor tersebut. Pada motor servo posisi putaran sumbu (*axis*) dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo.



Gambar 2.7 Servo Motor (Electrosome, 2018)

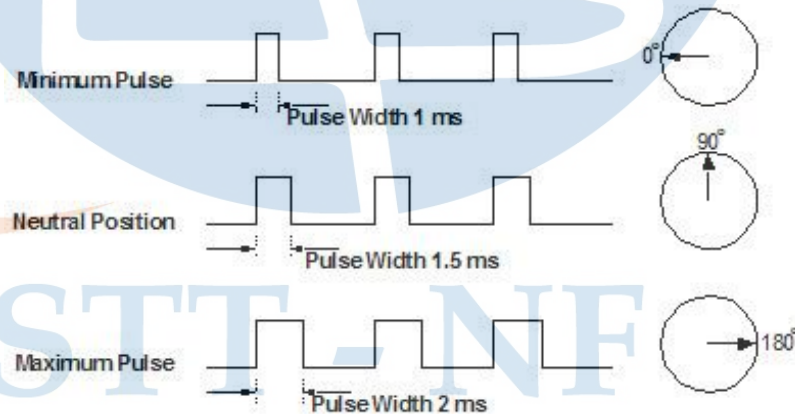
Rangkaian motor servo (Gambar 2.8) disusun dari sebuah motor DC, gearbox, variabel resistor (VR) atau potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas maksimum putaran sumbu (axis) motor servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang pada pin kontrol motor servo.

STT - NF



Gambar 2.8 Rangkaian Servo Motor (Elprocus, 2018)

Pengaturan sudut servo diatur oleh *Pulse Width Modulation* (PWM). Pulsa PWM (Gambar 2.9) secara umum adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam suatu periode, untuk mendapatkan tegangan rata-rata yang berbeda (Rudito Prayogo, 2012).



Gambar 2.9 Duty Cycle pada PWM (Servo City, 2018)

Servo akan menerima gelombang pulsa secara periodik. Panjangnya gelombang pulsa menentukan seberapa jauh motor akan berputar. Sebagai contoh, gelombang pulsa 1,5 ms akan membuat servo bergerak 90°. Jadi jika semakin rapat gelombang pulsa yang diterima oleh servo motor maka akan semakin tinggi

derajat putarannya. Maksimal derajat putaran adalah 180° atau dengan panjang gelombang pulsa 2 ms.

2.2 Penelitian Terkait

Berikut ini Penulis menjabarkan beberapa penelitian terkait dan posisi penelitian yang dilakukan oleh penulis, sebagaimana dirangkum pada Tabel 2.1:

1. Penelitian yang dilakukan oleh Choirul Umul Islami, 2011, “Rancang Bangun Sistem Pelacakan Obyek Menggunakan CCTV dan Webcam”, disimpulkan bahwa pelacakan obyek dengan menggunakan Kamera bermotor digerakan oleh mikrokontroller dan disambungkan ke PC melalui jalur RS 232 dengan tujuan akurasi pelacakan terhadap satu obyek lebih tinggi.
2. Penelitian yang dilakukan oleh Ari Kurniawan Saputra, 2016, “Aplikasi Deteksi Obyek Menggunakan *Histogram of Oriented Gradient* Untuk Modul Sistem Cerdas Pada Robot Nao”, disimpulkan bahwa dengan menggunakan *Histogram of Oriented Gradient* untuk melacak sebuah obyek bisa menghasilkan akurasi 99,10% pada modul sistem cerdas Robot NAO.
3. Penelitian yang dilakukan oleh Haryansyah, 2015, “Deteksi dan penghitungan manusia pada video pengunjung intansi pemerintah di Tarakan menggunakan metode histogram of oriented gradients.”, disimpulkan bahwa *Histogram of Oriented Gradients* bisa digunakan untuk menghitung jumlah pengunjung disuatu intansi pemerintahan.
4. Penelitian yang dilakukan Penulis, yaitu “Penerapan Pelacakan Obyek Dengan Kamera Berbasis Raspberry Pi Dengan *Histogram Oriented Gradients* dan *Support Vector Machine*”, meneliti apakah pelacakan obyek manusia dengan algoritma *Histogram of Oriented Gradients* memiliki akurasi yang baik dan dapat dijalankan di atas Raspberry Pi.

Tabel 2.1 Penelitian Terkait dan Penelitian yang Sedang Dilakukan

No.	Judul Penelitian	Tahun	Kesimpulan
1.	Rancang Bangun Sistem Pelacakan Obyek Menggunakan CCTV dan Webcam (Choirul, 2011)	2011	<p>Sistem pelacakan obyek merupakan bagian dari sistem kontrol dan keamanan. Dalam pemanfaatannya, sistem Pelacakan obyek sering digunakan dalam bidang robotika, game, maupun keamanan gedung. Dalam bidang keamanan, pelacakan obyek banyak digunakan untuk sistem pengamanan dalam sebuah gedung seperti rumah, museum, maupun tempat - tempat penting lainnya.</p> <p>Pelacakan obyek dengan menggunakan kamera bermotor digerakan oleh mikrokontroller dan disambungkan ke PC melalui jalur RS 232 dengan tujuan akurasi pelacakan terhadap satu obyek lebih tinggi.</p>
2.	Deteksi dan Penghitungan Manusia Pada Video Pengunjung Intansi Pemerintah Di Tarakan Menggunakan Metode Histogram of Oriented Gradients (Haryansyah, 2015).	2015	<p>Pelacakan obyek manusia menggunakan algoritma <i>Histogram of Oriented Gradients</i> bisa digunakan untuk menghitung jumlah pengunjung disuatu intasi pemerintahan.</p> <p>Tentunya hal ini sangat berguna apabila pihak instansi ingin mengetahui seberapa besar tingkat kepadatan pengunjung setiap harinya yang tidak memungkinkan untuk dilakukan perhitungan secara manual karena manusia yang terus bergerak.</p>

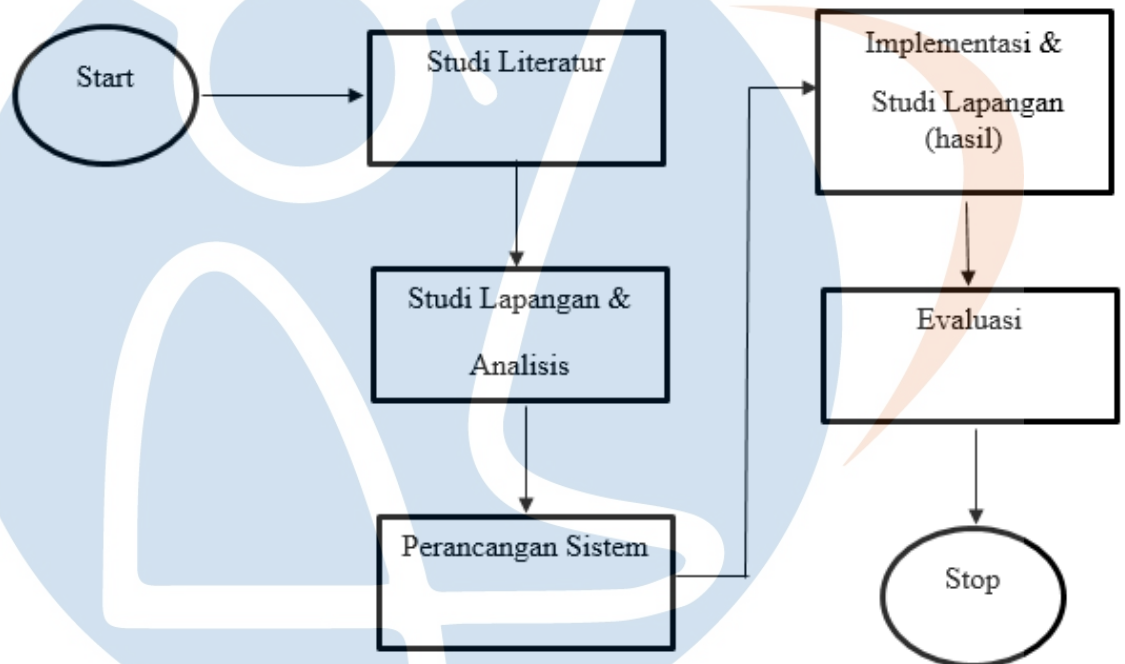
3.	Aplikasi Deteksi Obyek Menggunakan Histogram Of Oriented Gradient Untuk Modul Sistem Cerdas Pada Robot Nao (Ari Kurniawan, 2016).	2016	<p>Tujuan dari penelitian tugas akhir ini adalah mengembangkan modul sistem cerdas pada Robot NAO dengan membangun aplikasi yang dapat mendeteksi obyek di sekitar</p> <p>Dengan menggunakan Histogram Of Oriented Gradient untuk melacak sebuah obyek bisa menghasilkan akurasi 99,10% pada modul sistem cerdas Robot NAO.</p>
4	Penerapan Pelacakan Obyek Dengan Kamera Berbasis Raspberry Pi Dengan Histogram Oriented Gradients dan Support Vector Machine (Adrianto Prasetyo, 2018)	2018	Pelacakan Obyek dapat diproses hanya menggunakan mini komputer Raspberry Pi dengan kemampuan hampir sama dengan komputer pada umumnya ditambah servo motor yang membuat pelacakan obyek semakin mudah.

STT - NF

BAB III METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian yang dilakukan dalam tugas akhir ini dibagi menjadi 5 tahapan yaitu studi literatur, studi lapangan dan analisis kebutuhan, perancangan sistem, implementasi serta hasil studi lapangan, dan yang terakhir adalah evaluasi. Alur tahapan metodologi penelitian ini dijelaskan pada Gambar 3.1.



Gambar 3.1 Alur tahapan metodologi penelitian

3.1.1 Studi Literatur

Studi literatur dikerjakan di awal penyusunan dengan cara membaca jurnal, paper, serta *website* untuk mengkaji lebih dalam mengenai *Histogram of Oriented Gradient*, *Support Vector Machine*, *OpenCV*, *Raspberry Pi*, *Arduino Uno*, *Servo Motor*. Dengan output dari metode *Waterfall* adalah referensi dan *knowledge* mengenai *Histogram of Oriented Gradient*, *Support Vector Machine*, *OpenCV*, *Raspberry Pi*, *Arduino Uno*, *Servo Motor*. Adapun daftar referensi dari studi literatur ini dapat dilihat di daftar pustaka tugas akhir ini.

Output:

Referensi dan *knowledge* mengenai *Histogram of Oriented Gradient*, *Support Vector Machine*, OpenCV, Raspberry Pi, Arduino Uno, Servo Motor.

3.1.2 Studi Lapangan dan Analisis Kebutuhan

Menganalisis kebutuhan dari tugas akhir ini dengan cara menyimpulkan kegiatan studi literatur yang relevan dengan kebutuhan *hardware* dan *software* termasuk lokasi dari perangkat pelacakan obyek ini dipasang. Analisis kebutuhan sistem yang dirancang berdasarkan metode pengumpulan data yang telah dilakukan. Tahapan ini meliputi:

- Identifikasi masalah: Peneliti mendefinisikan masalah dari hasil metode pengumpulan data yang telah dilakukan.
- Identifikasi kebutuhan: Peneliti mempertimbangkan kebutuhan *hardware* dan *software* yang digunakan dalam perancangan aplikasi pelacakan obyek ini.

Output:

Menyimpulkan kegiatan studi literatur yang relevan dengan kebutuhan *hardware* dan *software* serta lokasi dari perangkat pelacakan obyek ini dipasang.

3.1.3 Perancangan Sistem

Pada tahap ini pengembangan aplikasi dilakukan dengan cara menulis program sesuai alur program dari ditangkapnya citra oleh kamera *webcam* lalu diolah di Raspberry Pi melalui OpenCV dengan metode *Histogram of Oriented Gradient* dan *Support Vector Machine*, dan dikirim ke Arduino yang akan menggerakkan servo motor.

Output:

Pengembangan aplikasi dilakukan dengan cara menulis program sesuai alur program dari awal sampai akhir.

3.1.4 Simulasi dan Studi Lapangan

Melakukan uji coba pemasangan perangkat di lokasi yang ditentukan, dan menjalankan aplikasi pelacakan obyek secara *real-time*.

Output:

Simulasi: Melakukan uji coba yang bertujuan melihat apakah pelacakan obyek sudah dapat menjawab permasalahan dan tujuan penelitian.

Studi Lapangan: Melihat hasil pengolahan pelacakan obyek dari segi akurasi dan kinerja aplikasi ini bekerja.

3.1.5 Evaluasi

Melakukan evaluasi untuk mengidentifikasi apa saja yang menjadi penyebab dari hasil akurasi dan kinerja.

Output:

Mengambil kesimpulan dan perbaikan kode program dengan tujuan memperbaiki akurasi dan kinerja aplikasi.

3.2 Metode Pengumpulan Data

Metode pengumpulan data digunakan untuk mencari referensi dan sebagai acuan rancangan aplikasi. Metode yang digunakan oleh Penulis dijelaskan sebagai berikut.

3.2.1 Studi Literatur

Tugas akhir yang baik harus disusun secara sistematis untuk memudahkan langkah – langkah yang akan diambil. Data studi literatur akan dijadikan acuan untuk implementasi dan perancangan sistem pelacakan obyek dengan kamera berbasis Raspberry Pi menggunakan metode *Histogram of Oriented Gradients* dan *Support Vector Machine*.

Studi literatur dikerjakan diawal penyusunan dengan cara membaca jurnal, paper, serta *website* untuk mengkaji lebih dalam mengenai *Histogram of Oriented*

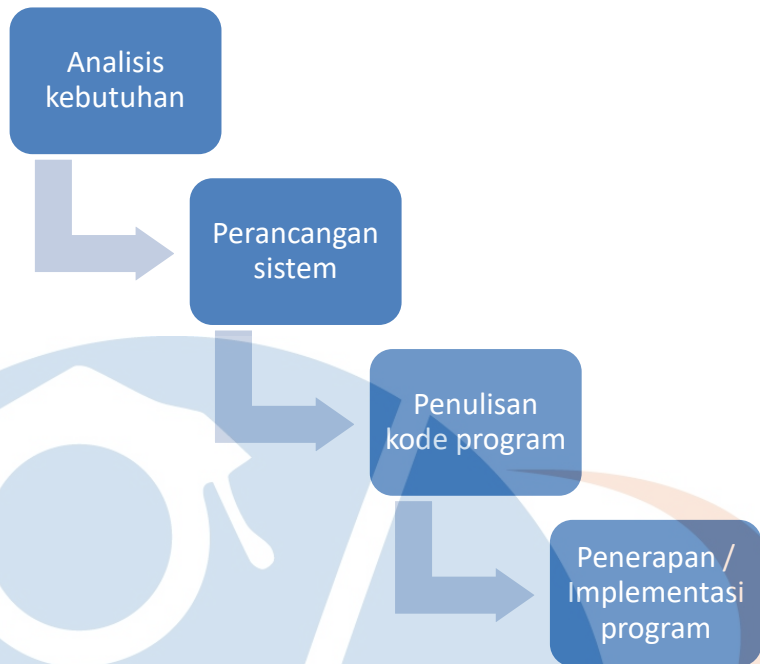
Gradient, Support Vector Machine, OpenCV, Raspberry Pi, Arduino Uno, Servo Motor. Dengan output dari metode *Waterfall* adalah referensi dan *knowledge* mengenai *Histogram of Oriented Gradient, Support Vector Machine, OpenCV, Raspberry Pi, Arduino Uno, Servo Motor.* Adapun daftar referensi dari studi literatur ini dapat dilihat di daftar pustaka tugas akhir ini.

3.2.2 Eksperimen

Eksperimen yang dilaksanakan dalam tugas akhir ini berupa merekam aktifitas di suatu ruangan menggunakan *webcam* dan mencari obyek yang dideteksi sebagai manusia yang kemudian diikuti oleh kamera ketika sudah tidak masuk dalam jarak pandang kamera. Dari pelaksanaan eksperimen yang dilakukan nanti akan dicari tingkat akurasi pendeteksian obyek manusia berdasar dua parameter, yaitu jarak kamera terhadap obyek dan intensitas cahaya pada saat perekaman aktifitas menggunakan *webcam*. Dari pengujian yang dilakukan, maka akan ditarik kesimpulan yang akan menjadi bahan pertimbangan untuk perbaikan tugas akhir ini selanjutnya sehingga dapat menambahkan fitur ataupun perangkat penunjang lainnya.

3.3 Metode Pengembangan Sistem

Dalam pengembangan Aplikasi Pelacakan Obyek, Penulis menggunakan metode *waterfall*. Metode *waterfall* (Gambar 3.2) merupakan pendekatan perangkat lunak yang sistematis dan sekuensial yang dimulai dari tahap analisis, desain, kode, pengujian dan pemeliharaan. Metode *waterfall* juga dikenal sebagai *linier sequential* atau *classic life cycle* (M. Hidayatullah).



Gambar 3.2 Metode *waterfall* (Pengetahuan dan Teknologi, 2018)

Tahapan-tahapan metode *waterfall*:

1. **Analisa kebutuhan (*Requirement Analysis*)**

Langkah ini merupakan menganalisis kebutuhan dari tugas akhir ini dengan cara menyimpulkan kegiatan studi literatur yang relevan dengan kebutuhan *hardware*, *software* dan lokasi dari perangkat pelacakan obyek ini dipasang. Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan ekspektasi dari user dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan pengembangan aplikasi yang selanjutnya ditulis kedalam bahasa pemrograman.

2. **Perancangan Sistem (*System Design*)**

Perancangan sistem adalah proses desain alur cara kerja pelacakan obyek yang Penulis lakukan untuk mengidentifikasi syarat kebutuhan dalam membangun aplikasi pelacakan obyek sebelum proses *coding*. Tahapan ini akan menghasilkan dokumen yang disebut *system requirement*. Dokumen inilah yang akan digunakan *programmer* untuk melakukan aktivitas pembuatan sistemnya.

3. **Penulisan Kode Program (*coding*)**

Coding merupakan proses yang dilakukan peneliti untuk mengimplementasikan desain alur kerja aplikasi yang sudah dirancang sehingga menjadi sebuah sistem aplikasi secara utuh. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat.

4. Penerapan / pengujian program (*Integration & Testing*)

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, desain dan pengkodean maka sistem yang sudah jadi akan di uji coba sebelum benar-benar digunakan diperangkat pemindai seperti *Camera Surveillance*. Tujuan pengujian adalah mengetahui kinerja sistem dan menemukan kesalahan-kesalahan untuk kemudian bisa diperbaiki.

Pada aplikasi pelacakan obyek ini dilakukan pengujian dalam kondisi yaitu:

- i) Pemasangan lokasi perangkat di tengah dinding dengan perangkat menghadap ke depan.
- ii) Pencahayaan di dalam ruangan ketika menggunakan lampu ditambah cahaya matahari.
- iii) Pengujian dengan setting dan jarak pelacakan obyek yang bervariasi untuk mengetahui tingkat akurasi dan kinerja aplikasi yang terpasang ke dalam perangkat.

3.4 Lingkungan Pengembangan

Jenis lingkungan fisik pengembangan sistem untuk tugas akhir yang sedang dilaksanakan dapat dilihat pada Tabel 3.1.

Tabel 3.1 Deskripsi lingkungan fisik pengembangan

Parameter lingkungan fisik	Deskripsi
Lokasi:	<i>Indoor</i>
Luas:	6.5 x 9.5 meter.
Luas yang digunakan untuk pengujian:	6.5 x 4 meter.

Jarak pengujian:	0 meter hingga 8 meter
Jarak kamera <i>webcam</i> dari lantai:	± 124 cm.

Bahan dan alat penelitian yang dilakukan adalah sebagai berikut:

A. Perangkat Keras

Laptop:

HP 240 G1

- Processor Intel(TM) i3-3110M CPU @ 2.40 GHz
- RAM 2.00 GB

Mini PC:

Raspberry Pi 3 Model B

- Processor Broadcom BCM2837 64bit Quad Core CPU at 1.2Ghz
- RAM 1.00 GB
- 4 x USB 2 Ports
- 10/100 LAN Port
- 3.5mm 4-pole Composite Video and Audio Output Jack
- CSI Camera Port
- Full Size HDMI Video Input
- Micro USB Power Input. Upgraded switched power source that can handle up to 2.5 Amps
- DSI Display Port
- Micro SD Card Slot
- 40 Pin Extended GPIO
- Dimension 85.6mm x 56mm x 21mm

Microcontroller:

Arduino Uno Rev 3

- Microcontroller ATmega328P

- Digital I/O 14 Pins (of which 6 provide PWM output)
- PWM Digital I/O 6 Pins
- Analog Input 6 Pins
- Operating Voltage 5V and Input Voltage 7 - 12 V
- Flash Memory 32 KB of which 0.5 KB used by bootloader
- SRAM 2 KB
- EEPROM 1 KB
- Clock Speed 16 MHz
- Dimension 68.6mm x 53.4mm
- Weight 25 gram

Webcam:

Logitech

- Resolution VGA

Servo Motor:

Tower Pro MG995

- Digital Modulation
- Gear Type Metal
- Pulse Cycle 1ms
- Dimensions 40.7mm x 19.7mm x 42.9mm

B. Perangkat Lunak

- OpenCV 3.0
- Arduino IDE 1.6.12
- Python 2.7.14
- Python IDLE 2.4.6

STT INF