



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EE 184801

**SISTEM PEMBUATAN DATASET DENGAN METODA
GAUSSIAN MIXTURE MODEL (GMM) UNTUK
PEMBELAJARAN OBYEK**

Kelvin Liusiani
NRP 0711164000080

Dosen Pembimbing
Astria Nur Irfansyah, ST., M.Eng., Ph.D.
Muhammad Attamimi, B.Eng., M.Eng., Ph.D.

DEPATERMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - EE 184801

***DATASET MAKER SYSTEM USING GAUSSIAN MIXTURE
MODEL (GMM) METHOD FOR OBJECT LEARNING***

Kelvin Liusiani
NRP 0711164000080

Supervisor

Astria Nur Irfansyah, ST., M.Eng., Ph.D.
Muhammad Attamimi, B.Eng., M.Eng., Ph.D.

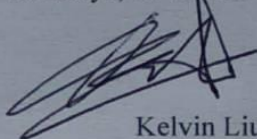
***ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020***

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “Sistem Pembuatan Dataset dengan Metoda *Gaussian Mixture Model* (GMM) untuk Pembelajaran Obyek” adalah benar – benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan – bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 15 Januari 2020



Kelvin Liusiani
NRP. 0711164000080

-----*Halaman ini sengaja dikosongkan*-----

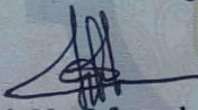
**SISTEM PEMBUATAN DATASET DENGAN
METODA *GAUSSIAN MIXTURE MODEL* (GMM)
UNTUK PEMBELAJARAN OBYEK**

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Menyetujui:

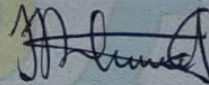
Dosen Pembimbing I



Astria Nur Irfansyah, ST.,
M.Eng., Ph.D

NIP. 198103252010121002

Dosen Pembimbing II



Muhammad Attamimi, B.Eng.,
M.Eng., Ph.D

NIP. 198503272019031006



-----Halaman ini sengaja dikosongkan-----

SISTEM PEMBUATAN DATASET DENGAN METODA *GAUSSIAN MIXTURE MODEL* (GMM) UNTUK PEMBELAJARAN OBYEK

Nama : Kelvin Liusiani
Pembimbing : 1. Astria Nur Irfansyah, ST., M.Eng., Ph.D.
2. Muhammad Attamimi, B.Eng., M.Eng., Ph.D.

ABSTRAK

Informasi visual sangat diperlukan dalam membangun sistem cerdas termasuk di dalamnya robot cerdas. Khususnya *Domestic Service Robot* (DSR) dalam menjalankan tugasnya di lingkungan rumah tangga yang kompleks dan dinamis. Pengenalan tersebut bermacam-macam jenisnya, salah satunya adalah pengenalan terhadap obyek (*object recognition*). Langkah pertama dalam pengenalan obyek adalah pembelajaran terhadap obyek yang belum dikenali, yang selanjutnya akan disebut dengan *object learning*. Ekstraksi obyek dan pemberian label merupakan bagian dari pembelajaran obyek. Dalam penelitian ini, informasi visual yang dapat diambil dari Kinect V2 akan dimanfaatkan untuk pembelajaran obyek dengan menggunakan metode *Unsupervised Learning*, yaitu *Gaussian Mixture Model* (GMM) dimanfaatkan untuk melakukan ekstraksi obyek, aplikasi *smartphone* dibuat untuk membuat pemberian label nama obyek lebih interaktif dan dikenalkan istilah meja belajar yang digunakan untuk memutar obyek sehingga pembelajaran dapat dilakukan secara 360° dengan ketelitian 9° setiap informasi visualnya. Berdasarkan pengujian yang telah dilakukan, sistem ini dapat melakukan ekstraksi obyek dengan nilai kepresisian 83,1%, dapat bekerja di lingkungan yang dinamis dan dapat mempelajari obyek dengan spesifikasi ukuran lebih besar dari 3 x 3 x 2cm, lebih kecil dari 28 x 28 x 30cm, serta dengan berat tidak melebihi 800 gr. Pada penelitian ini obyek yang telah dipelajari adalah sejumlah 72 obyek. Dengan spesifikasi tersebut, sebagian besar obyek – obyek di rumah tangga dapat dipelajari. Hasil ekstraksi dari obyek akan disimpan bersama label dan pose dari obyek tersebut sehingga menjadi dataset untuk dimanfaatkan pada sistem cerdas, salah satunya adalah DSR.

Kata kunci: *Robotics, Machine Vision, Machine Learning*

-----*Halaman ini sengaja dikosongkan*-----

DATASET MAKER SYSTEM USING GAUSSIAN MIXTURE MODEL (GMM) METHOD FOR OBJECT LEARNING

Name : Kelvin Liusiani
Supervisor : 1. Astria Nur Irfansyah, ST., M.Eng., Ph.D.
2. Muhammad Attamimi, B.Eng., M.Eng., Ph.D.

ABSTRACT

Visual information is needed in building intelligent systems including intelligent robots. Especially Domestic Service Robot (DSR) in carrying out their duties in a complex and dynamic household environment. The introduction of various types, one of which is the recognition of objects. The first step in object recognition is the learning of objects that are not yet recognized, which will then be referred to as object learning. Object extraction and labeling are part of object learning. In this research, visual information that can be taken from Kinect V2 will be used for object learning by using the Unsupervised Learning method, namely the Gaussian Mixture Model (GMM) used to extract objects, smartphone applications are made to make labeling objects names more interactive and introduce study table used to rotate objects so that learning can be done 360° with accuracy of 9° for each visual information. Based on testing that has been done, this system can extract objects with a precision value of 83.1%, can work in a dynamic environment and can study objects with size specifications greater than 3 x 3 x 2cm, smaller than 28 x 28 x 30cm, and with a weight not exceeding 800 gr. In this study, 72 objects were studied. With these specifications, most objects in the household can be studied. The results of the extraction of objects will be stored with the labels and poses of the object so that it becomes a dataset to be used in intelligent systems, one of which is DSR.

Keywords: Robotics, Machine Vision, Machine Learning

-----*Halaman ini sengaja dikosongkan*-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan tugas akhir yang berjudul “**Sistem Pembuatan Dataset Dengan Metoda *Gaussian Mixture Model* (GMM) untuk pembelajaran Obyek**”. Pada kesempatan ini penulis menyampaikan terima kasih atas segala bantuan dan dukungannya yang telah diberikan selama proses pembuatan tugas akhir ini kepada:

1. Keluarga tercinta yang selalu memberikan do’a dan dukungan yang sangat berarti dalam keadaan apapun.
2. Bapak Astria Nur Irfansyah, ST., M.Eng., Ph.D. dan Bapak Muhammad Attamimi, B.Eng., M.Eng., Ph.D. selaku Dosen Pembimbing atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya tugas akhir,
3. Seluruh Staff/Karyawan/Dosen Departemen Teknik Elektro yang telah memberikan banyak ilmu dan menciptakan suasana belajar yang mendukung.
4. Teman-teman e56 yang telah berjuang bersama semenjak awal masuk perkuliahan.
5. Semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir ini yang tidak dapat disebutkan satu per satu.

Penulis berharap semoga tugas akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 15 Januari 2020

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR	i
TUGAS AKHIR.....	iii
ABSTRAK.....	v
<i>ABSTRACT</i>	7
KATA PENGANTAR	9
DAFTAR ISI.....	11
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
BAB I Pendahuluan	1
1.1 Latar Belakang.....	1
1.2 Permasalahan.....	2
1.3 Tujuan.....	2
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Penulisan.....	4
1.7 Relevansi	5
Bab II Tinjauan Pustaka dan Teori Penunjang.....	7
2.1 Dataset Obyek.....	7
2.2 Kinect V2.....	8
2.3 Motor Stepper Nema 17	9
2.4 Modul Bluetooth	10
2.5 Arduino Nano.....	11
2.6 Pengolahan Citra Digital	12
2.6.1 RGB	12
2.6.2 HSV.....	12
2.6.3 Depth.....	13
2.6.4 NIR.....	15
2.7 Point Cloud	16
2.8 OpenCV	17
2.9 GMM.....	17
2.10 Metrics	19
2.11 Ionic Framework	20
2.12 Firebase.....	22
BAB III Perancangan Sistem	25
3.1 Diagram Blok Sistem.....	25

3.2 Perancangan Perangkat Keras.....	26
3.2.1 Perancangan Elektronik	27
3.2.2 Perancangan Mekanik.....	29
3.3 Pengambilan Data Berbasis Informasi Visual	31
3.3 Proses Pembelajaran Permukaan Meja	34
3.4 Proses Pengambilan Data Obyek.....	36
3.4.1 Ekstraksi Obyek.....	38
3.4.2 Pemutar Obyek	42
3.5 Mencari SOI.....	43
3.5.1 Pencarian Titik Tengah Permukaan Meja.....	44
3.5.1 Inisiasi SOI	46
3.6 Firebase.....	47
3.7 Aplikasi <i>Smartphone</i> Antarmuka Pengguna	48
Bab IV Pengujian dan Analisis	53
4.1 Pengujian Perangkat Keras	54
4.2 Pengujian Pengambilan Data	56
4.2.1 Pengujian Pencarian Titik Tengah Permukaan Meja	56
4.2.2 Pengujian Pengambilan Data pada Berbagai Jenis Obyek....	59
4.2.3 Pengujian Pengambilan Data pada Lingkungan yang Dinamis	61
4.3 Evaluasi Hasil Pembelajaran Obyek	65
4.4 Hasil Akhir Dataset.....	70
Bab V Penutup.....	75
5.1 Kesimpulan	75
5.2 Saran.....	75
Daftar Pustaka.....	77
Lampiran.....	79
A. Evaluasi Pembelajaran Obyek	79
B. Dataset.....	83
C. Tampilan Antarmuka Pengguna.....	89
D. Dokumentasi Eksperimen	91
E. Program	97
E.1 Program Utama (PC).....	97
E.2 Program Arduino	131
F. Datasheet Stepper Motor	135
BIODATA PENULIS	137

DAFTAR GAMBAR

Gambar 2.1 Obyek-Obyek pada <i>RGB-D Object Dataset</i> [7]	7
Gambar 2.2 Hasil deteksi obyek pada beberapa kategori dan <i>instance</i> ... 8	
Gambar 2.3 Kinect V2	8
Gambar 2.4 Motor Stepper.....	10
Gambar 2.5 Bluetooth HC-05	10
Gambar 2.6 Arduino Nano.....	11
Gambar 2.7 Visualisasi HSV [9].....	13
Gambar 2.8 Visualisasi metode SR [10]	14
Gambar 2.9 Visualisasi metode stereo depth [10].....	14
Gambar 2.10 Visualisasi ToF [10].....	15
Gambar 2.11 Panjang gelombang Near-Infrared [11].....	15
Gambar 2.12 Point Cloud hasil dari Kinect V2 [13].....	16
Gambar 2.13 Logo OpenCV	17
Gambar 2.14 <i>Mixed Distribution</i>	18
Gambar 2.15 Distribusi gaussian (normal)	19
Gambar 2.16 Logo Ionic Framework [15]	21
Gambar 2.17 Penampilan Awal Ionic Framework.....	22
Gambar 2.18 Logo Firebase [17]	22
Gambar 2.19 Layanan Firebase.....	23
Gambar 3.1 Skema ilustrasi keseluruhan sistem.....	25
Gambar 3.2 Dataset keluaran sistem	26
Gambar 3.3 Perancangan rangkaian elektronik.....	28
Gambar 3.4 Komunikasi Sistem	28
Gambar 3.5 Bentuk dan dimensi motor stepper	29
Gambar 3.6 Kerangka badan meja belajar	29
Gambar 3.7 Desain permukaan meja	30
Gambar 3.8 Komponen perangkat keras meja belajar.....	31
Gambar 3.9 Perangkat keras meja belajar	31
Gambar 3.10 Data berbasis informasi visual (RGB-D NIR).....	32
Gambar 3.11 Informasi visual warna dengan resolusi (a) 1920 x 1080 dan (b) 512 x 424 yang telah dipetakan	33
Gambar 3.12 Blok diagram proses pembelajaran permukaan meja	34
Gambar 3.13 Penggalan program GMM dan keterangannya.....	35
Gambar 3.14 Blok diagram proses pengambilan data obyek.....	37

Gambar 3.15 Visualisasi matriks probabilitas permukaan meja	38
Gambar 3.16 Visualisasi matriks probabilitas bukan permukaan meja	39
Gambar 3.17 Informasi visual yang berada dalam SOI	40
Gambar 3.18 Visualisasi matriks probabilitas obyek (a) dan matriks hasil thresholding (b).....	40
Gambar 3.19 Hasil ekstraksi obyek pada gambar warna terkalibrasi (a) dan pada gambar warna HD (b).....	41
Gambar 3.20 Keseluruhan proses ekstraksi obyek	41
Gambar 3.21 Diagram alir pemutaran meja belajar	42
Gambar 3.22 Visualisasi ROI (a) dan SOI (b).....	43
Gambar 3.23 Diagram alur pencarian titik tengah permukaan meja.....	45
Gambar 3.24 Visualisasi hasil pencarian titik tengah meja	46
Gambar 3.25 Diagram alur penginisiasian SOI	47
Gambar 3.26 Tampilan <i>realtime database service</i> Firebase.....	48
Gambar 3.27 Tampilan antarmuka pengguna halaman percakapan	49
Gambar 3.28 Diagram alir sistem dengan masukan perintah.....	50
Gambar 3.29 Tampilan antarmuka pengguna halaman <i>storage</i>	50
Gambar 4.1 Situasi eksperimen	53
Gambar 4.2 Situasi eksperimen	53
Gambar 4.3 Pengujian 1 pencarian titik tengah meja	57
Gambar 4.4 Pengujian 2 pencarian titik tengah meja	57
Gambar 4.5 Pengujian 3 pencarian titik tengah meja	57
Gambar 4.6 Pengujian 4 pencarian titik tengah meja	58
Gambar 4.7 Pengujian 5 pencarian titik tengah meja	58
Gambar 4.8 Pengujian SOI.....	62
Gambar 4.9 Pengujian 1 Ekstraksi Obyek	63
Gambar 4.10 Pengujian 2 ekstraksi obyek.....	64
Gambar 4.11 Pengujian 3 ekstraksi obyek.....	65
Gambar 4.12 Sampel proses ekstraksi obyek	66
Gambar 4.13 Contoh <i>bounding box</i> ekstraksi obyek	68
Gambar 4.14 Evaluasi ekstraksi obyek	69
Gambar 4.15 Dataset informasi visual warna obyek lifebuoy	73
Gambar 4.16 Dataset informasi visual warna obyek sendok	73
Gambar 4.17 Visualisasi matriks probabilitas obyek.....	74

DAFTAR TABEL

Tabel 2.1 Spesifikasi Kinect V2.....	9
Tabel 2.2 Spesifikasi Motor Stepper 17HS8401	10
Tabel 2.3 Pinout HC-05	11
Tabel 2.4 Spesifikasi Arduino Nano	11
Tabel 2.5 Hubungan hasil prediksi dan sesungguhnya	19
Tabel 4.1 Hasil pengujian perangkat keras sebelum dan setelah diputar	54
Tabel 4.2 Pengujian ekstraksi obyek pada berbagai bahan	59
Tabel 4.3 Tabel volume obyek yang terukur dengan beberapa variabel jarak	67
Tabel 4.4 Tabel volume obyek sebenarnya	67
Tabel 4.5 Perhitungan kesalahan pengukuran	68
Tabel 4.6 Perbandingan hasil evaluasi ekstraksi obyek	69
Tabel 4.7 Daftar label dan kategori dataset obyek	70
Tabel 4.8 Tabel spesifikasi alat pada sistem ini	74

-----*Halaman ini sengaja dikosongkan*-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan di bidang teknologi, komunikasi, dan informasi telah membawa Indonesia memasuki era yang baru dimana kebutuhan akan industri robot cerdas menjadi tak terhindari. Salah satunya adalah *Domestic Service Robot* (DSR) yang berada di lingkungan rumah tangga, dimana manusia sebagai target yang ingin dilayani. Skenario semacam ini menuntut sistem cerdas termasuk DSR agar dapat berinteraksi sebaik mungkin karena lingkungan rumah tangga cenderung kompleks dan dinamis [1]. Untuk melayani manusia, robot ini dituntut dapat melakukan manipulasi terhadap obyek – obyek yang ada di lingkungan rumah tangga. Namun, untuk melakukan manipulasi, robot harus mengenali obyek tersebut terlebih dahulu [2].

Metode pengenalan obyek atau *object recognition* yang paling lazim adalah menggunakan pendekatan *supervised – learning*. Dalam pendekatan ini, dibutuhkan masukan gambar dan anotasi atau label [3]. Agar dapat mengenali obyek dengan baik serta beraneka ragam macamnya, dibutuhkan banyak data yang terlabel. Label ini umumnya didapatkan dengan cara mengotaki obyek disetiap fotonya untuk membedakan obyek yang ingin dikenali dengan yang bukan, seperti obyek lain ataupun background. Lalu, dilanjutkan dengan pemberian label sebagai nama dari obyek tersebut. Penglabelan ini tidak efektif karena dengan data yang sangat banyak, dilakukan dengan satu – persatu oleh manusia dan memakan waktu yang lama. Bahkan kegiatan pemberian label ini dijadikan sebagai pekerjaan, dan terdapat beberapa penyedia jasa pemberian label seperti sklye.ai [4], WorkAround [5], precise BPO solution [6].

Lalu dalam pengaplikasiannya, pengenalan obyek yang baik tidak hanya dapat dilakukan dari sisi depan saja, tetapi bisa dilakukan dari berbagai sisi, layaknya manusia lakukan. Karena obyek di lingkungan rumah tangga tidak selalu sama dari berbagai sisi, baik dalam konteks bentuk dan persebaran warnanya. Oleh karena itu, dibutuhkan juga informasi visual obyek dari berbagai sisi untuk meningkatkan keyakinan dari robot dalam pengenalan [6].

Pada penelitian ini, diajukan sistem pengambilan informasi visual dari suatu obyek, dikenalkan perangkat keras meja belajar yang akan

memudahkan dalam pengambilan data dari berbagai sisi dan juga diajukan metode dalam pengestrakan obyek yang memudahkan dalam pemberian label. Informasi visual yang diambil adalah berupa warna RGB, jarak (*Depth*), dan NIR (*Near-Infrared*). Ketiga data tersebut akan dimanfaatkan pada proses ekstraksi obyek. Keseluruhan sistem ini akan disebut sebagai proses pembelajaran obyek (*object learning*) sehingga DSR dapat menjalankan tugasnya dengan baik. Dengan adanya sistem ini, diharapkan DSR dapat mempelajari obyek yang belum ia ketahui sebelumnya.

Seperti contoh kasusnya adalah dalam pengambilan barang yang diperintahkan oleh manusia, yaitu mengambilkan mug. Dalam kasus ini, DSR harus dapat melakukan pencarian obyek. Setelah itu dilakukanlah pengenalan obyek. Jika telah ditemukan mug oleh DSR, DSR harus mengetahui pose mug tersebut. Misalkan saja, pose mug tersebut adalah 90 derajat, dimana pegangan mug merupakan acuannya. Pada kasus ini, dengan DSR mengetahui posenya, DSR dapat mengambil mug selayaknya manusia, dengan memasukan jarinya ke pegangan mug tersebut sehingga DSR dapat menyajikannya dengan baik. Begitu juga dengan obyek – obyek lainnya, sehingga DSR dapat memanipulasi obyek selayaknya manusia.

1.2 Permasalahan

Perumusan masalah pada Penelitian ini adalah:

1. Bagaimana pengambilan informasi visual suatu obyek dari berbagai sisi dapat dilakukan
2. Bagaimana memanfaatkan hasil pengambilan data untuk melakukan ekstraksi obyek
3. Bagaimana pemberian label dari suatu obyek dapat dilakukan

1.3 Tujuan

Tujuan usulan penelitian ini dapat dijabarkan sebagai berikut:

1. Pembuatan meja belajar dalam membantu pengambilan data dari berbagai sisi
2. Pemanfaatan informasi visual RGBD dan NIR untuk pembelajaran obyek (*object learning*) pada sistem cerdas
3. Pembuatan aplikasi antarmuka pengguna untuk membantu pemberian label dari suatu obyek

1.4 Batasan Masalah

Adapula beberapa batasan permasalahan dalam penelitian ini:

1. Sensor yang digunakan adalah Kinect V2
2. Metode ekstraksi obyek yang digunakan adalah *Gaussian Mixture Model* (GMM)
3. Obyek yang dipelajari adalah obyek di sekitar lingkungan rumah tangga dan berukuran tidak lebih dari 28x28x30cm serta memiliki berat <800 gram
4. Penelitian ini hanya sebatas dalam pembuatan dataset obyek

1.5 Metodologi

Langkah – langkah yang dikerjakan pada penelitian ini adalah sebagai berikut:

1. Studi Literatur

Studi literatur berisikan kegiatan pengumpulan dan pengkajian dasar teori yang terpercaya untuk menunjang baik pengerjaan maupun penulisan penelitian ini.

2. Pembuatan Perangkat Lunak Pengambilan Data

Pada tahap ini, dilakukan pembuatan perangkat lunak untuk pengambilan informasi visual RGBD dan NIR secara bersamaan. Pemetaan informasi visual RGB pada informasi visual depth juga dilakukan untuk memudahkan pengolahan ketiga informasi visual tersebut.

3. Perancangan Perangkat Keras

Tahap perancangan perangkat keras dilakukan untuk membuat alat meja belajar yang dapat berputar dan pengadaan sensor pengambil data, meliputi kamera Kinect V2 dan tripod kamera. Perancangan perangkat keras untuk membuat meja belajar meliputi perancangan desain 3D dan pencetakan kerangka meja, perancangan dan pembuatan permukaan meja, perancangan elektronik rangkaian pemutar meja, pembuatan penghubung motor dan permukaan meja, dan perangkaian meja belajar.

4. Perancangan Perangkat Lunak Utama

Pada tahap perancangan perangkat lunak utama meliputi, pembuatan program pemutar meja, program ekstraksi obyek, pembuatan program interaksi antarmuka pada *smartphone*.

5. Pengintegrasian Sistem

Pada tahap ini, perangkat keras meja belajar akan diintegrasikan dengan program pengambilan data, dan program interaksi antarmuka melalui smartphone.

6. Pengujian Sistem

Setelah pengintegrasian sistem dilakukan, pengujian dari sistem akan dilakukan untuk mengetahui kehandalan dari sistem. Pengujian perangkat keras dan perangkat lunak akan dilakukan secara terpisah. Pengujian perangkat keras meja belajar dilakukan untuk mengetahui spesifikasi obyek yang dapat diputar oleh meja belajar. Pengujian perangkat lunak dilakukan untuk mengetahui kehandalan sistem dalam lingkungan yang kompleks dan dinamis.

7. Pengambilan Data

Pada tahap ini, sistem akan digunakan dalam pengambilan informasi visual yang akan menghasilkan RGBD dan NIR dari obyek yang belum diekstraksi dan informasi visual yang sudah diekstraksi.

8. Analisa dan Evaluasi

Pada tahap ini, akan dilakukan analisa terhadap data yang didapatkan sehingga akan diketahui seberapa baik sistem yang telah dibuat. Analisa dilakukan pada informasi visual hasil ekstraksi dengan melihat *error* dan *losses* yang didapat dari membandingkan hasil pemberian label manual yang dilakukan oleh manusia. Lalu dilakukan evaluasi serta perbaikan apabila hasil sistem kurang memuaskan.

9. Penyusunan Laporan

Penyusunan laporan akan dilakukan seiring dengan tahap-tahap lainnya. Isinya berkaitan dengan penelitian yang dikerjakan, meliputi pendahuluan, studi literatur, perancangan dan pembuatan sistem, pengujian dan analisa serta penutup.

1.6 Sistematika Penulisan

Penulis membagi laporan penelitian ini menjadi lima bab yang terhubung satu sama lain. Hal ini untuk menghindari kesalahan interpretasi terhadap isi di dalam laporan. Penjelasan tentang masing-masing bab dibuat dengan sistematika penulisan sebagai berikut:

BAB 1 PENDAHULUAN

Pada bab ini dijelaskan tentang latar belakang, permasalahan, tujuan, batasan masalah, metodologi, sistematika penulisan, serta relevansi dari penelitian yang dilakukan.

BAB 2 TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai tinjauan pustaka yang membantu penelitian. Beberapa teori yang dapat membantu dalam penelitian ini berlangsung adalah motor stepper, komunikasi melalui Bluetooth, pengolahan citra, dan Gaussian Mixture Model (GMM)

BAB 3 PERANCANGAN SISTEM

Pada bab ini membahas tentang perancangan sistem yang akan digunakan pada penelitian ini. Perancangan sistem ini meliputi perancangan perangkat keras dan perancangan perangkat lunak.

BAB 4 PENGUJIAN DAN ANALISIS

Pada bab ini dibahas mengenai pengujian dari perangkat keras dan perangkat lunak. Sistem pengambilan data juga diuji dengan melakukan evaluasi pada data hasil dari pembelajaran obyek. Selanjutnya dilakukan analisa berdasarkan data hasil pengujian.

BAB 5 PENUTUP

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Hasil dari penelitian ini diharapkan alat dan sistem ini dapat memudahkan dalam pengambilan data dan proses pemberian label dari suatu obyek. Sistem ini juga diharapkan dapat diterapkan pada robot DSR, sehingga robot dapat mengenali obyek dan belajar mengenai obyek yang ia tidak kenali sebelumnya.

-----*Halaman ini sengaja dikosongkan*-----

BAB II

TINJAUAN PUSTAKA DAN TEORI PENUNJANG

Bab ini membahas tentang tinjauan pustaka dan teori penunjang yang dapat membantu penelitian. Tinjauan pustaka akan berisi mengenai penelitian – penelitian sebelumnya yang telah dilakukan. Teori ini nantinya akan digunakan sebagai dasar untuk perancangan sistem pada bab III.

2.1 Dataset Obyek

Pada umumnya, untuk membuat model yang dapat mengenali obyek atau objek dibutuhkan dataset informasi visual dari obyek - obyek atau objek itu sendiri. Umumnya setiap obyek memiliki beberapa informasi visual dan label dari obyek tersebut. Salah satu cara mudah dalam pengumpulan dataset adalah dengan mengambil gambar yang terdapat pada internet. Namun, dataset tersebut hanya mengandung informasi visual warna saja dan bahkan cenderung sulit didapatkan untuk obyek – obyek produk lokal tetapi banyak ditemui di rumah kita.

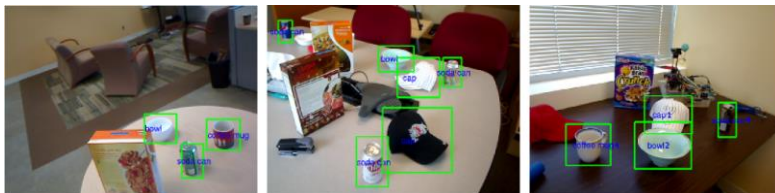
Pada penelitian yang dilakukan oleh K. Lai [7], peneliti membuat dataset yang mengandung informasi visual RGB ditambah dengan *Depth* dari berbagai obyek yang diambil dari berbagai sisi dengan menggunakan perangkat keras pemutar. Peneliti juga menggunakan informasi visual berupa warna dan *depth* untuk melakukan ekstraksi obyek pada penelitian ini. Pada gambar 2.1 atas merupakan contoh – contoh obyek yang terdapat dalam dataset yang diletakan di atas perangkat keras pemutar, sedangkan gambar 2.1 bawah merupakan hasil ekstraksi obyek pada penelitian ini.



Gambar 2.1 Obyek-Obyek pada *RGB-D Object Dataset* [7]

Dengan adanya data RGB-D tersebut peneliti dapat melakukan pengenalan dan pencarian obyek yang terlihat pada gambar 2.2. Pada pengenalan obyek yang peneliti lakukan, peneliti tidak hanya dapat

melakukan pengenalan kategori, maka dapat melakukan pengenalan *instance* yang terlihat pada gambar 2.2 kanan dengan memanfaatkan fitur yang diolah dan menggunakan metode SVM. Terlihat bahwa peneliti dapat mengenali bahwa *cap* tersebut merupakan *cap2*, *coffee mug* tersebut merupakan *coffee mug4*, *bowl* tersebut merupakan *bowl2*, dan *soda can* tersebut merupakan *soda can4*.



Gambar 2.2 Hasil deteksi obyek pada beberapa kategori dan *instance*

2.2 Kinect V2

Kinect merupakan sensor penangkap gambar seperti kamera pada umumnya namun menyediakan informasi visual yang lebih lengkap, yaitu adanya informasi visual *depth* dan *Near-Infrared* sebagai tambahannya. Sensor ini sering digunakan pada konsol game Xbox-360. Gambar 2.3 menunjukkan bentuk fisik dari Kinect V2.



Gambar 2.3 Kinect V2

Dalam pengambilan informasi visual, Kinect dapat menangkap informasi visual warna RGB, informasi visual *depth* (*Depth*), dan informasi visual NIR (*Near – Infrared*). Pada tabel 2.1 merupakan spesifikasi mengenai resolusi dari tiap informasi visual yang diambil, batas pandang dan spesifikasi lainnya. Pada Kinect V2, dalam

pengambilan informasi visual *depthnya* menggunakan metode *Time of Flight* yang akan dijelaskan pada subbab 2.6.3. Dalam pemanfaatannya, Kinect biasanya digunakan untuk mengetahui pose dan mengikut pergerakan dari tubuh manusia. Kemampuan ini dimanfaatkan dalam konsol game Xbox – 360 untuk menghadirkan permainan yang dikendalikan oleh pergerakan tubuh dari pemainnya. Meski Kinect V2 sudah tidak diproduksi lagi, perangkat lunak penunjang telah disediakan oleh Windows dan masih dapat digunakan. Tabel 2.1 merupakan tabel spesifikasi dari Kinect V2.

Tabel 2.1 Spesifikasi Kinect V2

<i>Feature</i>	<i>Kinect V2</i>
<i>Color Camera</i>	<i>1920 X 1080 @30 fps</i>
<i>Depth Camera</i>	<i>512 X 424</i>
<i>Max Depth Distance</i>	<i>4.5 M</i>
<i>Min Depth Distance</i>	<i>50 cm</i>
<i>Horizontal Field of View</i>	<i>70 degrees</i>
<i>Vertical Field of View</i>	<i>60 degrees</i>
<i>USB Standard</i>	<i>3.0</i>
<i>Supported OS</i>	<i>Win 8</i>

2.3 Motor Stepper Nema 17

Motor stepper merupakan motor listrik yang berputar dengan cara melangkah atau stepping. Yang dimaksud melangkah adalah dalam mencapai satu putaran penuh, motor ini memerlukan beberapa langkah untuk mencapainya. Salah satu contohnya adalah motor stepper nema 17 yang berputar sebesar 1.8 derajat untuk setiap satu langkahnya, sehingga dibutuhkan 200 langkah untuk melakukan satu putaran penuh. Untuk menggerakkan motor stepper ini, dapat dilakukan dengan mengirimkan pulsa yang bergantian pada tiap saluran coilnya. Pada gambar 2.4 menunjukkan bentuk fisik stepper motor. Kelebihan motor stepper adalah memiliki presisi pergerakan sudut yang lebih baik dibandingkan motor DC, dan juga dapat bergerak penuh dan berputar layaknya motor DC, tidak seperti servo motor.



Gambar 2.4 Motor Stepper

Pada nema 17HS8401 merupakan dua fase stepper motor yang akan menghasilkan tenaga hold torque sebesar 3.2N.cm jika disalurkan arus 1.7 ampere pada tegangan 2.8 Volt pada setiap phasanya, sedangkan tabel 2.2 merupakan tabel spesifikasi dari nema 17HS8401.

Tabel 2.2 Spesifikasi Motor Stepper 17HS8401

Series Model	Step Angle (deg)	Motor Length (mm)	Phase Resistance (ohm)	Phase Inductance (mH)	Holding Torque (N.cm Min)	Detent Torque (N.cm Max)	Rotor Inertia (g.cm ²)	Lead Wire (No.)	Motor Weight (g)
17HS8401	1.8	48	1.7	1.8	3.2	2.6	68	4	350

2.4 Modul Bluetooth

Bluetooth adalah protokol komunikasi wireless yang bekerja pada frekuensi radio 2.4 GHz yang dimanfaatkan untuk mengirim data dari satu divais ke divais lainnya seperti pada divais mikrokontroler Arduino yang berkomunikasi dengan PC, telepon genggam ataupun divais lainnya yang memiliki fasilitas Bluetooth ataupun dapat menggunakan modul Bluetooth tambahan.



Gambar 2.5 Bluetooth HC-05

Salah satu modul Bluetooth yang sering dipakai adalah modul HC-05. Modul ini memiliki 6 pin yang dapat dilihat pada gambar 2.5 dengan beserta keterangannya pada tabel 2.3.

Tabel 2.3 Pinout HC-05

No	Nama Pin	Fungsi
1	EN	Enable
2	VCC	Sumber positif (5 Volt)
3	GND	Gnd tegangan
4	TX	Mengirim data
5	RX	Menerima data
6	STATE	Status bluetooth

2.5 Arduino Nano



Gambar 2.6 Arduino Nano

Arduino nano adalah sebuah mikrokontroler yang menggunakan *chip* Atmega328 yang diproduksi oleh perusahaan Arduino. Dapat dilihat pada gambar 2.6, Arduino Nano memiliki kelebihan yaitu ukurannya yang kecil, sesuai dengan namanya. Mikrokontroler ini memiliki 14 pin digital input/output (pin D0-D13) yang berkerja pada tegangan 5 volt. Terdapat juga 8 pin input analog (pin A0-A7) yang biasa digunakan untuk membaca tegangan dari sensor dengan resolusi 10 bit (0 – 1023), 6 pin output analog (pin D3, D5, D6, D9, D10, D11) yang dapat dimanfaatkan untuk mengeluarkan sinyal PWM (*Pulse Width Modulation*), pin 0 (Rx) dan 1 (Tx) yang dimanfaatkan untuk komunikasi serial serta pin SDA(A4), SCL (A5) yang digunakan untuk komunikasi I2C [8]. Spesifikasi Arduino Nano dijelaskan pada tabel 2.4.

Tabel 2.4 Spesifikasi Arduino Nano

<i>Microcontroller</i>	<i>Atmega328</i>
<i>Architecture</i>	<i>AVR</i>

<i>Operating Voltage</i>	5 V
<i>Input Voltage</i>	7-12 V
<i>DC Current per I/O Pin</i>	40 mA
<i>DC Current for 3.3 V pin</i>	50 mA
<i>Flash Memory</i>	32 Mbyte (Atmega328)
<i>SRAM</i>	2 KB (Atmega328)
<i>EEPROM</i>	512 byte (Atmega328)
<i>Clock Speed</i>	16 MHz
<i>Analog Input Pins</i>	8
<i>Digital I/O Pins</i>	14 (6 PWM Outputs)
<i>PCB Size</i>	18 mm x 45 mm
<i>Weigth</i>	7 Gram

2.6 Pengolahan Citra Digital

Pengolahan citra digital banyak dimanfaatkan baik dalam lingkup masyarakat, perusahaan maupun peneliti. Hal ini dilakukan untuk menyajikan citra digital menjadi lebih informatif. Sehingga data tersebut dapat dimanfaatkan untuk keperluannya masing – masing, seperti menaikkan kualitas foto, melakukan editing, filtering dan lainnya. Berikut beberapa hal penting yang perlu diketahui dalam pengolahan citra digital:

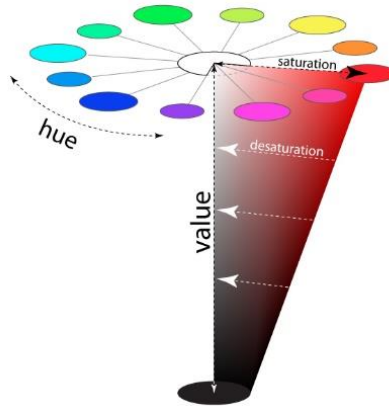
2.6.1 RGB

Ruang warna RGB merupakan ruang warna yang umum digunakan dalam ruang dimensi tiap pixel dalam suatu gambar ataupun video. RGB itu sendiri merupakan kombinasi warna primer dari merah (*Red*), hijau (*Green*), dan biru (*Blue*). Masing – masing komponen memiliki nilai jangkau dari 0-255. Dengan kombinasi dari ketiga nilai tersebut, seluruh warna dapat dihasilkan.

2.6.2 HSV

Ruang warna HSV biasanya digunakan pada pengolahan citra agar mendapatkan hasil yang lebih baik. HSV sendiri terdiri dari *Hue*, *Saturation* dan *Value*. Pada gambar 2.7 merupakan visualisasi dari HSV

yang dapat membantu dalam mengerti pengaruh dari masing – masing perubahan nilai HSV.



Gambar 2.7 Visualisasi HSV [9]

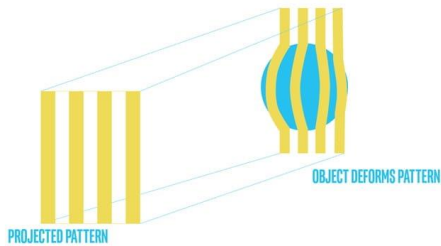
Hue dapat diartikan sebagai nilai warna dari suatu pixel yang memiliki nilai jangkauan 0-360. Untuk *Saturation* sendiri diartikan sebagai nilai dari pencampuran color dengan warna putih yang memiliki nilai jangkauan 0-100. Sedangkan pada *Value* diartikan sebagai nilai dari pencampuran color dengan warna hitam (*gray level*). Perbedaan yang paling besar dari ruang warna RGB dan HSV adalah pada RGB, nilai dari warna sangat bergantung dari ketiga nilai tersebut (RGB), sedangkan pada HSV faktor warna berdiri sendiri pada *Hue* dan dibantu dengan nilai S dan V.

2.6.3 Depth

Depth merupakan data informasi visual yang berisikan jarak dari suatu obyek atau lainnya yang dimasukkan pada Pixel. Ada beberapa cara utk mendapatkan *depth*:

1. Structured Light and Coded Light

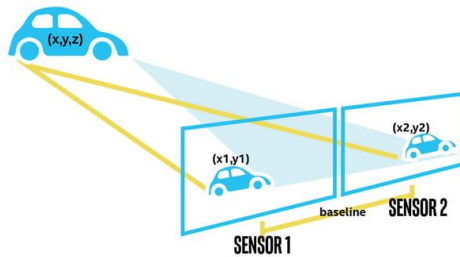
Atau disingkat menjadi SL, merupakan metode untuk mendapatkan nilai *depth* yang berasal dari proyeksi IR yang berpola [10]. Dapat dilihat pada gambar 2.8 merupakan visualisasi metode SR. Jika bola pada gambar mendekati *projector*, pola dari IR akan berubah. Dari perubahan tersebut dapat menghitung *depth*nya untuk setiap pixelnya. Metode ini hanya baik digunakan pada indoor.



Gambar 2.8 Visualisasi metode SR [10]

2. Stereo Depth

Metode ini juga menggunakan pancaran IR, namun berbeda dengan metode sebelumnya yang menggunakan pancaran IR yang terstruktur atau berpola, *stereo depth* menggunakan seluruh pancaran IR. Metode ini membutuhkan dua kamera yang terletak pada plane yang sama dengan jarak yang berdekatan. Dengan mengambil gambar dari kedua sensor kamera yang diketahui jarak antar keduanya, gambar tersebut akan dibandingkan dan akan didapatkan informasi depth [10]. Gambar 2.9 merupakan visualisasi dari metode ini.



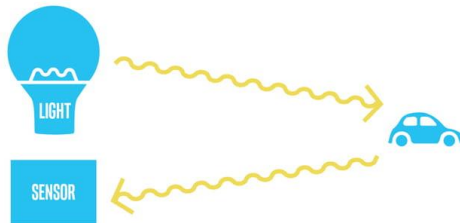
Gambar 2.9 Visualisasi metode stereo depth [10]

Metode ini memiliki konsep yang sama dengan cara kerja kedua mata manusia. Kelebihan dari metode ini adalah dapat digunakan baik di *outdoor* maupun *indoor*. Kemampuan pengukuran jarak dari metode ini bergantung dengan jarak antar dua sensor yang digunakan tersebut.

Metode ini juga digunakan para ahli astronomi dalam mengukur bintang atau hal lainnya yang berada di luar angkasa.

3. Time of Flight (ToF)

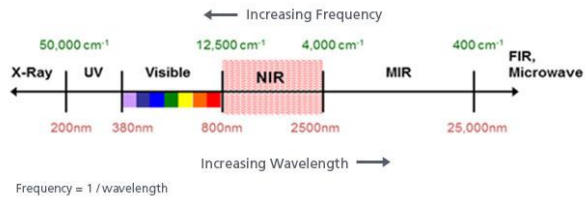
Pada metode – metode sebelumnya, menggunakan parameter yang telah diketahui sebelumnya, seperti pola dan jarak. Pada metode ini waktu perjalanan dari suatu cahaya juga sudah diketahui sebelumnya, dan informasi tersebut digunakan dalam menghitung jarak [10]. LIDAR sensor memanfaatkan metode ini untuk mendapatkan informasi jaraknya, dan biasanya digunakan pada *self-driving car*. Namun, kekurangan dari metode ini adalah cukup rentan dibandingkan sensor lainnya karena dapat terganggu dari cahaya yang dipancarkan oleh sumber lain. Gambar 2.10 merupakan visualisasi dari metode ToF.



Gambar 2.10 Visualisai ToF [10]

2.6.4 NIR

Near infrared merupakan gelombang elektromagnetik yang memiliki Panjang gelombang pada wilayah spektrum elektromagnetik sekitar 800 sampai 2500 nm, mendekati panjang gelombang dari cahaya tampak merah. Pada gambar 2.11 membantu dalam mengvisualisasi daerah Panjang gelombang dari *near infrared* dibandingkan dengan cahaya lainnya.



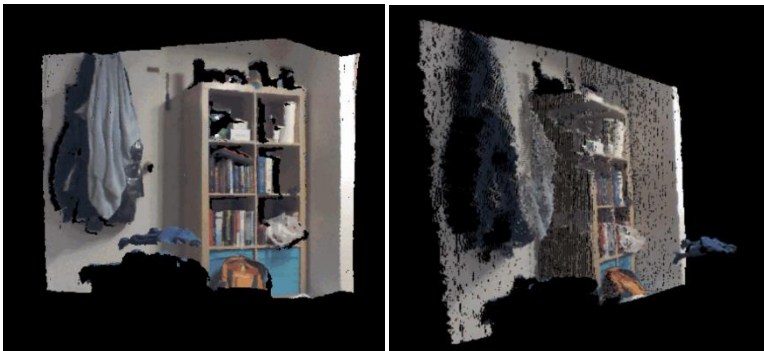
Gambar 2.11 Panjang gelombang Near-Infrared [11]

Cahaya ini juga sering di aplikasikan pada kamera, salah satunya adalah Kinect, dimana pada Kinect, NIR akan diproyeksikan ke segala sesuatu yang ada didepan sensor yang tidak dapat dilihat oleh mata manusia. NIR tersebut akan ditangkap kembali dengan Kinect dengan sensor khusus penangkap cahaya *infrared*.

Dari hasil proyeksi cahaya NIR, dapat dimanfaatkan untuk mendapatkan intensitas NIR yang merupakan hasil dari pantulan NIR oleh permukaan obyek apapun yang berada didepan Kinect. Data tersebut dapat dimanfaatkan untuk mengetahui material dari suatu obyek.

2.7 Point Cloud

Point Cloud merupakan kumpulan data yang terdiri dari titik – titik yang diartikan sebagai koordinat dalam dimensi ruang. Biasanya point cloud digunakan untuk mengetahui koordinat ruang dari sekumpulan informasi visual. Dimana informasi visual pada umumnya dinyatakan dalam bidang 2 dimensi akan diubah menjadi bidang 3 dimensi (ruang). *Point cloud* dapat digunakan dalam banyak tujuan, termasuk diantaranya adalah untuk membuat model CAD 3D, aplikasi visualisasi dan animasi, *rendering*, *3D scanning* dan lainnya.



Gambar 2.12 Point Cloud hasil dari Kinect V2 [13]

Salah satu pemanfaatan lainnya adalah dapat mengetahui koordinat dunia nyata (*world coordinate*) dari informasi visual yang ditangkap oleh kamera yang memiliki atau ditambah dengan informasi visual *depth* [12]. Dengan adanya point cloud yang mengandung koordinat dunia nyata,

posisi dari obyek – obyek juga dapat diketahui. Dengan begitu, dapat dimanfaatkan untuk membantu navigasi pada robot ataupun manipulasi yang dilakukan oleh robot. Gambar 2.12 menunjukkan data point cloud hasil pengolahan data yang telah dipetakan, ditangkap menggunakan Kinect V2.

2.8 OpenCV

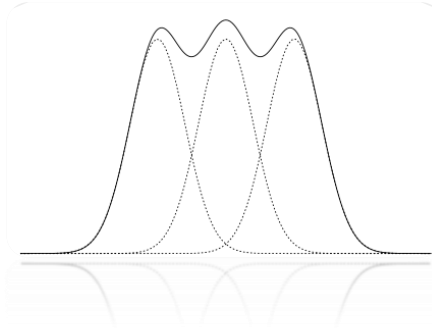


Gambar 2.13 Logo OpenCV

OpenCV (*Open Source Computer Vision Library*) merupakan suatu pustaka perangkat lunak sumber terbuka yang memudahkan pengguna berkulit di bidang computer vision dan pembelajaran mesin. Pustaka ini menyediakan lebih dari 2500 algoritma yang telah dioptimalkan. Algoritma – algoritma tersebut memudahkan pengguna dalam menyelesaikan pekerjaan seperti mencari dan mengenali muka, mengikuti pergerakan kamera, dan berbagai proses pengolahan gambar lainnya. Pustaka ini juga digunakan oleh berbagai perusahaan besar seperti Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota dan lainnya. OpenCV dapat digunakan pada berbagai bahasa pemrograman, seperti C++, Python, Java, dan MATLAB serta dijalankan di berbagai sistem operasi seperti Windows, Linux, Android, dan Mac OS. [14]

2.9 GMM

Gaussian Mixture Model merupakan model probabilistik untuk mewakili subpopulasi yang terdistribusi normal dalam populasi keseluruhan. Pada dasarnya, model campuran ini merupakan gabungan dari subpopulasi. Masing – masing subpopulasi ini tidak diketahui karena sudah tercampur satu dengan lainnya. Gambar 2.14 merupakan visualisasi dari model dari distribusi campuran.



Gambar 2.14 *Mixed Distribution*

Dengan menggunakan *Gaussian Mixture Model*, subpopulasi akan dicari dan dipelajari secara otomatis. Oleh karena itu, GMM merupakan salah satu metode dari pembelajaran tanpa pengawasan (*unsupervised learning*). GMM ini telah digunakan luas, sebagai salah satu contohnya adalah dalam mengekstraksi fitur dari data ucapan, pelacakan obyek dari beberapa obyek dan lainnya [1]. Pada persamaan 1 merupakan persamaan dari model gmm.

$$P(x) = \sum_{i=1}^K \phi_i \cdot N(x|\mu_i, \sigma_i) \quad (1)$$

Ket:

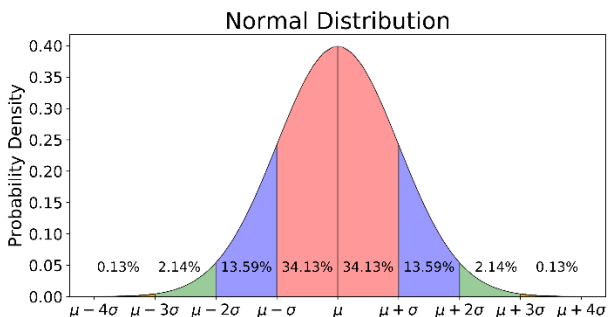
- N (*|*) = distribusi Gaussian (normal)
- K = banyaknya distribusi yang tercampur
- μ_i = mean dari data i,
- σ_i = standard deviasi dari data i
- ϕ_i = nilai pembebanan pada data i

Nilai dari parameter – parameter (ϕ_i, μ_i, σ_i) bisa didapat dengan menggunakan salah satu pendekatan dari algoritma EM (*expectation maximization*).

$$N(x|\mu_i, \sigma_i) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}}, \quad (2)$$

untuk $-\infty < x < \infty$

Pada persamaan 2 merupakan rumus distribusi gaussian (normal) dan gambar 2.15 merupakan visualisasi dari distribusi gaussian (normal) yang digunakan pada GMM.



Gambar 2.15 Distribusi gaussian (normal)

2.10 Metrics

Dalam melakukan prediksi, klasifikasi dan lainnya, model akan dibuat dan dicari yang terbaik sesuai dengan setiap permasalahan dan situasinya. Untuk mengetahui seberapa baik model yang telah kita buat, dapat dilakukan pengujian. *Metrics* atau pengujian model terdiri dari beberapa macam, salah satu diantaranya adalah *Precision*, *Recall* dan *F1-Score*. Untuk menghitung pengujian tersebut, diperlukan sebuah tabel *confusion matrix*, seperti pada tabel 2.5.

Pada tabel 2.5, terlihat bahwa terdapat empat kategori, yaitu benar positif, benar negatif, salah positif dan salah negatif. Keempat kategori nilai tersebut yang akan digunakan dalam perhitungan *precision*, *recall* dan *F1-Score*.

Tabel 2.5 Hubungan hasil prediksi dan sesungguhnya

		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Precision digunakan untuk mengetahui seberapa persis atau akurat model kita dalam mencari atau menebak benar positif. Pada persamaan (3) merupakan rumus dari *precision*.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (3)$$

Recall digunakan untuk mengetahui seberapa banyak benar positif yang dapat dicari atau ditebak oleh model kita. Pada persamaan (4) merupakan rumus dari *recall*.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (4)$$

F1-Score digunakan untuk mengetahui keseimbangan antara akurasi dan banyak benar positif yang dapat dicari atau ditebak oleh model kita. Pada persamaan (5) merupakan rumus dari *F1-Score*.

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall} \quad (5)$$

2.11 Ionic Framework

Ionic Framework merupakan kerangka kerja UI terbuka yang ditujukan untuk membuat *mobile apps* dan desktop dengan menggunakan teknologi berbasis web, seperti HTML, CSS dan JavaScript. Kerangka kerja ini difokuskan pada pengalaman pengguna *front-end* atau interaksi UI suatu aplikasi seperti pengontrolan, interaksi, gerakan dan animasi. Kerangka kerja ini dapat dipadukan dengan baik dengan pustaka atau kerangka kerja lain, seperti Angular atau dapat digunakan sendiri tanpa kerangka *front-end* menggunakan skrip sederhana.

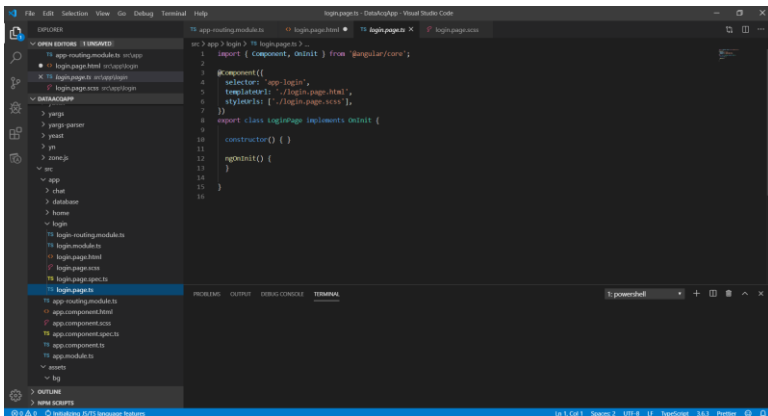
Saat ini, kerangka kerja Ionic memiliki integrasi resmi dengan Angular, React dan Vue. Beberapa kelebihan pada kerangka kerja ini adalah, pertama adalah lintas-platform. Dengan membuat satu proyek dengan menggunakan kerangka kerja Ionic, dapat dijalankan diberbagai platform, seperti iOS, Android, Desktop, dan Website. Kedua adalah

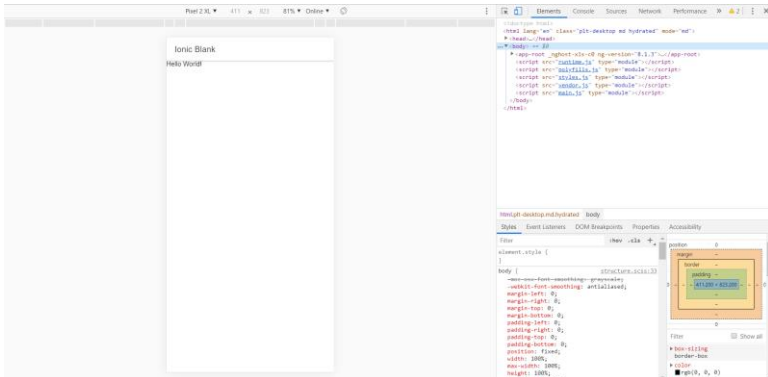
berbasis standar web. Kerangka ini dibangun diatas teknologi web yang handal dan terstandarisasi: HTML, CSS dan JavaScript, menggunakan API Web modern seperti *Custom Elements*, dan Shadow DOM.



Gambar 2.16 Logo Ionic Framework [15]

Oleh karena itu, komponen Ionic memiliki API yang stabil. Ketiga adalah lisensi. Kerangka kerja ini sudah terlisensi dan menjadi sumber terbuka dan gratis yang dirilis di bawah lisensi MIT, seperti proyek – proyek populer lainnya seperti JQuery dan Ruby on Rails. Ini berarti dapat digunakan dalam proyek pribadi atau komersial secara gratis. Pada gambar 2.17 merupakan penampilan dari awal dari Ionic Framework yang dibuka dengan menggunakan Visual Code Editor dan web-browser Google Chrome.[16]





Gambar 2.17 Penampilan Awal Ionic Framework

2.12 Firebase

Firebase adalah platform pengembangan aplikasi seluler dan web yang menyediakan sejumlah layanan kepada pengembang untuk membantu mereka mengembangkan aplikasi berkualitas tinggi, menumbuhkan basis pengguna, dan mendapatkan lebih banyak keuntungan [17]. Gambar 2.18 merupakan logo dari Firebase.



Gambar 2.18 Logo Firebase [17]

Salah satu layanan yang sangat populer dari Firebase adalah databasenya yang handal dan memiliki pilihan tipe database yang dibutuhkan oleh pengguna sesuai dengan kebutuhannya. Berikut beberapa layanan yang disediakan oleh Firebase, diantaranya adalah dalam membuat aplikasi, meningkatkan kualitas aplikasi ataupun dalam membantu mengembangkan bisnis pengguna [17]. Gambar 2.19 merupakan daftar layanan yang disediakan Firebase.

Membuat aplikasi

- Cloud Firestore
iOS </>
- ML Kit
iOS
- Cloud Functions
iOS </> C++
- Autentikasi
iOS </> C++
- Hosting
</>
- Cloud Storage
iOS </> C++
- Realtime Database
iOS </> C++

Meningkatkan kualitas aplikasi

- Crashlytics
iOS
- Pemantauan Performa
iOS
- Test Lab
iOS

Kembangkan bisnis Anda

- Analytics
iOS C++
- Prediction
iOS C++
- Pengujian A/B Firebase
iOS C++
- Cloud Messaging
iOS </> C++
- Remote Config
iOS C++
- Dynamic Links
iOS C++
- App Indexing
iOS
- Invites
iOS C++

Gambar 2.19 Layanan Firebase

-----*Halaman ini sengaja dikosongkan*-----

BAB III PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai perancangan sistem secara keseluruhan, meliputi perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras terdiri dari perancangan mekanik dan elektronik. Sedangkan perancangan perangkat lunak berhubungan dengan pemrograman untuk mengendalikan meja belajar, pengambilan data, interaksi pada android, dan pengolahan data untuk menghasilkan dataset dari obyek yang sudah dipelajari (terekstraksi).

3.1 Diagram Blok Sistem

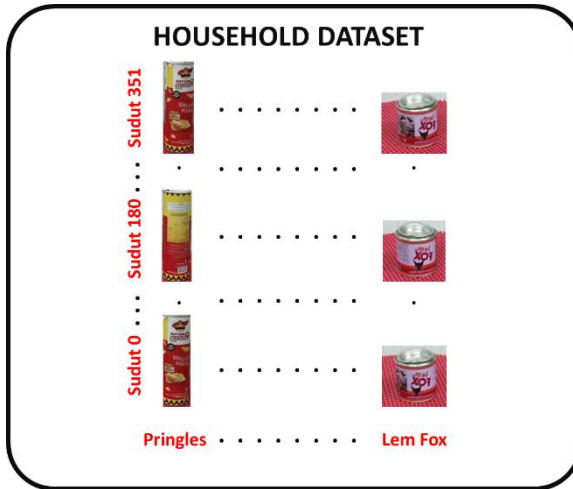
Pada penelitian ini, diagram blok keseluruhan sistem dapat dilihat pada gambar 3.1. Terdapat 5 blok utama pada sistem yang dibuat di penelitian ini, Antarmuka Pengguna, Pembelajaran Permukaan Meja, Pengambilan Data Obyek dan Pengenalan Obyek yang akan dijelaskan lebih lanjut pada subbab – subbab selanjutnya.



Gambar 3.1 Skema ilustrasi keseluruhan sistem

Secara umum sistem ini memiliki masukan, yaitu perintah manusia dengan menggunakan perantara antarmuka pengguna dari aplikasi yang dibuat dalam *smartphone* android. Interaksi percakapan melalui pesan dimanfaatkan untuk meningkatkan kenyamanan pengguna dalam memasukan perintah. Pemberian label juga dilakukan dengan memasukan nama obyek pada fitur percakapan pesan ini. Data informasi visual dari permukaan meja tanpa dan dengan obyek juga menjadi masukan sistem ini yang ditangkap menggunakan Kinect V2.

Keluaran dari sistem ini adalah dataset yang berisikan kumpulan informasi visual dari obyek yang telah dipelajari. Untuk setiap obyeknya, data akan diambil dari sisi – sisi yang berbeda. Pada penelitian ini, setiap satu obyek akan menghasilkan 40 data informasi visual (warna, *depth*, dan NIR) dengan perbedaan 9 derajat untuk setiap datanya. Gambar 3.2 merupakan visualisasi isi dari dataset diharapkan.



Gambar 3.2 Dataset keluaran sistem

3.2 Perancangan Perangkat Keras

Pada penelitian ini, perancangan meja belajar merupakan pekerjaan pertama yang dilakukan. Ada beberapa pertimbangan dalam perancangan perangkat keras untuk meja belajar ini:

1. Meja belajar diharapkan dapat berputar sebesar sudut yang cukup kecil (<15)
2. Meja belajar ini ditaruh di tempat yang relatif tetap layaknya meja belajar biasanya, namun tidak menutup kemungkinan akan dipindahkan sewaktu – waktu.
3. Perputaran meja belajar diharapkan dapat bersinkronisasi dengan pengambilan data yang dilakukan oleh Kinect.
4. Obyek di atas diharapkan tidak bergerak atau berpindah akibat perputaran dari meja.

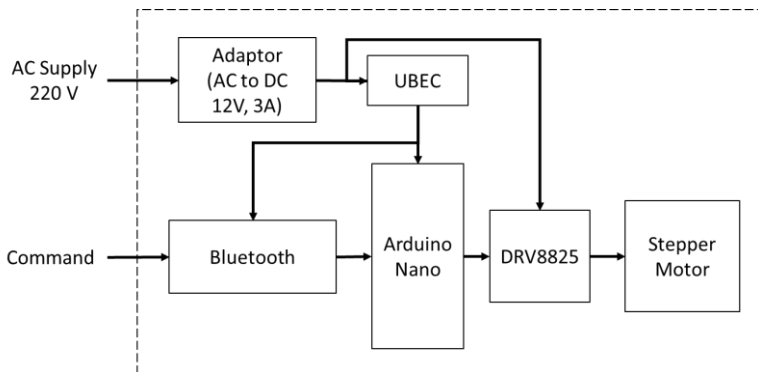
Berdasarkan empat poin di atas, meja belajar akan dirancang sebaik mungkin sehingga dapat mencapai pertimbangan tersebut. Perancangan ini dibagi menjadi dua bagian, perancangan mekanik dan perancangan elektronik.

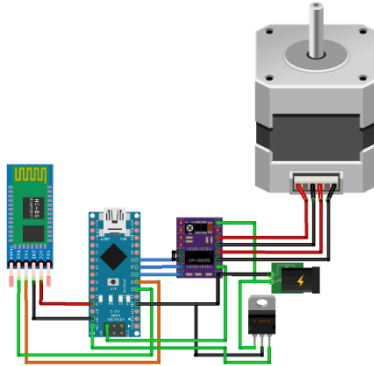
3.2.1 Perancangan Elektronik

Pada perancangan elektronik, meliputi pembuatan PCB yang akan memuat komponen – komponen yang diantaranya adalah motor stepper, *driver* motor stepper, Arduino Nano dan Bluetooth modul (HC-05).

Motor stepper digunakan karena dapat melakukan perputaran yang presisi; dapat berputar penuh, tidak seperti motor servo pada umumnya; memiliki torsi yang cukup baik pada kecepatan yang rendah, yang mana pada umumnya motor lain memiliki torsi yang rendah pada kecepatan yang rendah. Meskipun memiliki beberapa kekurangan seperti efisiensi yang lebih rendah dibandingkan motor lainnya, dimana dibutuhkan arus yang sangat besar.

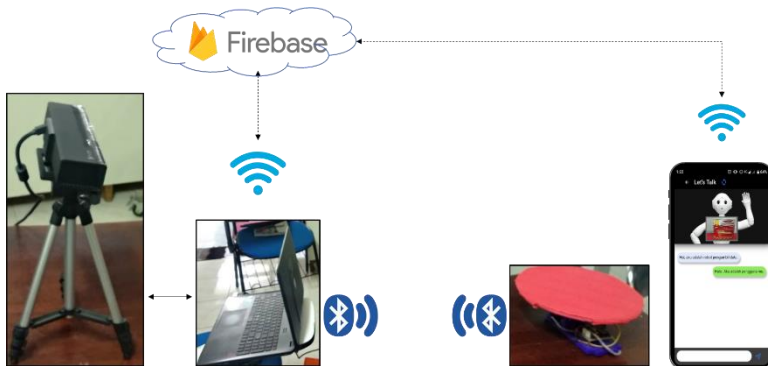
Karena meja ini tidak akan berpindah – pindah, rangkaian elektroniknya dapat menggunakan sumber listrik AC (220V) sebagai supply yang diubah kedalam sumber DC (12 volt) dengan menggunakan adaptor dan akan disambungkan dengan UBEC sebagai regulator DC 5 volt. Hal ini dilakukan agar pengambilan data tidak terganggu oleh kapasitas baterai, dimana pengambilan data beberapa obyek dapat berlangsung cukup lama (3 jam lebih). Perancangan rangkaian elektronik dapat dilihat pada gambar 3.3.





Gambar 3.3 Perancangan rangkaian elektronik

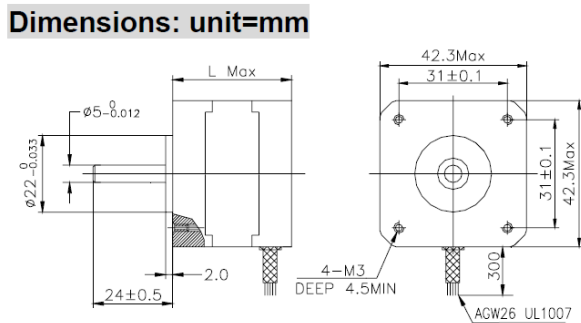
Pada penelitian ini, Bluetooth dimanfaatkan untuk dapat melakukan sinkronisasi dalam pengambilan data yang akan diambil oleh Kinect. Kinect akan dihubungkan dan dikendalikan melalui PC karena Kinect menggunakan USB 3.0. Sehingga meja belajar akan berkomunikasi dengan PC melalui Bluetooth. Sedangkan PC akan mengirimkan dan menerima data ke/dari database Firebase melalui saluran Internet. Firebase disini akan digunakan untuk melakukan komunikasi terhadap aplikasi *Smartphone* yang merupakan tampilan antarmuka pengguna utama. Gambar 3.4 merupakan visualisasi dari komunikasi yang digunakan pada sistem ini.



Gambar 3.4 Komunikasi Sistem

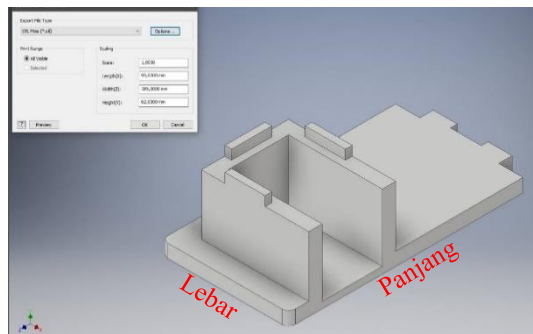
3.2.2 Perancangan Mekanik

Pada perancangan mekanik, meliputi bagian kerangka atau badan perangkat dan permukaan meja. Pada perancangan bagian kerangka perangkat, hal yang diperhatikan adalah membuat tempat untuk motor stepper, sehingga motor stepper dapat berposisi berdiri dan memiliki inersia, suatu usaha obyek untuk menolak perubahan terhadap keadaan gerakannya, yang baik sehingga tetap stabil pada posisinya. Gambar 3.5 merupakan bentuk fisik dan dimensi dari motor stepper.



Gambar 3.5 Bentuk dan dimensi motor stepper

Dengan bentuk fisik dari motor stepper dan memiliki lubang baut hanya pada satu sisi saja, yaitu atas. Diputuskan untuk membuat kerangka dengan bantuan mesin 3D – Printing, dengan desain 3D seperti pada gambar 3.6.

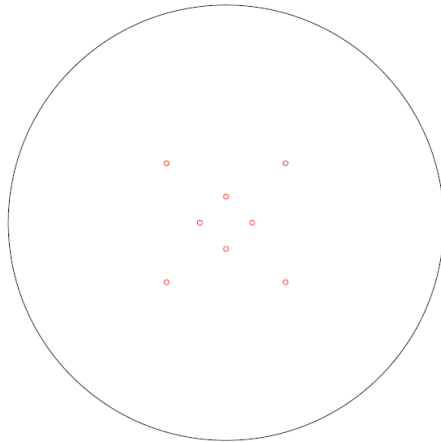


Gambar 3.6 Kerangka badan meja belajar

Bahan yang digunakan adalah filamen PLA. Spesifikasi ukuran dari kerangka badan ini adalah sebagai berikut:

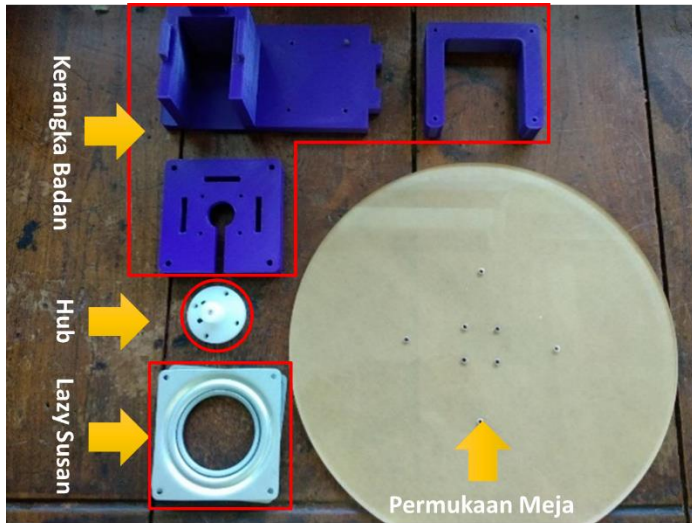
Ukuran kerangka badan
Panjang = 185 mm
Lebar = 95 mm
Tinggi = 62 mm

Pada perancangan permukaan meja, digunakan bentuk yang lingkaran untuk menghindari benturan terhadap obyek lain saat melakukan perputaran dan memiliki massa yang tidak begitu berat. Sehingga dipilih permukaan meja dibuat dengan menggunakan akrilik yang dipotong dengan laser dengan desain yang dapat dilihat pada gambar 3.7. Permukaan meja yang digunakan memiliki ukuran dengan diameter lingkaran 30 cm dan memiliki ketebalan sebesar 4 mm.

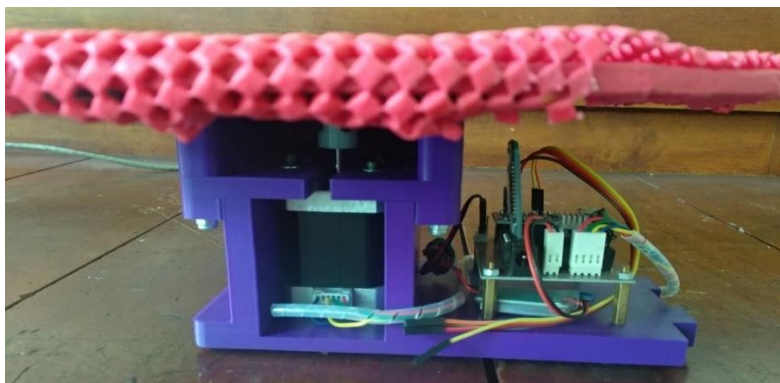


Gambar 3.7 Desain permukaan meja

Adapula komponen pembantu lainnya seperti hub yang digunakan untuk menghubungkan motor stepper dengan permukaan meja dan lazy susan yang digunakan untuk mengurangi beban dalam pemutaran permukaan meja. Pada gambar 3.8 terlihat seluruh komponen sebelum dirangkai dan gambar 3.9 merupakan hasil perakitan seluruh komponen (mekanik dan elektronik).



Gambar 3.8 Komponen perangkat keras meja belajar

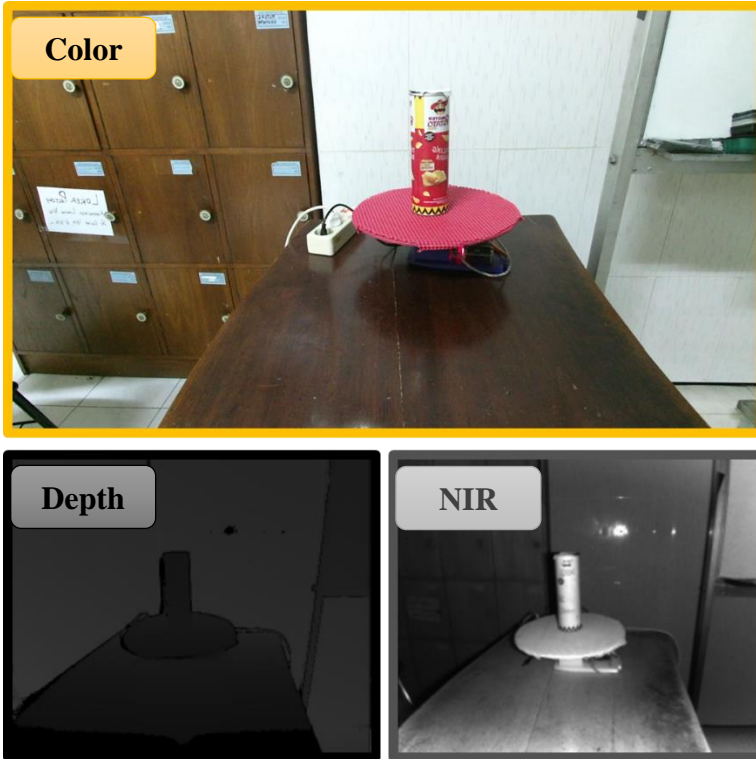


Gambar 3.9 Perangkat keras meja belajar

3.3 Pengambilan Data Berbasis Informasi Visual

Pengambilan data merupakan tahapan penting dalam sistem ini. Data yang diambil dengan menggunakan Kinect V2 merupakan data yang

berisikan informasi visual, berupa informasi visual warna, *Near-Infrared*, dan *Depth*. Data ini akan dimanfaatkan untuk pada proses pembelajaran obyek. Gambar 3.10 merupakan visualisasi dari data – data tersebut.



Gambar 3.10 Data berbasis informasi visual (RGB-D NIR)

Ketiga data tersebut akan dimanfaatkan baik dalam melakukan pembelajaran permukaan meja maupun dalam proses ekstraksi obyek. Resolusi data yang dihasilkan adalah 1920 x 1080 untuk informasi visual RGB dan resolusi 512 x 424 untuk informasi visual Depth dan NIR. Karena Kinect V2 menggunakan dua tipe sensor yang berbeda, yaitu kamera *Charge Coupled Device* (CCD) dan kamera *Time of Flight* (ToF) sehingga akan menghasilkan permasalahan yang harus diperhatikan. Dilanjutkan dengan adanya perbedaan pada resolusi. Dibandingkan

dengan mengkompensasi informasi dari depth, data RGB akan dipetakan sehingga informasi RGB, Depth, dan NIR dapat diintegrasikan. Dengan menggunakan *library* dari Kinect (*Kinect For Windows SDK*) untuk menghitung titik koresponden antara *color space* dan *depth space*.

Gambar 3.11 merupakan perbandingan antara visualisasi gambar warna dengan resolusi 1920 x 1080 dan 512 x 424 yang dipetakan dalam informasi visual *depth* dan NIR.



(a)



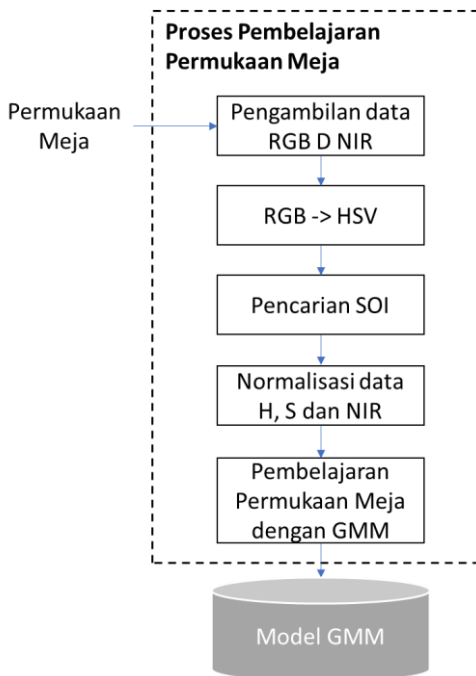
(b)

Gambar 3.11 Informasi visual warna dengan resolusi (a) 1920 x 1080 dan (b) 512 x 424 yang telah dipetakan

Informasi visual pada gambar 3.11 (b) selanjutnya akan disebut sebagai data warna yang telah terkalibrasi.

3.3 Proses Pembelajaran Permukaan Meja

Proses pembelajaran permukaan meja dilakukan untuk membedakan antara informasi visual permukaan meja dengan informasi visual yang bukan permukaan meja. Dimana pada awalnya tergabung dalam satu informasi visual yang sama. Informasi visual dari permukaan meja akan dipelajari dan diingat sehingga nantinya digunakan untuk membedakan obyek yang berada diatas permukaan dengan permukaan meja itu sendiri pada proses ekstraksi obyek. Proses tersebut dilakukan dengan menggunakan pendekatan dari *Gaussian Mixture Model* (GMM). Pada gambar 3.12, menunjukkan blok diagram proses pembelajaran permukaan meja.



Gambar 3.12 Blok diagram proses pembelajaran permukaan meja

Tahapan pertama pada proses ini adalah mencari ruang SOI yang akan digunakan dalam melakukan pembelajaran permukaan meja. Masukan dari pencarian SOI adalah gambar warna (RGB) yang mana dalam gambar tersebut terdapat permukaan meja belajar. Gambar warna ini sebelumnya akan diubah dari bentuk data RGB menjadi HSV. Proses ini dilakukan agar pencarian SOI dapat lebih handal, terutama pada suasana pencahayaan yang berbeda. Terkait bagaimana pencarian SOI akan dilakukan, akan dijelaskan lebih lanjut pada subbab 3.5. Pencarian SOI akan menghasilkan lokasi dan daerah dari permukaan meja belajar.

Setelah mengetahui lokasi dan daerah dari permukaan meja belajar, proses selanjutnya adalah pembelajaran permukaan meja dengan GMM. Masukan dari proses ini adalah informasi informasi visual berupa NIR dan HSV yang telah ternormalisasi beserta lokasi dan daerah dari permukaan meja belajar.

Dengan pendekatan *Gaussian Mixture Model*, pembelajaran ini dapat dilakukan dengan memasukan informasi informasi visual berupa data warna dan Near Infrared. Dari kedua data tersebut, akan dilakukan pembelajaran unsupervised learning dengan pendekatan GMM untuk dilakukan clustering, sehingga subpopulasi yang telah tercampur dapat dipisahkan, dalam permasalahan ini adalah memisahkan permukaan meja dengan yang bukan permukaan meja.

```
status = model2.learn(data2, 5, arma::maha_dist, arma::random_subset, 25, 100, 1e-10, true);
bool learn<T1>(const arma::Base<double, T1> &data, const arma::uword n_gauss,
              const arma::gmm_dist_mode &dist_mode, const arma::gmm_seed_mode &seed_mode, const arma::uword km_iter,
              const arma::uword em_iter, const double var_floor, const bool print_mode)
```

Gambar 3.13 Penggalan program GMM dan keterangannya

Pada gambar 3.13 merupakan penggalan program GMM beserta parameter yang digunakan dengan memanfaatkan *library* Armadillo pada bahasa pemrograman C++. Berikut beberapa penjelasan mengenai parameter yang digunakan:

n_gauss	= 5
dist_mode	= <i>Mahalanobis distance</i>
seed_mode	= random_subset
km_iter	= 25 (iterasi K-Means)
em_iter	= 100 (iterasi EM)
var_floor	= 1e-10 (nilai varian terkecil yang diizinkan)

Dengan n_gauss berarti banyaknya distribusi gauss. Pada kali ini, digunakan perhitungan Mahalanobis untuk mencari jarak terkecil dan digunakan juga K-Means dan EM, dimana K-Means dimanfaatkan untuk estimasi inisiasi dengan cepat, sedangkan EM digunakan untuk mengestimasi kemiripan dari masukan data terhadap distribusi yang ada.

Koefisien n_gauss dipilih 5 karena peneliti menganggap bahwa pada lingkungan rumah tangga terdapat kurang lebih terdapat 5 jenis bahan, yaitu padat(semen), plastik, kayu, besi, dan kertas, yang memantulkan pancaran NIR dengan intensitas yang berbeda. Sehingga, segmentasi obyek berdasarkan bahannya dapat dilakukan dan dimanfaatkan, bersama informasi visual warna, untuk melakukan segmentasi obyek yang merupakan meja dan yang merupakan bukan meja.

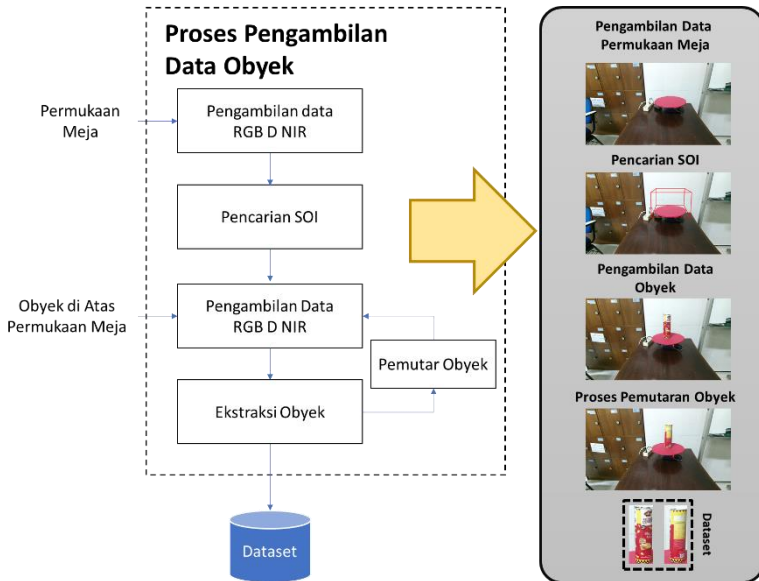
Sedangkan koefisien km_iter dan em_iter dipilih nilai 25 dan 100 karena merupakan nilai iterasi yang tidak terlalu banyak dan tidak terlalu dikit. Jika iterasi dilakukan terlalu sedikit, distribusi masih belum dipisahkan dengan baik. Sedangkan jika iterasi dilakukan terlalu banyak, distribusi akan terlalu terpisah dan mengakibatkan model terlalu *fit* dengan kondisi saat ini (saat pembelajaran). Hal ini mengakibatkan model tidak dapat bekerja dengan baik jika data mengalami perubahan meskipun sedikit saja.

Setelah melakukan pembelajaran dengan menggunakan gmm, model akan disimpan dalam bentuk model.gmm sehingga dapat digunakan pada proses ekstraksi obyek. Proses pembelajaran permukaan meja ini hanya dilakukan sekali jika permukaan meja tidak berubah, baik bentuk dan bahan. Sedangkan pada proses ekstraksi obyek, dilakukan berkali – kali sesuai dengan banyaknya obyek yang ingin diekstrak.

3.4 Proses Pengambilan Data Obyek

Pada tahap ini merupakan tahapan utama dari keseluruhan sistem. Tahap ini dapat disebut juga sebagai pembelajaran obyek (*object learning*). Proses ini akan menghasilkan obyek yang telah terekstraksi. Tahap – tahap pada proses ini diperlihatkan pada gambar 3.14.

Tahapan pertama pada proses ini adalah mencari ruang SOI seperti pada proses sebelumnya, yaitu pembelajaran permukaan meja. Pencarian pada kali ini akan digunakan dalam melakukan ekstraksi obyek. Pencarian ruang SOI akan dilakukan sebelum obyek ditaruh diatas permukaan meja belajar, sehingga masukannya adalah gambar warna (RGB) dari frame permukaan meja



Gambar 3.14 Blok diagram proses pengambilan data obyek

Setelah ruang SOI sudah ditemukan, pengambilan data beserta ekstraksi dari suatu obyek siap dilakukan. Proses dari pengambilan informasi visual akan dilakukan setelah obyek ditaruh di atas permukaan meja. Dilanjutkan dengan proses dari ekstraksi obyek dengan masukan hasil pengambilan informasi visual RGB, Depth dan NIR serta ruang SOI. Proses tersebut akan dijelaskan lebih lanjut pada subbab 3.4.1. Keluaran dari proses ini adalah informasi visual dari obyek yang telah terekstraksi.

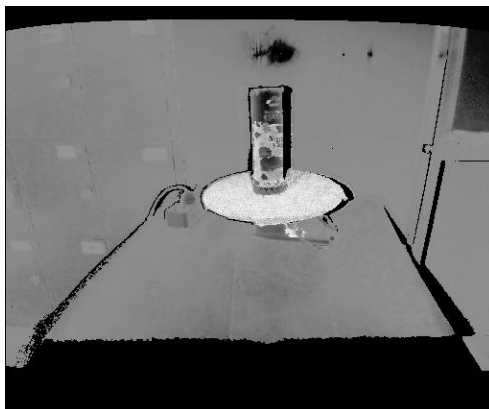
Selanjutnya, obyek akan diputar oleh meja belajar yang sudah didesain dapat berputar sesuai sudut yang diinginkan. Sehingga informasi visual dari suatu obyek dapat diambil dari berbagai sudut. Penjelasan dari pemutaran obyek akan dipaparkan pada subbab 3.4.2. Proses dari pengambilan data, ekstraksi obyek dan pemutaran obyek akan dilakukan secara berulang, hingga obyek telah diputar dalam satu putaran penuh dan mengakibatkan informasi visual dari satu obyek telah dikumpulkan dan diekstrak.

Setelah proses tersebut selesai, pemutar obyek akan bekerja untuk memutar sudut obyek dan informasi visual akan diambil lagi dan begitu pula proses selanjutnya akan berulang hingga 40 kali untuk mendapatkan data 360 derajat dari suatu obyek.

3.4.1 Ekstraksi Obyek

Tahap ekstraksi obyek merupakan tahapan untuk mendapatkan informasi visual obyek saja, dimana akan dilakukan pemisahan informasi visual permukaan meja dan informasi visual lainnya. Masukan dari tahapan ini merupakan informasi visual warna, informasi visual Depth dan informasi visual NIR. Informasi visual warna yang digunakan adalah informasi visual warna yang telah terkalibrasi. Keluaran dari tahapan ini merupakan informasi visual dari obyek yang telah terekstraksi, yang merupakan hasil akhir dari keseluruhan sistem pada penelitian ini.

Pertama, informasi visual warna yang telah terkalibrasi dan NIR akan dimasukkan pada model gmm yang telah didapat dari pembelajaran permukaan meja. Keluaran dari proses tersebut adalah probabilitas yang dipetakan dalam bentuk matriks. Matriks probabilitas ini berisikan nilai probabilitas, dimana nilai yang sangat tinggi diasumsikan sebagai permukaan meja belajar, sedangkan nilai yang rendah sebagai bukan permukaan meja. Gambar 3.15 merupakan visualisasi dari matriks hasil GMM. Jika dilihat pada gambar 3.15, semakin putih menandakan nilai probabilitas yang tinggi, sedangkan gelap menandakan nilai probabilitas yang rendah.

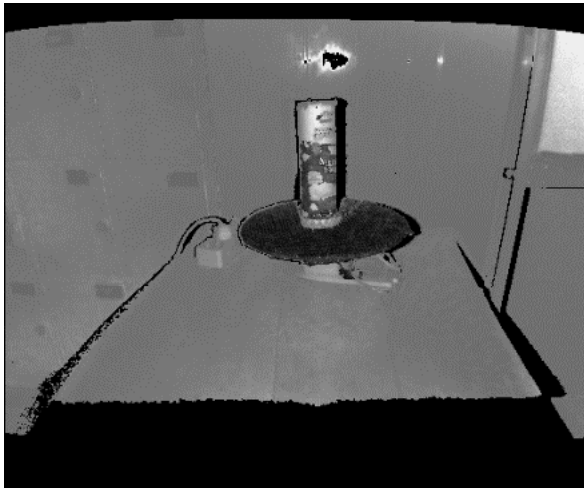


Gambar 3.15 Visualisasi matriks probabilitas permukaan meja

Selanjutnya untuk menghasilkan probabilitas yang tinggi pada bukan permukaan meja, dan probabilitas yang rendah untuk permukaan meja, nilai probabilitas tersebut akan diberlakukan perhitungan seperti pada persamaan (6).

$$P(\text{Non-table}) = 1 - P(\text{table}) \quad (6)$$

Hal tersebut dilakukan agar informasi visual dari permukaan meja tidak ikut terekstraksi. Jika dilihat pada gambar 3.16, semakin putih menandakan besarnya peluang dari pixel tersebut bukan merupakan pixel dari permukaan meja.



Gambar 3.16 Visualisasi matriks probabilitas bukan permukaan meja

Selanjutnya adalah pengimplementasian ruang SOI untuk mengeliminasi informasi visual yang tidak berada di ruang SOI. Pada gambar awal masih terdapat background dan obyek lainnya yang bukan merupakan obyek yang ingin kita ekstrak. Dengan mengetahui SOI yang telah dicari pada tahapan sebelumnya, informasi visual yang tidak berada pada SOI dapat tereliminasi seperti pada gambar 3.17, yang dinyatakan dalam bentuk nilai biner (0 atau 1).



Gambar 3.17 Informasi visual yang berada dalam SOI

Data biner hasil dari SOI tersebut digunakan untuk mengekstraksi obyek yang berada diatas permukaan meja saja. Dengan cara mengalikan matriks probabilitas informasi visual yang bukan merupakan permukaan meja dengan matriks biner hasil SOI, akan didapatkan matriks probabilitas dari obyek. Selanjutnya akan dilakukan thresholding pada matriks tersebut sehingga menghasilkan hasil yang terlihat pada gambar 3.18 (b).



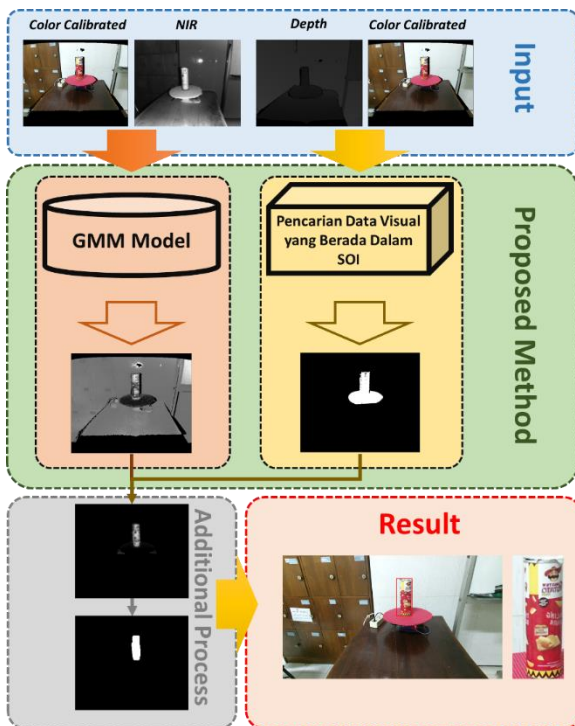
Gambar 3.18 Visualisasi matriks probabilitas obyek (a) dan matriks hasil thresholding (b)

Tahapan selanjutnya adalah memotong dan mengambil informasi visual dari obyek saja. Gambar 3.19 (a) dan gambar 3.19 (b) merupakan informasi visual warna kalibrasi dan informasi visual warna dalam HD beserta pengotakan untuk obyek pada kedua gambar tersebut. Dapat dilihat dari gambar 3.19 (a) dan (b) memiliki pergeseran pada pengotakan. Hal ini disebabkan oleh pemetaan dan konversi informasi visual warna

kalibrasi dan HD tidak sempurna, seperti yang telah dijelaskan pada subbab 3.3. Pergeseran ini tidaklah konstan untuk setiap pengestrakan obyek, sehingga tidak dapat dilakukan transformasi dilatasi.



Gambar 3.19 Hasil ekstraksi obyek pada gambar warna terkalibrasi (a) dan pada gambar warna HD (b)



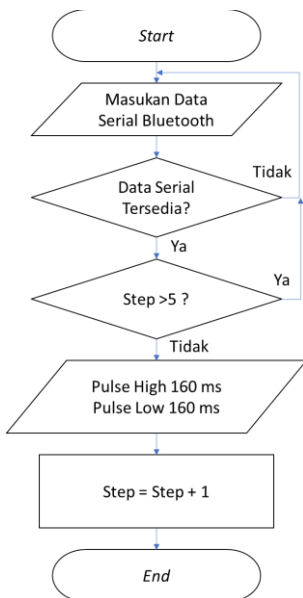
Gambar 3.20 Keseluruhan proses ekstraksi obyek

Agar lebih mudah dipahami, proses – proses sebelumnya akan digabungkan dan divisualisasikan pada gambar 3.20.

3.4.2 Pemutar Obyek

Pada tahap ini, meja belajar akan berputar sehingga obyek di atasnya juga ikut berputar. Proses ini akan diatur menggunakan mikrokontroler Arduino Nano, dengan masukan yaitu perintah untuk berputar yang diterima melalui Bluetooth. Sedangkan keluarannya adalah motor stepper yang berputar dengan sudut tertentu, pada penelitian ini motor stepper akan berputar sebesar 9 derajat.

Tahap ini sangat berkaitan dengan tahapan pengambilan data dan ekstraksi obyek. Hubungan ketiganya dapat dilihat pada gambar 3.14, dimana setelah ekstraksi obyek telah dilakukan, PC akan mengirimkan pesan ke Arduino melalui Bluetooth untuk berputar. Setelah meja belajar sudah diputar, pengambilan data dapat dilakukan kembali. Proses ini berulang kali dilakukan hingga informasi visual obyek dari berbagai sisi dalam satu putaran penuh telah terekstraksi semua. Gambar 3.21 merupakan diagram alir dari program pemutaran meja belajar.



Gambar 3.21 Diagram alir pemutaran meja belajar

Stepper akan siap berputar ketika mikrokontroler menerima pesan dari PC melalui komunikasi serial Bluetooth, sehingga stepper akan memutar meja sebesar 9 derajat setiap menerima pesan. Untuk menghasilkan perputaran meja dengan sudut 9 derajat, stepper akan berputar sebanyak 5 langkah. Untuk menggerakkan satu langkah pada stepper, pulsa HIGH dan LOW akan dikirimkan secara bergantian dengan delay 160 ms untuk setiap pengiriman pulsanya.

Pulsa ini akan dikirimkan ke driver stepper motor (DRV8825), dimana driver ini akan mengeluarkan mikrostep pulsa yang akan menjadi masukan motor stepper yang dibagi menjadi dua bagian, yaitu *A-Phase* dan *B-Phase*. DRV8825 akan mengeluarkan sinyal yang berbeda untuk masing – masing *phase*, yaitu dengan perbedaan sinyal yang tertinggal sebesar 45 derajat. Sinyal ini akan menggerakkan stepper yang dapat dikendalikan untuk setiap langkahnya.

3.5 Mencari SOI

Space of Interest (SOI) atau ruang perhatian merupakan salah satu metode yang diajukan pada penelitian ini, dimana istilah SOI adalah pengembangan dari *Region of Interest (ROI)* yang berarti daerah yang diperhatikan/diminati. Perbedaan mendasar dari kedua hal tersebut adalah dari dimensinya, dimana ROI memiliki ukuran 2 dimensi, sedangkan SOI sendiri memiliki ukuran 3 dimensi. Gambar 3.22 menunjukkan perbedaan dari ROI dan SOI.



(a)

(b)

Gambar 3.22 Visualisasi ROI (a) dan SOI (b)

SOI ini digunakan untuk membatasi ruang yang sangat besar ataupun hingga tidak terbatas. Keuntungan dari metode ini adalah untuk mengurangi data yang diproses yang tidak diperlukan karena obyek yang

dipelajari selalu berada di dalam SOI. Metode ini hanya dapat dilakukan jika memiliki informasi visual *depth* saja, sehingga pembatasan wilayah secara dimensi ruang dapat dilakukan.

Dalam kasus penelitian ini, dengan membatasi ruang pembelajaran akan memudahkan dan meningkatkan akurasi dalam melakukan ekstraksi fitur dari suatu obyek. Ruang perhatian (SOI) yang dimaksud adalah ruang diatas meja belajar, yang merupakan tempat penaruhan obyek yang ingin diekstrak atau dipelajari.

Dalam pencarian SOI, dibagi menjadi dua tahap. Pertama adalah pencarian titik tengah permukaan meja dan kedua adalah Inisiasi SOI. Kedua tahap tersebut akan dijelaskan lebih lanjut pada subbab dibawah ini.

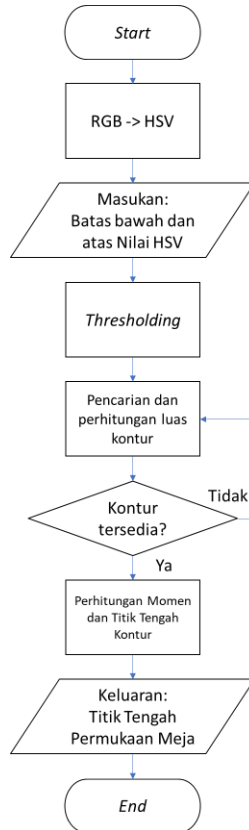
3.5.1 Pencarian Titik Tengah Permukaan Meja

Pada pengerjaan penelitian ini, pencarian titik tengah permukaan meja dibutuhkan untuk mempermudah dalam menginisiasi *Space of Interest*. Tanpa adanya pencarian ini, inisiasi lokasi atau jarak dari permukaan meja diperlukan. Akibatnya adalah sistem pada penelitian ini tidak dapat flexible karena dibatasi oleh kegiatan dalam mengukur dan memasukan lokasi atau jarak dari sensor Kinect dan permukaan meja.

Hal ini yang menyebabkan permukaan meja berwarna khusus, pada penelitian ini menggunakan warna merah dan tidak menutup kemungkinan warna lain dapat digunakan. Warna merah dipilih karena pada lingkungan rumah tangga peneliti (Lab B202 dan tempat tinggal peneliti) warna merah tidak begitu banyak dan tidak berukuran besar yang dapat mengganggu pencarian titik tengah permukaan meja terganggu.

Pencarian titik tengah dari permukaan meja dapat dilakukan dengan melakukan pencarian kontur yang didapat dari pencarian warna dari permukaan meja yang akan selalu sama. Dengan bantuan OpenCV, pencarian warna akan dilakukan dengan memberlakukan batas bawah dan atas dari warna yang dimiliki oleh permukaan meja. Gambar 3.23 merupakan diagram alur mengenai bagaimana pencarian titik tengah dari permukaan meja dapat dilakukan.

Untuk mencari kontur warna merah, dibutuhkan dua pasang batas bawah dan atas karena nilai HSV warna merah bertepatan pada perbatasan batas bawah dan batas atas *range* nilai HSV. Pada penelitian ini digunakan batas pertama, yaitu batas bawah HSV (10, 255, 255), dan batas atas (10, 255, 255). Sedangkan batas kedua, yaitu batas bawah (160, 100, 120) dan batas atas (179, 255, 255).



Gambar 3.23 Diagram alur pencarian titik tengah permukaan meja

Untuk pencarian titik tengah dari kontur, dilakukan dengan menggunakan pencarian moment dengan OpenCV. Moment yang dimaksud merupakan peminjaman istilah yang digunakan pada ilmu belajar fisika, yaitu titik pusat massa pada suatu luasan. Pada persamaan (7) merupakan rumus dari pencarian titik pusat dengan menggunakan moment. Dimana m_{10} merupakan titik pusat massa pada sumbu x, m_{01} merupakan titik pusat massa pada sumbu y dan m_{00} merupakan titik pusat massa dari keseluruhan luasan. Gambar 3.24 merupakan hasil dari pencarian titik tengah permukaan meja.

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \quad (7)$$



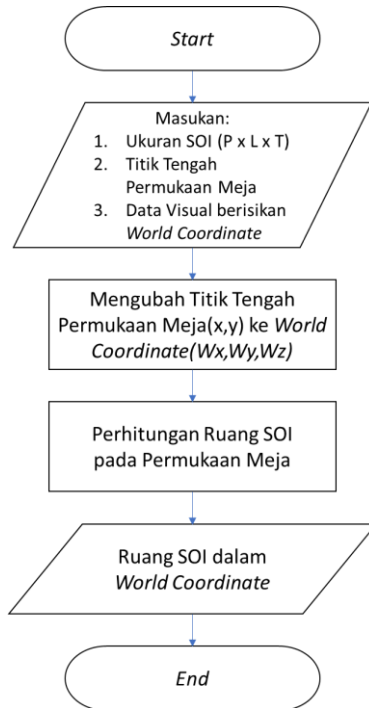
Gambar 3.24 Visualisasi hasil pencarian titik tengah meja

3.5.1 Inisiasi SOI

Pada tahap inisiasi SOI, penentuan ruang SOI akan dilakukan. Dengan memiliki masukan dari proses sebelumnya, yaitu titik tengah permukaan meja (x,y) satuan pixel. Karena ukuran dari permukaan meja sesungguhnya adalah Panjang x Lebar, sehingga ukuran ruang yang diinisiasi pada penelitian ini adalah Panjang x Lebar x Tinggi. Ilustrasi dari ruang ini dapat dilihat pada gambar 3.12 (b).

Penginisian dari ruang perhatian ini dilakukan pada koordinat dunia nyata (*world coordinate*). Hal tersebut dilakukan untuk meningkatkan ketepatan dari ruang perhatian yang lebih baik dinyatakan pada tiga dimensi dibandingkan pada dua dimensi. Koordinat dunia nyata bisa didapatkan dari pengolahan informasi visual *depth*. Dengan mengetahui titik tengah dari permukaan meja (dalam koordinat dunia nyata), ruang perhatian dapat dicari. Gambar 3.25 merupakan diagram alur dari inisiasi SOI.

Keluaran dari tahapan ini merupakan enam titik koordinat yang akan membentuk suatu balok yang akan digunakan untuk membatasi ruang dalam pengestrakan obyek.

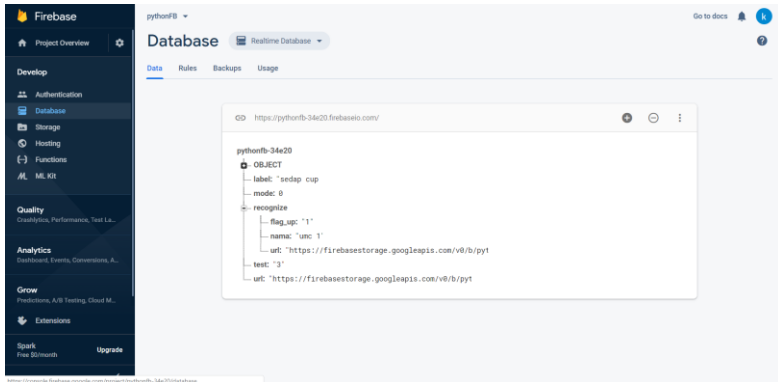


Gambar 3.25 Diagram alur penginisiasian SOI

3.6 Firebase

Pada penelitian ini, memanfaatkan servis yang diberikan disediakan Google sebagai *real-time database*. Servis ini digunakan untuk menghubungkan aplikasi *smartphone* yang dimanfaatkan untuk melakukan interaksi dalam melakukan perintah dalam pembelajaran, memberikan label nama dan kategori dari obyek – obyek yang dipelajari. Database dapat dijangkau dengan koneksi internet, dan juga pengguna aplikasi tidak terbatas hanya satu pengguna saja.

Pada gambar 3.26 merupakan tampilan dari *realtime database service* Firebase yang disediakan oleh Google.



Gambar 3.26 Tampilan *realtime database service* Firebase

3.7 Aplikasi *Smartphone* Antarmuka Pengguna

Pada penelitian ini aplikasi antar pengguna dimanfaatkan untuk memudahkan pengguna dalam menggunakan alat. Pada aplikasi ini, terdapat dua halaman penting, yaitu halaman percakapan dan halaman *storage*. Halaman percakapan merupakan halaman yang digunakan untuk memberikan masukan perintah terhadap sistem ini, sedangkan halaman *storage* digunakan untuk melihat obyek apa saja yang telah dipelajari oleh sistem ini, pemberian label kategori juga dilakukan pada halaman ini. Berikut penjelasan lebih jelas mengenai kedua halaman tersebut.

Pada halaman percakapan, peneliti menggunakan interaksi berubah percakapan pesan yang digunakan untuk memberikan perintah terhadap sistem ini. Pada gambar 3.27, merupakan penampilan antarmuka dari halaman percakapan. Sedangkan pada gambar 3.28 merupakan diagram alir sistem dengan masukan perintah.

Pada halaman percakapan, terdapat beberapa perintah yang dapat digunakan, yaitu perintah:

- “Lihat meja” : Perintah untuk sistem dalam pembelajaran permukaan meja sehingga menghasilkan model gmm
- “Belajar” : Perintah untuk sistem dalam pengambilan data
- “Update” : Perintah untuk sistem dalam menambahkan data yang baru pada halaman *storage*

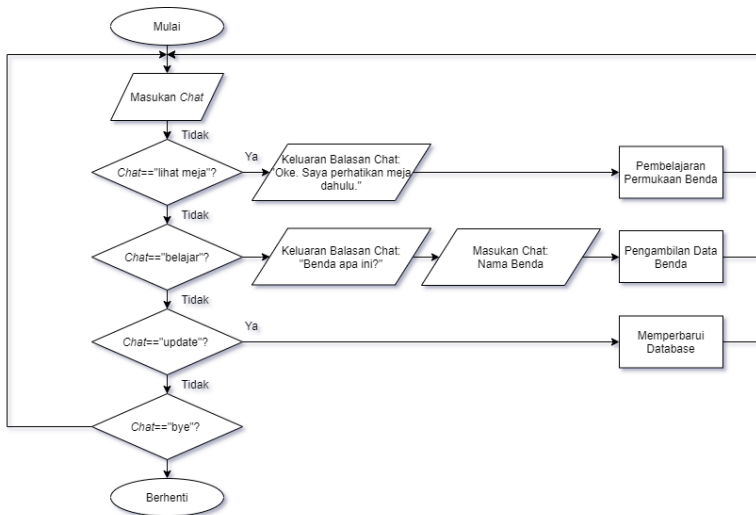
“Bye” : Perintah untuk membuat sistem menjadi *offline*

Tampilan dari penggunaan masing – masing perintah akan dilampirkan pada bagian lampiran.

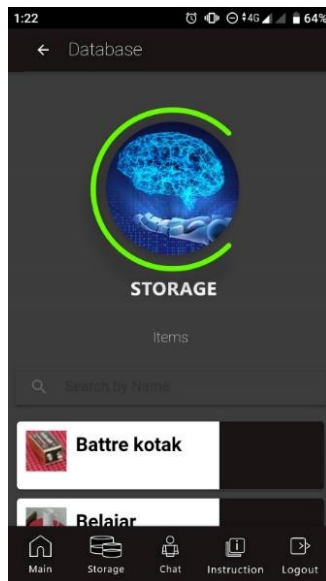


Gambar 3.27 Tampilan antarmuka pengguna halaman percakapan

Pada halaman *storage*, berisikan mengenai seluruh obyek yang telah dipelajari, ditampilkan dalam gambar beserta label nama dan kategorinya. Pada awal pembelajaran, pemberian label kategori belum dilakukan dan dapat dilakukan pada tampilan ini dengan menekan tampilan sebelah kanan dari obyek tersebut. Lalu, label kategori dapat dimasukkan. Label tersebut akan langsung masuk ke database Firebase dan dapat diambil datanya sehingga menjadi dataset yang lengkap. Berikut pada gambar 3.29 merupakan tampilan antarmuka dari halaman *storage*.



Gambar 3.28 Diagram alir sistem dengan masukan perintah



Gambar 3.29 Tampilan antarmuka pengguna halaman *storage*

Aplikasi ini dibuat dengan menggunakan bahasa Angular.js (Javascript) untuk pengaturan fungsinya, HTML dan CSS untuk tampilannya dan beberapa konfigurasi lainnya, yang mana semuanya dibuat dalam *Ionic Framework*. Kelebihan dari pengembangan aplikasi menggunakan Ionic merupakan aplikasi ini dapat digunakan dalam *smartphone* baik android maupun Iphone, dan juga dapat digunakan dapat diimplementasikan pada *website*. Pada aplikasi ini juga terdapat tampilan menu utama, instruksi dan login. Ketiga tampilan tersebut dapat dilihat pada bagian lempiran.

Dengan adanya bantuan aplikasi ini, pengguna dapat lebih mudah dalam memberikan label dengan penampilan yang interaktif dan dapat langsung melihat hasil pembelajaran obyek dengan melihat sampelnya. Sehingga jika terjadi kesalahan dalam pemberian label ataupun pembelajaran obyek, pengguna dapat mudah mengetahuinya dan memperbaiki data tersebut dengan cepat. Ketika sedang berada di luar rumah, pengguna juga dapat langsung mengetahui database dari sistem ini dengan bantuan aplikasi ini, sehingga pengguna dapat mengetahui obyek apa yang perlu dibeli dan yang tidak perlu karena sudah ada di rumah. Kedepannya juga, dengan adanya aplikasi ini, dapat diintegrasikan dengan sistem lainnya yang ada pada robot DSR, sehingga aplikasi ini dapat lebih berguna lagi, baik untuk mengontrol robot, memberikan perintah ataupun melakukan pemantauan.

-----Halaman ini sengaja dikosongkan-----

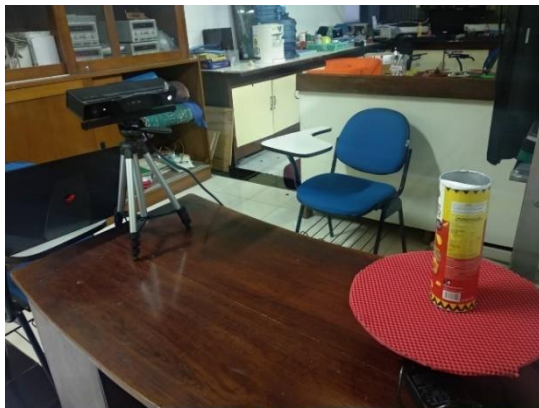
BAB IV

PENGUJIAN DAN ANALISIS

Pada bab ini membahas mengenai pengujian dan analisis dari sistem yang telah dirancang dan dijelaskan pada bab III. Pengujian ini dilakukan untuk melihat performa dari sistem dan menganalisis hasil pengujian ataupun permasalahan yang muncul pada saat pengujian berlangsung.



Gambar 4.1 Situasi eksperimen



Gambar 4.2 Situasi eksperimen



Berikut ada beberapa pengujian, diantaranya adalah pengujian perangkat keras, pengujian pengambilan data, pengujian sistem dengan tambahan tampilan antarmuka, dan evaluasi hasil pembelajaran obyek. Pada gambar 4.1 dan 4.2 merupakan gambar kondisi dan suasana lingkungan dari perangkat keras saat melakukan pembelajaran obyek







4.1 Pengujian Perangkat Keras

Pengujian yang pertama dilakukan adalah pengujian perangkat keras, yaitu meja belajar. Keandalan dari meja belajar dalam memutar obyek akan menjadi acuan pada pengujian ini. Harapan yang diinginkan adalah meja belajar dapat berputar dengan baik tanpa adanya gangguan. Beberapa faktor yang diperhatikan adalah apakah obyek akan berpindah tempat ataupun jatuh saat proses pemutaran terjadi, ataupun apakah meja belajar dapat memutar obyek yang memiliki massa lebih besar dibandingkan obyek pada umumnya serta durabilitas perangkat keras dalam bekerja.

Pengujian ini akan dilakukan dengan memutar obyek dengan ciri fisik (ukuran dan berat) yang berbeda. Meja akan berputar sebesar 9 derajat setiap dua detiknya hingga obyek berputar dalam satu putaran penuh. Posisi obyek sebelum diputar akan dibandingkan dengan posisi obyek setelah diputar satu putaran penuh. Karena sistem ini harapannya akan diterapkan pada robot DSR, obyek yang akan dipelajari sangat berkaitan dengan produk - produk yang berada di sekitar rumah tangga, seperti, makanan ringan, makanan cepat saji, sabun, pengharum ruangan dan lain – lainnya.

Tabel 4.1 Hasil pengujian perangkat keras sebelum dan setelah diputar

No.	Obyek	Sebelum diputar	Setelah diputar
1	<i>Earphone Mi</i> (Ringan dan kecil)		

2	Pringles (Ringan dan tinggi)		
3	Keranjang (Ringan dan Besar)		
4	Snack (Besar dan Berat)		

Pengujian dilakukan pada obyek – obyek berikut: *earphone mi* dengan ukuran yang kecil dan massa yang ringan, *pringles* dengan ciri ukuran yang sedang namun tinggi, *keranjang* dengan massa yang sedang namun berukuran besar, *snack* dengan ukuran yang besar dan memiliki massa yang berat. Hasil dari pengujian dapat dilihat pada tabel 4.1.

Berdasarkan pengujian yang telah dilakukan, peneliti melakukan analisis bahwa dalam pemutaran meja belajar dapat baik dilakukan dalam obyek – obyek yang memiliki ukuran kecil dan ringan karena adanya bantuan alas yang bertekstur. Tanpa adanya alas tersebut, obyek akan bergeser bahkan terjatuh dari meja belajar saat diputar. Sedangkan pada obyek yang berat seperti *snack*, meja pemutar dapat berputar sesuai dengan kebutuhan, namun pada prosesnya, motor stepper mengalami keberatan beban yang ditunjukkan dengan terlihatnya osilasi yang berlebih pada saat pemutaran obyek terjadi. Oleh sebab itu, peneliti membuat spesifikasi obyek yang diputar adalah tidak lebih dari satu Kg.

Pengujian kedua adalah durabilitas perangkat keras dalam bekerja. Pengujian ini dilakukan bersamaan dengan pengambilan data dilakukan. Dalam pengambilan data (termasuk pembelajaran obyek), sistem ini membutuhkan waktu kurang lebih selama 2 menit. Selama pengujian dan pengambilan data dilakukan, meja belajar ini digunakan selama kurang lebih 9 jam di ruangan lab B202. Perangkat keras ini masih memiliki performa yang baik dan tidak memiliki kendala dalam memutar obyek. Pengujian durabilitas maksimal dari perangkat keras tidak dilakukan karena perangkat keras hanya terdapat satu, namun dapat diasumsikan bahwa perangkat keras dapat bekerja dengan baik selama 9 jam ataupun lebih.

4.2 Pengujian Pengambilan Data

Pengujian pengambilan data dilakukan untuk mengetahui kehandalan sistem dalam pengambil data dari suatu obyek. Kehandalan dalam menghasilkan data hasil pembelajaran dari suatu obyek akan dijadikan acuan dalam pengujian ini. Pengujian dalam pencarian permukaan meja, pengujian pengambilan data pada berbagai jenis obyek, dan pengujian pengambilan data pada lingkungan dinamis dilakukan untuk mengetahui performa dari sistem.

4.2.1 Pengujian Pencarian Titik Tengah Permukaan Meja

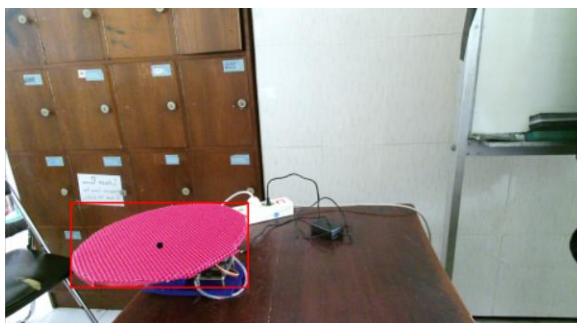
Pencarian titik tengah permukaan meja diperlukan untuk penginisiasian SOI yang dibutuhkan pada ekstraksi obyek. Pada pengujian ini, meja belajar diposisikan pada posisi yang berbeda – beda namun tetap dalam frame yang ditangkap oleh Kinect. Hasil pencarian titik tengah permukaan meja pada pengujian ini dapat dilihat pada gambar 4.3, 4.4, 4.5, 4.6 dan 4.7.



Gambar 4.3 Pengujian 1 pencarian titik tengah meja



Gambar 4.4 Pengujian 2 pencarian titik tengah meja



Gambar 4.5 Pengujian 3 pencarian titik tengah meja

Selanjutnya, pengujian dilakukan dengan memberikan beberapa obyek atau barang yang memiliki warna yang mirip dengan permukaan meja. Hasil pencarian titik tengah permukaan meja pada pengujian ini dapat dilihat pada gambar 4.6



Gambar 4.6 Pengujian 4 pencarian titik tengah meja



Gambar 4.7 Pengujian 5 pencarian titik tengah meja

Dari hasil pengujian diatas, peneliti melakukan analisis, bahwa pencarian titik tengah permukaan meja dapat handal dilakukan Meskipun

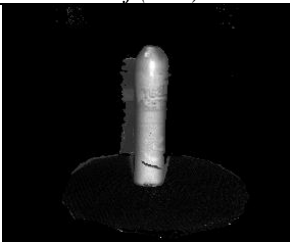

terdapat obyek berwarna merah lainnya seperti pada gambar 4.6, tetapi pencarian titik tengah meja belajar tetap dapat dilakukan. Dengan memilih kontur yang terbesar, pencarian tersebut tetap dapat dilakukan. Kekurangan dari metode ini adalah ketika adanya objek berwarna merah yang memiliki luasan kontur yang lebih besar, ketika objek memang lebih besar dari permukaan meja atau objek lebih dekat dibandingkan permukaan meja,

Pada gambar 4.7 kontur dari obyek berwarna merah menyatu dengan kontur permukaan meja. Meskipun begitu, pencarian titik tengah permukaan meja tetap dapat dilakukan dengan pergeseran yang tidak mempengaruhi proses ekstraksi obyek. Hal ini dikarenakan pada penelitian ini digunakan metode moment yang tidak menggunakan perhitungan geometri dalam pencarian titik tengah, melainkan menggunakan perhitungan titik pusat massa dengan luasan kontur yang dianggap massa pada pengolahan citra ini.

4.2.2 Pengujian Pengambilan Data pada Berbagai Jenis Obyek

Pada pengujian ini, dilakukan karena metode yang diajukan pada penelitian ini menggunakan GMM dengan masukan gambar warna dan NIR. Setiap obyek akan memiliki warna yang berbeda serta memiliki bahan yang berbeda – beda yang membuat data NIR juga berbeda. Hasil pengujian dapat dilihat pada tabel 4.2 yang terdiri dari tiga kolom utama, yaitu bahan dari obyek, pemetaan probabilitas obyek hasil model GMM dengan masukan informasi visual NIR dan warna, dan hasil ekstraksi obyek.

Tabel 4.2 Pengujian ekstraksi obyek pada berbagai bahan

No.	Bahan	<i>Prob. Obj (NIR)+ SOI</i>	Hasil Ekstraksi
1	Besi		

2	Plastik		
3	Kain		
4	Kramik		
5	Kertas		
6	Styrofoam		

Pada kolom ketiga (*Prob. Obj (NIR)+ SOI*), mengandung gambar yang menunjukkan seberapa mirip antara obyek dan permukaan meja. Jika nilai probabilitas semakin dekat, maka menunjukkan kemiripan yang sangat tinggi, Sedangkan semakin jauh nilai probabilitasnya (pada gambar ditunjukkan dengan intensitas warna yang sangat berbeda), menunjukkan ketidak-miripan obyek dengan permukaan meja.

Pada obyek dengan kain biru, kramik putih, besi putih menunjukkan ketidak-miripan antara obyek dan permukaan meja yang lebih tinggi dibanding dengan lainnya, sedangkan pada bahan Styrofoam putih dan kertas (kardus tipis) berwarna merah, menunjukkan ketidak-miripan terendah.

Pada tabel 4.2 juga dapat dianalisis bahwa tekstur dari suatu obyek juga mempengaruhi dalam ekstraksi obyek. Hal tersebut dibuktikan dengan adanya tekstur yang terlihat pada beberapa obyek seperti pada pengharum ruangan Bay Fresh, Mug dan Teh Kotak.

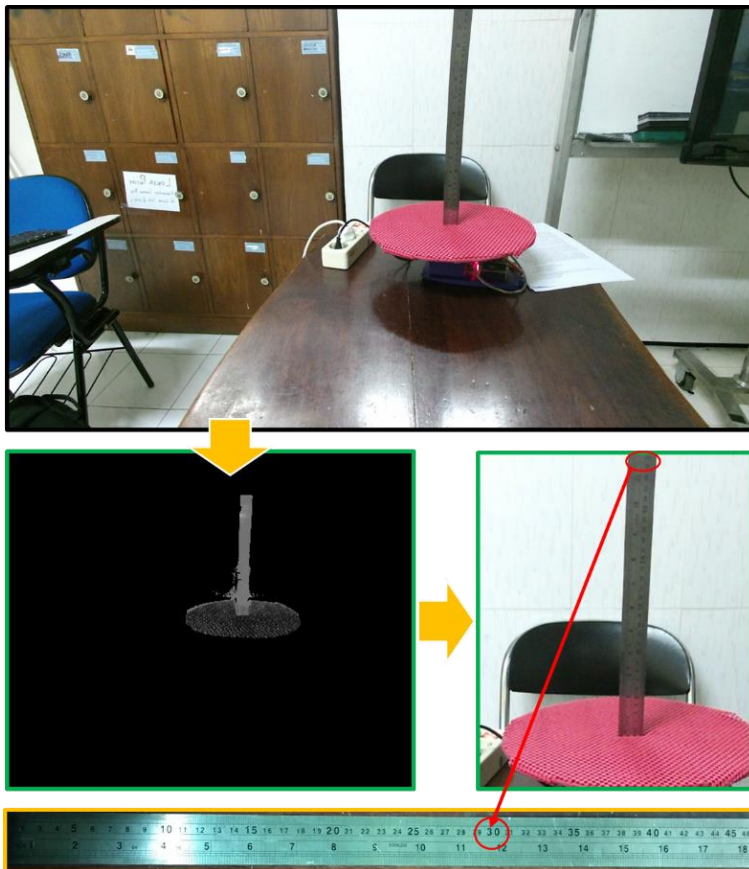
4.2.3 Pengujian Pengambilan Data pada Lingkungan yang Dinamis

Pengujian ini dilakukan untuk mengetahui kehandalan sistem dalam pengambil data pada lingkungan dinamis. Pada pengujian ini, lingkungan dinamis yang dimaksud adalah suasana lingkungan natural selayaknya lingkungan rumah tangga dan tidak dibatasi dengan kondisi – kondisi tertentu, seperti ruangan yang tidak diatur khusus ataupun kegiatan manusia yang dibatasi, seperti tidak boleh masuk dalam frame Kinect. Dengan harapan tersebut, pengujian ini dilakukan dengan berbagai kondisi seperti berikut:

Pertama, pengujian dilakukan untuk mengetahui apakah SOI bekerja dengan baik atau tidak. Dengan inisiasi ukuran SOI, yaitu Panjang, Lebar dan Tinggi yang berukuran 30cm, 30cm dan 30cm, dengan kemiringan sudut kamera, Tilt sebesar 20 derajat. Dengan memanfaatkan penggaris berukuran 50cm, pengujian dilakukan dengan mengambil data dari penggaris, lalu melihat hasil potongan dari sistem ini, seperti pada gambar 4.8.

Pada pengujian ini, didapatkan hasil pemotongan akibat dari metode SOI sebagai berikut: Tinggi sebesar 30cm (didapat dari potongan penggaris), sedangkan panjang dan lebar sebesar 30cm dan 30cm yang didapat dari ukuran permukaan meja. Peneliti melakukan analisis, bahwa

penginisiasi SOI ini memiliki sedikit error, namun masih dapat ditoleransi untuk melakukan pengambilan data yang handal. Error ini peneliti asumsikan akibat dari pengambilan data Depth dari Sensor yang sedikit meleset ataupun dari penginisiasian kemiringan tilt yang tidak presisi.



Gambar 4.8 Pengujian SOI

Pengujian selanjutnya adalah dengan meja belajar yang diletakan disekitar obyek lainnya tanpa adanya pengkondisian obyek lainnya.

Pengujian ini dilakukan diberbagai kondisi yang berbeda seperti pada gambar 4.9.



Gambar 4.9 Pengujian 1 Ekstraksi Obyek

Pada pengujian kedua, peneliti dapat melakukan analisis bahwa obyek sekitar tidak akan mempengaruhi proses dalam pembelajaran obyek. Hal ini dikarenakan pengaplikasian metode SOI, sehingga hasil ekstraksi dapat tepat dilakukan pada meja belajar saja.

Pengujian selanjutnya adalah pengujian dengan adanya kegiatan manusia yang berada dalam *frame* Kinect saat pengambilan data dilakukan. Pengujian pertama yang dilakukan adalah adanya manusia yang berada dalam *frame* Kinect, hasil pengujian ini dapat dilihat pada gambar 4.10.



Gambar 4.10 Pengujian 2 ekstraksi obyek

Pengujian kedua dilakukan dengan adanya manusia yang berada dibelakang meja belajar dan beraktivitas seperti biasa. Hasil pengujian ini dapat dilihat pada gambar 4.11.

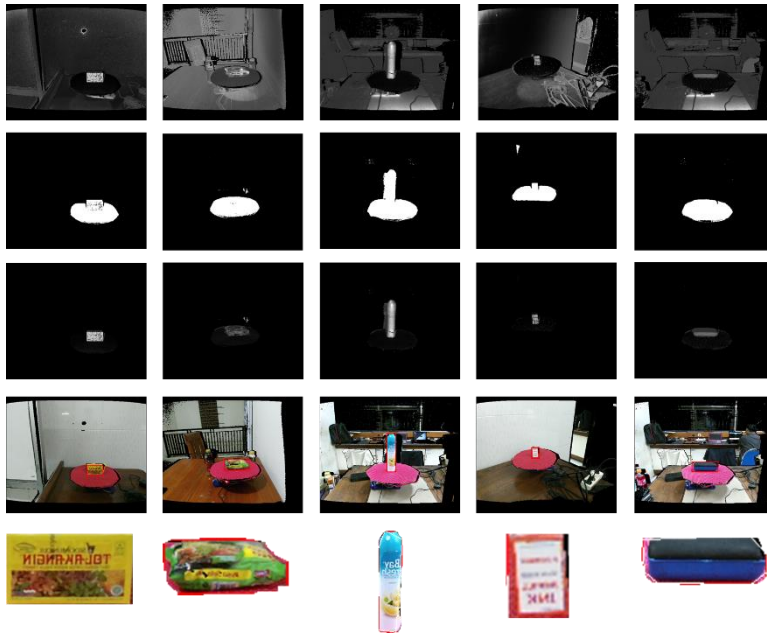
Berdasarkan tiga pengujian terakhir, peneliti melakukan analisis bahwa sistem dapat handal dalam mengambil data dalam berbagai lingkungan dinamis, seperti peletakan obyek lain yang tidak beraturan dan adanya objek yang diam ataupun bergerak yang berada dibelakang meja belajar. Hal ini dapat memungkinkan sistem dapat bekerja di lingkungan rumah tangga yang sangat tidak beraturan, baik karena objek sekitar yang tidak beraturan ataupun adanya kegiatan manusia yang beraneka ragam. Adapula beberapa contoh pengujian yang dilampirkan pada bagian lampiran.



Gambar 4.11 Pengujian 3 ekstraksi obyek

4.3 Evaluasi Hasil Pembelajaran Obyek

Pada pengujian ini, hasil pembelajaran dari obyek akan diuji berdasarkan keluaran hasil dari proses sistem yang telah dijelaskan pada bab sebelumnya. Pada gambar 4.12 merupakan contoh proses yang dikerjakan oleh sistem saat melakukan ekstraksi obyek, dimana pada baris pertama merupakan keluaran dari GMM model, dimana merupakan gambar yang berisikan nilai probabilitas dari objek yang bukan merupakan permukaan meja yang sebelumnya telah dipelajari. Pada baris kedua merupakan gambar dari SOI. Pada baris ketiga merupakan gambar dari probabilitas obyek yang berada di atas permukaan meja. Pada baris keempat, merupakan gambar warna keluaran sistem, dimana obyek yang dipelajari dikotaki dengan warna merah. Dan pada baris terakhir, merupakan hasil pemotongan dan pengambilan gambar obyeknya saja melalui blob yang telah dihasilkan.



Gambar 4.12 Sampel proses ekstraksi obyek

Peneliti melakukan analisis bahwa hasil yang dihasilkan sudah cukup baik, dilihat dari terlihatnya perbedaan yang cukup jelas antara probabilitas pada obyek dan permukaan meja, pencarian SOI yang cukup akurat, dan hasil ekstraksi yang dihasilkan. Meskipun hasil ekstraksi yang dihasilkan terkadang mengalami kesalahan dan kepresisian yang menurun, bergantung dari obyek yang akan dipelajari, dimana faktor yang membuat hal tersebut terjadi adalah tekstur, warna, dan material dari obyek yang sedang diekstrak.

Pengujian selanjutnya adalah pengukuran volume dari obyek yang dipelajari. Salah satu keluaran dataset yang dihasilkan adalah nilai *pointcloud* yang berisikan koordinat dunia nyata. Dari data tersebut, dapat juga dihasilkan sebuah pengukuran volume, dengan cara mengukur panjang lebar dan tinggi obyek yang dipelajari yang diukur berdasarkan data visualnya. Pengujian yang pertama adalah pengujian volume obyek

yang terukur dan realita. Didapatkan hasil yang terlihat pada tabel 4.3. Sedangkan pada tabel 4.4 merupakan pengukuran volume sebenarnya dari obyek yang diuji.

Tabel 4.3 Tabel volume obyek yang terukur dengan beberapa variabel jarak

No	Nama	Panjang(cm)	Lebar(cm)	Tinggi(cm)	Volume(cm ³)
1	Box 0.5m	20.252	11.990	6.180	1500.578
2	Box 1m	18.758	9.430	5.824	1030.156
3	Box 1.5m	18.541	9.781	4.381	794.438
4	Gatsby 0.5m	7.995	8.044	6.657	428.097
5	Gatsby 1m	7.372	6.759	5.618	279.935
6	Gatsby 1.5m	7.515	7.964	8.350	499.710
7	Isi tinta 0.5	4.894	4.747	6.787	157.694
8	Isi tinta 1m	5.422	5.357	7.480	217.241
9	Isi tinta 1.5m	4.558	4.002	5.777	105.381
10	Lifeboy 0.5	5.818	3.731	13.172	285.915
11	Lifeboy 1m	6.127	4.839	13.155	390.077
12	Lifeboy 1.5m	5.827	3.883	13.110	296.610
13	Oreo 0.5m	7.755	7.567	11.551	677.837
14	Oreo 1m	8.033	8.701	12.707	888.203
15	Oreo 1.5m	7.559	7.443	11.360	639.145
16	Spray 0.5m	7.989	7.611	15.678	953.250
17	Spray 1m	8.247	8.088	16.399	1093.791
18	Spray 1.5m	7.739	7.260	14.732	827.697

Tabel 4.4 Tabel volume obyek sebenarnya

No	Nama	Ukuran Sebenarnya Obyek			
		Panjang (cm)	Lebar (cm)	Tinggi (cm)	Volume (cm ³)
1	Box	20.9	11	4.5	1034.55
2	Gatsby	6.3	6.3	5	198.45
3	isi tinta	4	4	6.5	104
4	Lifeboy	5.4	3	13.5	218.7
5	Oreo	8.5	8.5	13	939.25
6	Spray	8.2	7.4	17	1031.56

Pengujian ini dilakukan terhadap beberapa obyek yang berbeda ukuran serta diuji pada jarak yang berbeda – beda, yaitu 0.5 meter, 1 meter dan 1.5 meter untuk setiap obyeknya. Berdasarkan hasil tersebut, peneliti melakukan perhitungan kesalahan dalam pengukuran dengan menggunakan RMSE dengan membandingkan hasil yang terukur dengan

hasil volume sebenarnya. Hasil tersebut dapat dilihat pada tabel 4.5 berikut.

Tabel 4.5 Perhitungan kesalahan pengukuran

No	Nama	RMSE (Ukuran Obyek Terukur, Ukuran Sebenarnya)			
		Panjang (cm)	Lebar (cm)	Tinggi (cm)	Volume(cm ³)
1	Box	1.716	0.600	0.962	55.093
2	Gatsby	1.327	1.289	1.875	199.486
3	isi tinta	0.958	0.702	0.181	51.760
4	Lifeboy	0.524	1.151	0.354	104.565
5	Oreo	0.718	0.596	1.127	208.957
6	Spray	0.209	0.253	1.397	77.314
Rata-rata		0.909	0.765	0.983	116.196

Peneliti melakukan analisis, bahwa berdasarkan hasil kesalahan yang terlihat pada tabel 4.5, didapat bahwa hasil pengukuran memiliki kesalahan, yaitu rata – rata sebesar kurang dari satu cm. Peneliti melakukan analisis, bahwa hasil tersebut sangat baik karena pengukuran tersebut didapatkan hanya berdasarkan informasi visual yang ditangkap oleh sensor Kinect, tidak ada tambahan sensor atau alat ukur lainnya.

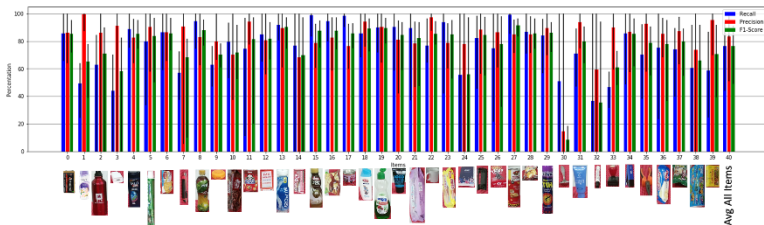
Pada pengujian selanjutnya, pengajuan metode dalam mempelajari obyek akan diuji



Gambar 4.13 Contoh *bounding box* ekstraksi obyek

Dengan melakukan pengambilan data dari 40 obyek dengan ketelitian sudut yaitu 9 derajat, akan menghasilkan foto sebanyak 40 obyek x 40 foto yaitu 1600 foto. Evaluasi dilakukan dengan membandingkan *bounding area* hasil dari ekstraksi obyek yang dilakukan oleh sistem ini dengan *bounding area* hasil dari pemberian label (pengotakan) secara manual (penyeretan kursor pada daerah yang merupakan obyek). Gambar 4.13 dengan kotak yang berwarna merah merupakan *bounding area* hasil dari ekstraksi obyek dan sedangkan kotak yang berwarna biru *bounding area* dengan cara manual.

Pada pengujian ini juga membandingkan 4 pendekatan yang berbeda untuk melakukan ekstraksi obyek, yaitu metode GMM+ *color* (RGB), *Depth only*, GMM + *Depth* + RGB, dan GMM + *Depth* + HSV. Evaluasi yang dilakukan disajikan dalam bentuk *recall*, *precision* dan *F1-Score*. Gambar 4.14 merupakan hasil evaluasi dari pendekatan metode GMM + *Depth* + HSV, sedangkan hasil evaluasi dari pendekatan lainnya dilampirkan pada bagian lampiran. Tabel 4.6 merupakan rata – rata dari hasil pembelajaran 40 obyek menggunakan empat pendekatan yang berbeda.



Gambar 4.14 Evaluasi ekstraksi obyek

Tabel 4.6 Perbandingan hasil evaluasi ekstraksi obyek

No	Method	Recall [%]	Precision [%]	F1-Score [%]
1	GMM + Color (RGB)	97.69 ± 1.25	1.65±0.59	3.22±1.13
2	Depth Only	95.06±6.88	14.09±8.55	22.21±10.64
3	GMM + Depth + RGB	91.09±15.13	64.97±26.75	69.01±24.27
4	GMM + Depth + HSV	76.24±25.00	83.18±23.85	76.43±20.96

Seperti yang telah dijelaskan pada bab dua, *recall* merupakan hasil dari banyaknya daerah benar yang tertangkap tanpa membandingkannya dengan daerah yang salah, sehingga dengan menggunakan metode pertama menjadi hasil yang terbaik pada *metrics recall* karena hasil dari metode pertama hampir selalu menanggapi semuanya benar (dapat dilihat pada bagian lampiran). Sedangkan *metrics precision* mengukur seberapa presisi metode kita dengan memperhatikan daerah yang benar dan daerah yang tidak benar juga (TP dan FP). Dan pada *metrics FI-Score* merupakan kombinasi dari *metrics Recall* dan *Precision*.

Dari hasil tersebut, dapat disimpulkan bahwa dengan menggunakan metode GMM + *Depth* + HSV merupakan hasil yang terbaik, disamping karena memiliki nilai keseluruhan (*FI-Score*) terbaik sistem ini juga lebih mementingkan hasil *precision* yang lebih baik dibandingkan dengan *recall*. Untuk itu, dalam pembelajaran obyek digunakan metode ini sehingga menghasilkan hasil dataset gambar yang terbaik.

4.4 Hasil Akhir Dataset

Hasil akhir dari penelitian ini merupakan dataset yang berisikan informasi – informasi dari obyek yang dipelajari, baik informasi visual maupun informasi berupa label. Dataset pertama adalah berupa kumpulan label dari nama obyek, kategori, dan volume yang disimpan dalam bentuk .csv. Tabel 4.7 merupakan tabel dari label nama, kategori dan volume dari obyek – obyek.

Tabel 4.7 Daftar label dan kategori dataset obyek

No	Nama	Kategori	Volume(cm ³)
1	Battre kotak	Elektronik	20.798
2	Bay fresh	Pembersih	669.330
3	Bear brand	Minuman	351.493
4	Beng-beng	Makanan	130.265
5	Biore	Pembersih	351.894
6	Botol pln	Lainnya	2596.026
7	Cabang	Elektronik	244.905
8	Charger	Elektronik	68.584
9	Chocolatos	Minuman	218.817

10	Clear	Pembersih	497.417
11	Closeup	Pembersih	229.384
12	Cracker	Makanan	682.165
13	Cutter	Atk	131.398
14	Floridina	Minuman	572.936
15	Freshtea	Minuman	841.515
16	Fruittea	Minuman	711.237
17	Gatsby rubber	Pembersih	275.805
18	Geadset redmi	Elektronik	82.085
19	Golda	Minuman	714.797
20	Good time	Makanan	1100.519
21	Hallo panda	Makanan	573.065
22	Hit	Pembersih	617.413
23	Indomie grg	Makanan	183.626
24	Indomie kuah	Makanan	236.074
25	Isi tinta	Atk	105.381
26	Isoplus	Minuman	490.657
27	Keranjang	Atk	10635.996
28	Kitkat	Makanan	79.832
29	Kopiko	Minuman	394.197
30	Kusuka	Makanan	1190.581
31	Lem fox	Atk	224.558
32	Lifeyboy	Pembersih	285.915
33	Mama lime	Pembersih	1124.819
34	Mens biore	Pembersih	935.398
35	Mitu	Pembersih	657.435
36	Mouse	Elektronik	436.496
37	Mug	Lainnya	1445.805
38	Nabati	Makanan	533.754
39	Notebook biru	Atk	586.809

40	Nu greent	Minuman	583.069
41	Oreo	Makanan	888.203
42	Parfum getsby	Pembersih	618.453
43	Pembolong krts	Atk	642.156
44	Penghapus papan	Atk	689.251
45	Pocari	Minuman	1255.579
46	Pocky	Makanan	679.049
47	Pringles	Makanan	1040.928
48	Qtela	Makanan	1641.476
49	Selotip	Atk	528.564
50	Semprotan auto	Pembersih	1031.560
51	Sendok	Lainnya	40.859
52	Senka	Pembersih	746.337
53	Snack	Makanan	1261.579
54	Spidol	Atk	99.098
55	Stample	Atk	138.854
56	Stapples	Atk	231.355
57	Stella air	Pembersih	903.625
58	Steroform	Lainnya	1090.855
59	Tang	Atk	411.378
60	Tango	Makanan	546.354
61	Taro	Makanan	693.445
62	Teh gelas	Minuman	392.893
63	Teh kotak	Minuman	462.039
64	Teh kotak ung	Minuman	514.736
65	Teh pucuk	Minuman	409.279
66	Tempat makan	Lainnya	1375.587
67	Tisu	Pembersih	605.785
68	Tmpt pensil	Atk	2254.900
69	Tolak angin	Lainnya	379.889

70	Ultramilk	Minuman	297.912
71	Viks	Lainnya	134.022
72	Yupi	Makanan	678.906

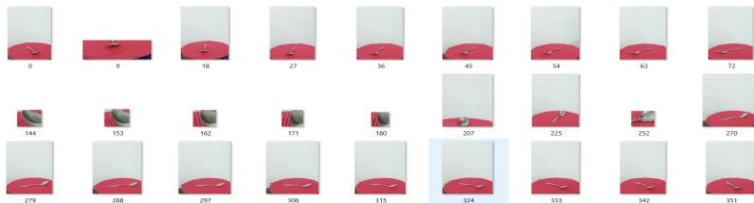
Dataset ini juga mengandung informasi visual warna yang telah dihilangkan latar belakangnya dengan metoda yang diajukan pada penelitian ini. Hasil tersebut juga diberlakukan *resize* beukuran (225,225) dalam satuan pixel. Hal ini dilakukan agar dataset menjadi standar dan akan baik digunakan karena ukuran tidak akan dipengaruhi oleh jarak obyek terhadap sensor pengambil gambar tersebut. Pada gambar 4.15 dataset informasi visual warna dari sampel obyek, yaitu lifebuoy.



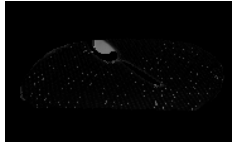
Gambar 4.15 Dataset informasi visual warna obyek lifebuoy

Lalu, terdapat juga data *pointcloud* yang disimpan dalam bentuk (.txt) yang berisikan dari koordinat dunia nyata dari masing – masing obyek yang telah dipelajari.

Pengambilan data ini tidak selalu berjalan dengan baik, contohnya saja pada dataset sendok yang terlihat pada gambar 4.16.



Gambar 4 16 Dataset informasi visual warna obyek sendok



Gambar 4.17 Visualisasi matriks probabilitas obyek

Peneliti melakukan analisis bahwa hal tersebut terjadi karena obyek sendok terlalu tipis dan memiliki luasan yang kecil. Seperti yang terlihat pada gambar 4.17, visualisasi matriks probabilitas obyek, luasan tersebut sangat kecil dan akan hilang jika diberlakukan morfologi filter.

Dari berbagai pengujian yang telah dilakukan, peneliti dapat menyimpulkan spesifikasi dari sistem yang akan dipaparkan pada tabel 4.8.

Tabel 4.8 Tabel spesifikasi alat pada sistem ini.

No	Spesifikasi	Keterangan
1	Supply Listrik Meja Belajar	220V (PLN)
2	Supply Listrik Kinect V2	220V (PLN)
3	Min. PC	Terdapat USB 3.0
4	Smartphone	Min. Android Nougat
5	Jarak Kinect-Obyek terdekat	>50cm
6	Jarak Kinect-Obyek terjauh	<250 cm
7	Ukuran obyek terkecil	Luas: 3cm x 3cm x 2cm
8	Ukuran obyek terbesar	28cm x 28cm x 30cm
9	Ukuran obyek terberat	< 800 Gram
10	Durasi pembelajaran	± 2 menit/obyek
11	Suasana Lingkungan	Dalam ruangan, tidak ada gangguan warna merah yang berukuran besar (>= ukuran baju)

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pengujian dan analisis yang telah dibahas pada bab IV, dapat disimpulkan hasil penelitian ini adalah sebagai berikut:

1. Metode *GMM + Depth + HSV* merupakan metode terbaik dalam melakukan ekstraksi obyek, dibandingkan dengan metode lainnya yang dibandingkan pada penelitian ini, dengan nilai *precision* sebesar 83.18 ± 23.85 %
2. Sistem ini dapat melakukan pengukuran ukuran obyek dengan toleransi kesalahan sebesar kurang dari 1 cm untuk setiap sisinya (Panjang, Lebar, Tinggi)
3. Sistem ini dapat baik melakukan ekstraksi obyek pada berbagai kondisi di lingkungan sekitar perumahan tinggal
4. Sistem ini dapat melakukan pembelajaran obyek dengan spesifikasi ukuran tidak lebih kecil dari 3cm x 3cm x 2cm, tidak lebih besar dari 28cm x 28cm x 30cm dan tidak lebih berat dari 800 gram.

5.2 Saran

Berdasarkan penelitian yang telah dikerjakan, terdapat beberapa kekurangan yang dapat diperbaiki dan ditingkatkan kembali serta beberapa saran tambahan untuk penelitian selanjutnya agar sistem untuk robot DSR dapat terealisasi dengan baik.

1. Pada pemetaan kembali informasi visual RGB ke resolusi HD, menghasilkan pergeseran yang telah dijelaskan pada analisa bab IV. Permasalahan ini dapat diperbaiki dengan menggunakan dua sensor Kinect V2 (*stereo depth*) atau sensor kamera yang dapat menghasilkan kalibrasi antara informasi visual RGB dan *Depth* dengan baik.
2. Pembuatan dataset dapat dikembangkan dengan melakukan pemrosesan data visual mentah dari *Depth* dan NIR, sehingga didapatkan informasi lainnya, seperti bentuk dan bahan dari obyek tersebut.

3. Pada penelitian selanjutnya, database diharapkan dapat menjadi data yang baik untuk membuat model *object recognition*, *object detection*, dan *object pose estimation*.

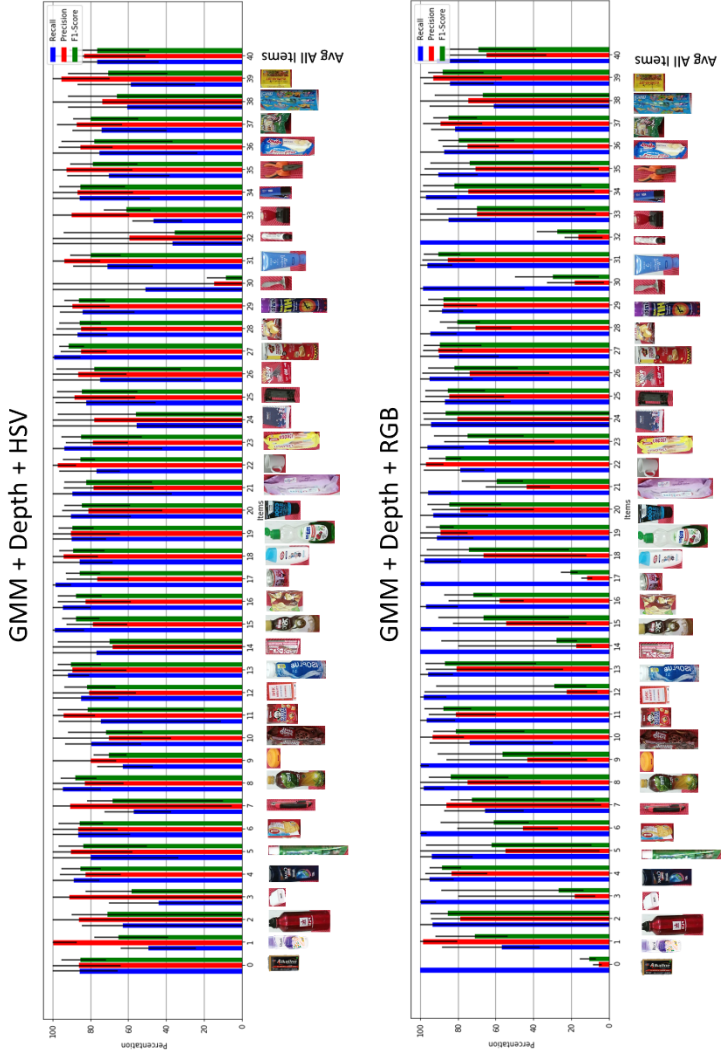
DAFTAR PUSTAKA

- [1] L. Iocchi, J. Ruiz-del-Solar, and T. van der Zant, “Domestic Service Robots in the Real World,” *J. Intell. Robot. Syst.*, vol. 66, no. 1–2, pp. 183–186, Apr. 2012.
- [2] M. Attamimi, T. Araki, T. Nakamura, and T. Nagai, “Visual Recognition System for Cleaning Tasks by Humanoid Robots,” *Int. J. Adv. Robot. Syst.*, vol. 10, no. 11, p. 384, Nov. 2013.
- [3] R. Muralidharan and D. C. Chandrasekar, “Object Recognition using SVM-KNN based on Geometric Moment Invariant,” *Int. J. Comput. Trends Technol.*, no. 2011, p. 7, 2011.
- [4] “Skyl.ai Data Labeling.” [Online]. Available: <https://skyl.ai/data-labeling>. [Accessed: 20-Nov-2019].
- [5] “WorkAround - Training data made easy.” [Online]. Available: <https://wao.ai/>. [Accessed: 20-Nov-2019].
- [6] “Home - Data Labeling and Document Management Company in India.” [Online]. Available: <https://www.precisebposolution.com/>. [Accessed: 20-Nov-2019].
- [7] K. Lai, L. Bo, X. Ren, and D. Fox, “A Large-Scale Hierarchical Multi-View RGB-D Object Dataset,” p. 8.
- [8] “Arduino Nano.” [Online]. Available: <https://store.arduino.cc/usa/arduino-nano>. [Accessed: 21-Nov-2019].
- [9] “hsv8.jpg (640×600).” [Online]. Available: <http://learn.leighcotnoir.com/wp-content/uploads/2011/07/hsv8.jpg>. [Accessed: 21-Nov-2019].
- [10] “Beginner’s guide to depth (Updated) – Intel® RealSense™ Depth and Tracking Cameras.” [Online]. Available: <https://www.intelrealsense.com/beginners-guide-to-depth/>. [Accessed: 21-Nov-2019].
- [11] S. S. Instruments, “UV-Vis Frequently Asked Questions - Light and Theory,” 16-Sep-2019. [Online]. Available: <https://www.ssi.shimadzu.com/>. [Accessed: 21-Nov-2019].
- [12] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011, pp. 1–4.

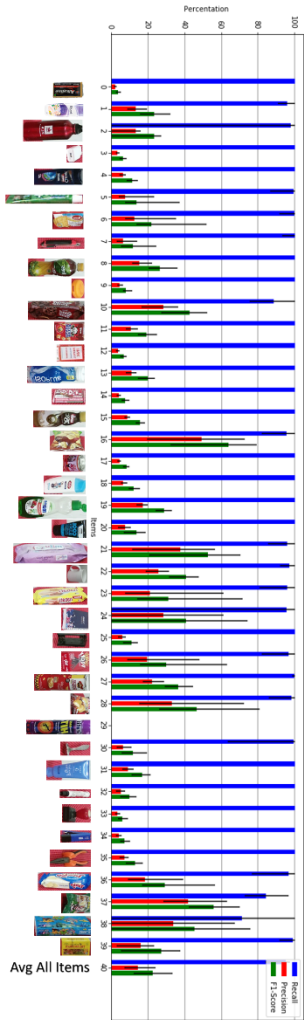
- [13] “Kinect SDK C++ Tutorials - 3. Point Clouds.” [Online]. Available: <https://homes.cs.washington.edu/~edzhang/tutorials/kinect2/kinect3.html>. [Accessed: 20-Nov-2019].
- [14] “About.” [Online]. Available: <https://opencv.org/about/>. [Accessed: 21-Nov-2019].
- [15] “Logo Ionic Scalable Graphics Computer Icons TypeScript, hypochlorous acid products transparent background PNG clipart | HiClipart.” [Online]. Available: <https://www.hiclipart.com/free-transparent-background-png-clipart-lnqin>. [Accessed: 08-Dec-2019].
- [16] Drifty, “Welcome to Ionic - Ionic Framework.” [Online]. Available: <https://ionicframework.com/docs/v1/guide/preface.html>. [Accessed: 08-Dec-2019].
- [17] “Dokumentasi,” *Firebase*. [Online]. Available: <https://firebase.google.com/docs>. [Accessed: 21-Nov-2019].

LAMPIRAN

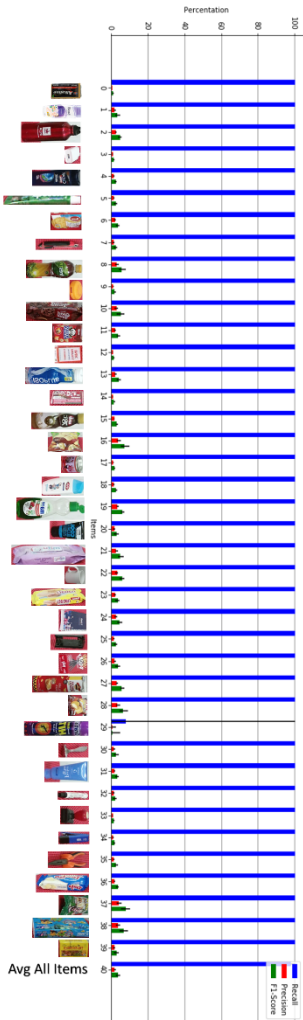
A. Evaluasi Pembelajaran Obyek



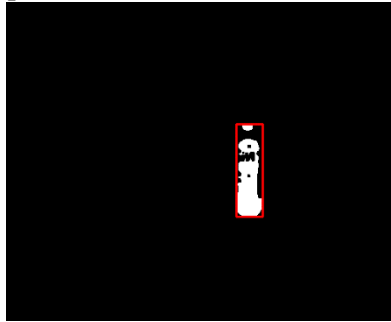
DEPTH ONLY



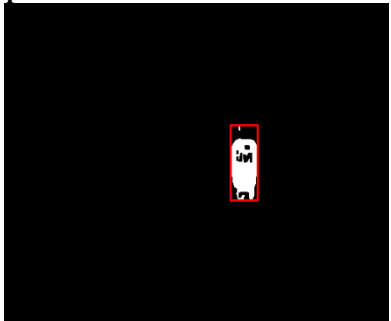
GMM + RGB



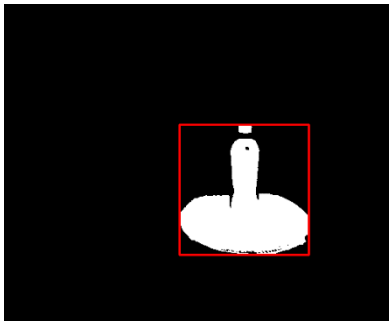
Contoh Hasil Ekstraksi Obyek
GMM + Depth + HSV



GMM + Depth + RGB



Depth Only



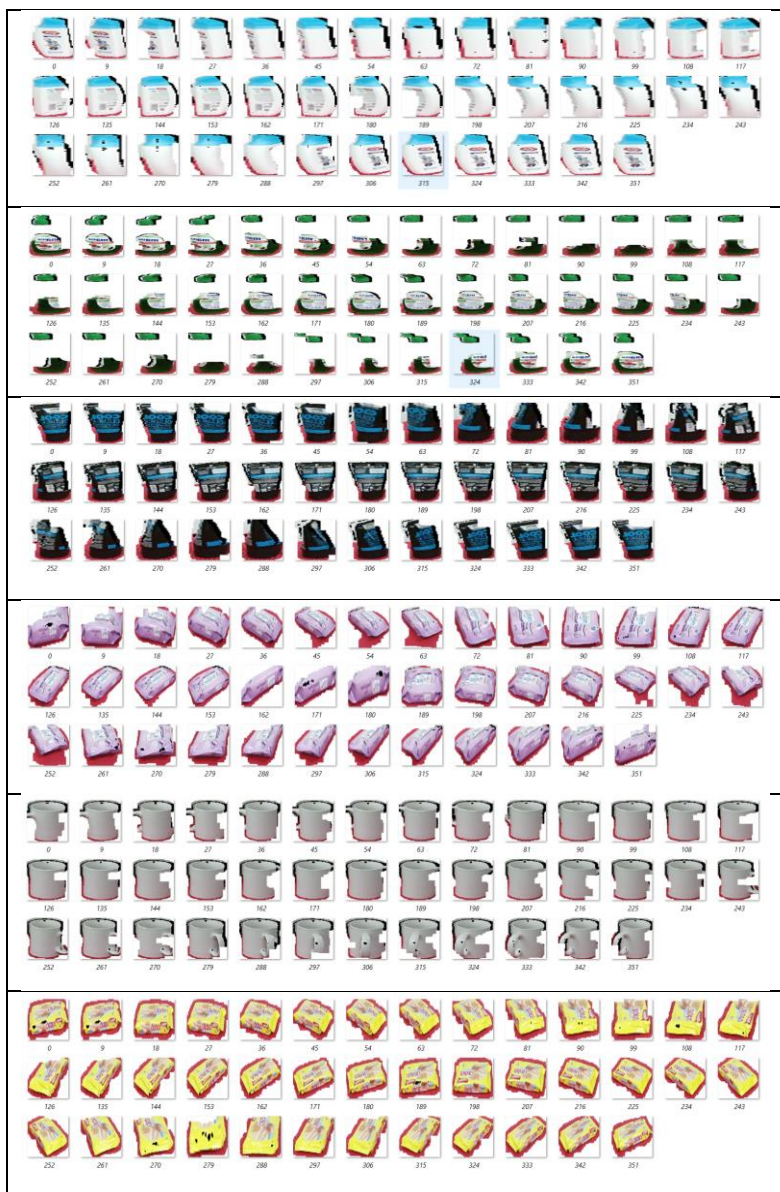
GMM + Color (RGB)



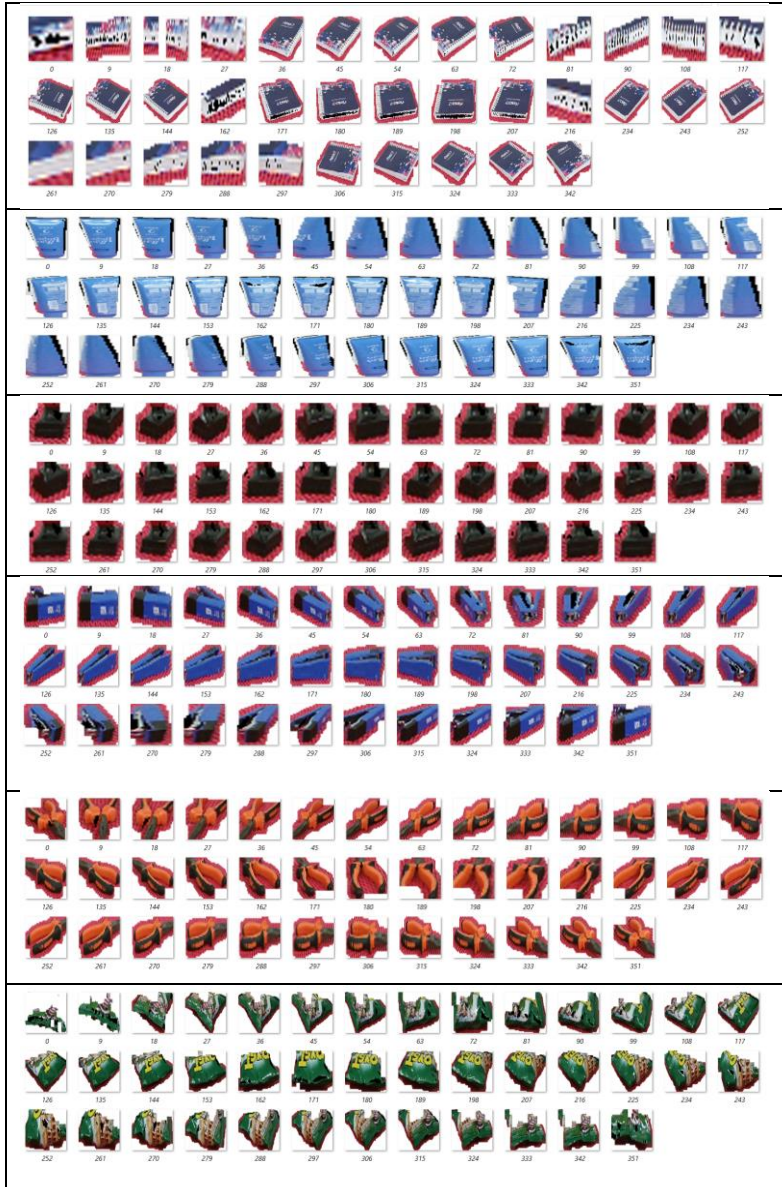
B. Dataset





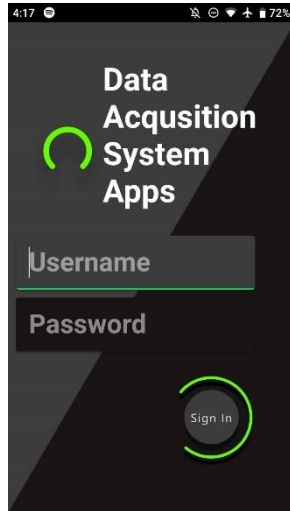






-----*Halaman ini sengaja dikosongkan*-----

C. Tampilan Antarmuka Pengguna Halaman Masuk



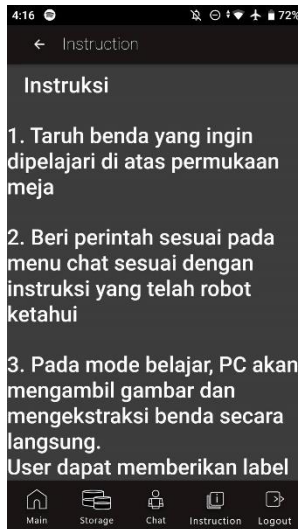
Halaman Menu Utama



Halaman Percakapan

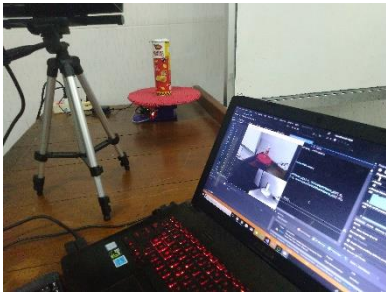


Halaman Instruksi



D. Dokumentasi Eksperimen

Eksperimen Setup



Lingkungan Dinamis sekitar Rumah Tangga.







Level Pencahayaan Berbeda



-----*Halaman ini sengaja dikosongkan*-----

E. Program

E.1 Program Utama (PC)

Program Utama Kinect

```
#include <iostream>
#include "KinectV2.h"
#include "pch.h"
#include "ArduSerial.cpp"
#include "SoiModel.h"
#include "ColorNirModel.h"

#pragma warning(disable:4996)

constexpr auto NUM_OF_FRAMES = 40;

int cntFrame = 0;
int cntObject = 0;
int mode = 0;
bool isNewFolder = true;
bool flag_captSoi;
bool flag_captObj;
bool flag_recog;
bool flag_detectCenter = false;
float tilt = -7;

char foldernam[512], foldernamSOI[512], pathnam[512],
word[512], foldernamResult[512];

std::string datas;
std::string itemLabel;

void setupBlueSerial11()
{
    Serial11.begin(9600);
    // sebelumnya 115200
    std::cout << "Starting..." << std::endl;
    while (!Serial11);
    while (Serial11.available())
```

```

        Serial11.read();
        std::cout << "Connected to Serial11" << std::endl;
        std::string data = "test123";
        std::cout << "Sending data: " << data << std::endl;
        Serial11.println(data);
    }

void setupBlueSerial6()
{
    Serial6.begin(9600);
        // sebelumnya 115200
    std::cout << "Starting..." << std::endl;
    while (!Serial6);
    while (Serial6.available())
        Serial6.read();
    std::cout << "Connected to Serial6" << std::endl;

    std::string data = "test123";
}

void setupBlueSerial1()
{
    Serial1.begin(9600);
        // sebelumnya 115200
    std::cout << "Starting..." << std::endl;
    while (Serial1.available())
        Serial1.read();
    std::cout << "Connected to Serial1" << std::endl;
}

std::string receiveDataSerial1()
{
    datas = "";
    while (Serial1.available())
    {
        buffer:
            char data_in = Serial1.read();
    }
}

```

```

        datas += data_in;
        Sleep(10);
        if (Serial1.available()) { goto buffer; }
    }
    datas = datas.substr(0, datas.length() - 2); // 2 data terakhir
dibuang, karena ada kelebihan 2 data yg tidak diketahui( asumsi =spasi
+ /n)
    return datas;
}

void detectBlue(cv::Mat halfSizedColorMat, cv::Mat& frame,
cv::Point& center2) {
    std::vector<std::vector<cv::Point> > contours;
    std::vector<cv::Vec4i> hierarchy;
    double area, largest = 0;
    int idxContour = 0;
    float radius;
    cv::Rect rect;

    cv::Mat maskHSV, maskHSV1, maskHSV2, resultHSV;
    cv::Point2f center;

    frame = halfSizedColorMat.clone();

    //thresholding
    cvtColor(frame, maskHSV, cv::COLOR_BGR2HSV);
    inRange(maskHSV, cv::Scalar(0, 100, 120), cv::Scalar(10,
255, 255), maskHSV1);
    inRange(maskHSV, cv::Scalar(160, 100, 120),
cv::Scalar(179, 255, 255), maskHSV2);
    cv::addWeighted(maskHSV1, 1.0, maskHSV2, 1.0, 0.0,
maskHSV);
    bitwise_and(frame, frame, resultHSV, maskHSV);

    //find contours
    findContours(maskHSV, contours, hierarchy,
cv::RETR_CCOMP, cv::CHAIN_APPROX_SIMPLE);
    for (int i = 0; i < contours.size(); i++)

```

```

    {
        area = contourArea(contours[i]);
        if (area > largest)
        {
            largest = area;
            idxContour = i;
        }
    }

    // jika terdeteksi contour
    if (idxContour > 0)
    {
        rect = boundingRect(contours[idxContour]);
        minEnclosingCircle(contours[idxContour], center,
radius);
        cv::Moments m =
cv::moments(contours[idxContour], true);
        cv::Point p(m.m10 / m.m00, m.m01 / m.m00);
        center2 = p;
        if (radius > 10) {
            cv::rectangle(frame, rect, cv::Scalar(0, 0,
255), 2);
            cv::circle(frame, center2, 5, (255, 0, 0), -1);
        }
        //rectangle(frame, center + cv::Point2f(-10,10), center +
cv::Point2f(-10, 10), cv::Scalar(0,255,0), 2);
        //std::cout << center2; // HASIL AKHIR CENTER
    }

    td::string receiveDataSerial11()
    {
        datas = "";
        while (Serial11.available())
        {
            buffer:
                char data_in = Serial11.read();
                datas += data_in;
        }
    }

```



```

        //Sleep(10);
        if (Serial11.available()) { goto buffer; }
    }
    datas = datas.substr(0, datas.length() - 2); // 2 data terakhir
    dibuang, karena ada kelebihan 2 data yg tidak diketahui( asumsi =spasi
    + /n)
    return datas;
}

std::string receiveDataSerial6()
{
    datas = "";
    while (Serial6.available())
    {
        buffer:
        char data_in = Serial6.read();
        datas += data_in;
        //Sleep(10);
        if (Serial6.available()) { goto buffer; }
    }
    datas = datas.substr(0, datas.length() - 2); // 2 data terakhir
    dibuang, karena ada kelebihan 2 data yg tidak diketahui( asumsi =spasi
    + /n)
    return datas;
}

void objectExtraction(char* foldername, std::string itemLabel, int
objView, float tilt)
{
    // Load color image in Full-HD resolution
    char fn[1024];

    sprintf(fn, "%s/%03d.color.png", foldername, objView);
    cv::Mat collmg;
    createColorImg(fn, collmg);

    //K: LOAD NIR
    sprintf(fn, "%s/%03d.nir.png", foldername, objView);

```

```

        cv::Mat          nirImg          =          cv::imread(fn,
cv::IMREAD_UNCHANGED);

        //K: LOAD DEPTH
        sprintf(fn, "%s/%03d.depth.png", foldername, objView);
        cv::Mat          depthImg          =          cv::imread(fn,
cv::IMREAD_UNCHANGED);

        // Load color calib image
        sprintf(fn, "%s/%03d.calib.png", foldername, objView);
        cv::Mat          calibColImg          =          cv::imread(fn,
cv::IMREAD_UNCHANGED);

        // In: CamPosX, Y, Z, Pan, Tilt; Out: TF
        Eigen::Matrix4f tf_all;
        createTransformMatrix(0, 0, 0, 0, tilt, tf_all, false);

        std::vector<PixelInfo> pixInfo;
        createPixelsInfo(tf_all, pixInfo, foldername, objView);

        // Load Object ProbMap
        cv::Mat probMaps_obj[3], bin_obj[3];

        for (int i = 0; i < 2; i++)
        {
            sprintf(fn,
"Results/%s/%03d/%03d.probMaps_obj.png", itemLabel, objView, i);
            probMaps_obj[i]          =          cv::imread(fn,
cv::IMREAD_UNCHANGED);

            if (probMaps_obj[i].empty())
            {
                printf("Error in objectExtraction(), cannot
load %s \n", fn);
                return;
            }
        }
        //===== color

```

```

        sprintf(fn,
"Results/%s/%03d/%03d.probMaps_colnir_obj.png",    itemLabel,
objView, 1);
        probMaps_obj[2] = cv::imread(fn,
cv::IMREAD_UNCHANGED);

        cv::Mat colRes = cv::Mat::zeros(cv::Size(512, 424),
CV_8UC3);
        cv::Mat colImgRes, calibColImgRes, croppedImg,
croppedCalibImg, croppedNir, croppedDepth;

        //for (int i = 0; i < 2; i++)
        for (int i = 0; i < 2; i++)
        {
            // Binarization
            bin_obj[i] = cv::Mat::zeros(cv::Size(512, 424),
CV_8UC1);

            // Otsu's thresholding after Gaussian filtering
            cv::GaussianBlur(probMaps_obj[i],
probMaps_obj[i], cv::Size(5, 5), 0);
            cv::threshold(probMaps_obj[i], bin_obj[i], 100,
255, cv::THRESH_BINARY + cv::THRESH_OTSU);

            // Erosion and Dilation
            //cv::morphologyEx(bin_obj[i], bin_obj[i],
cv::MORPH_CLOSE, cv::Mat(), cv::Point(-1, -1), 5);
            //cv::erode(bin_obj[i], bin_obj[i], cv::Mat(),
cv::Point(-1, -1), 1, 1, 1);
            cv::dilate(bin_obj[i], bin_obj[i], cv::Mat(),
cv::Point(-1, -1), 1, 1, 1);
            cv::erode(bin_obj[i], bin_obj[i], cv::Mat(),
cv::Point(-1, -1), 3, 1, 1);
            cv::erode(bin_obj[i], bin_obj[i], cv::Mat(),
cv::Point(-1, -1), 1, 1, 1);
            cv::dilate(bin_obj[i], bin_obj[i], cv::Mat(),
cv::Point(-1, -1), 4, 1, 1);
            cv::dilate(bin_obj[i], bin_obj[i], cv::Mat(),
cv::Point(-1, -1), 2, 1, 1);

```

```

        // Find rectangle
        auto r = cv::boundingRect(bin_obj[i]);
        cv::cvtColor(bin_obj[i], colRes,
cv::COLOR_GRAY2BGR);
        cv::rectangle(colRes, r, CV_RGB(255, 0, 0), 2);

        sprintf(fn, "Results/%s/%03d/%03d.bin_obj.png",
itemLabel, objView, i);
        cv::imwrite(fn, bin_obj[i]);

        sprintf(fn, "Results/%s/%03d/%03d.colRes.png",
itemLabel, objView, i);
        cv::imwrite(fn, colRes);

        // TO DO: Find corresponding points in color image
(Full-HD)
        cv::Point lt_pt, rb_pt;
        /*
        if (!findCorrespondingPoint(pixInfo, cv::Point(r.x,
r.y), lt_pt, 1))
            printf("No corresponding point of Left Top
-> v:%03d!\n", objView);
        if (!findCorrespondingPoint(pixInfo, cv::Point(r.x +
r.width, r.y + r.height), rb_pt, 2))
            printf("No corresponding point of Right
Bottom -> v:%03d!\n", objView);
        */

        sprintf(fn, "Results/%s/%03d/%03d.rect.txt",
itemLabel, objView, i);
        FILE* fp = fopen(fn, "w");
        fprintf(fp, "%d\t%d\t%d\t%d\n", r.x, r.y, r.x +
r.width, r.y + r.height);
        fprintf(fp, "%d\t%d\t%d\t%d\n", lt_pt.x, lt_pt.y,
rb_pt.x, rb_pt.y);
        fclose(fp);

        calibColImgRes = calibColImg.clone();

```

```

cv::rectangle(calibColImgRes, r, CV_RGB(255, 0,
0), 2);
    sprintf(fn,
"Results/%s/%03d/%03d.calibColImgRes.png", itemLabel, objView,
i);
    cv::imwrite(fn, calibColImgRes);

    croppedCalibImg = calibColImg.clone();
    croppedCalibImg = croppedCalibImg(r);
    sprintf(fn,
"Results/%s/%03d/%03d.croppedCalibImg.png", itemLabel,
objView, i);
    cv::imwrite(fn, croppedCalibImg);

//=====
=====
=====

    croppedNir = nirImg.clone();
    croppedNir = croppedNir(r);
    sprintf(fn,
"Results/%s/%03d/%03d.croppedNir.png", itemLabel, objView, i);
    cv::imwrite(fn, croppedNir);

    croppedDepth = depthImg.clone();
    croppedDepth = croppedDepth(r);
    sprintf(fn,
"Results/%s/%03d/%03d.croppedDepth.png", itemLabel, objView, i);
    cv::imwrite(fn, croppedDepth);

//=====
=====
=====

//-----
----- FOR HD
    while (!findCorrespondingPoint(pixInfo,
cv::Point(r.x + r.width, r.y + r.height), rb_pt, 2)) {

```

```

        printf("No corresponding point of Right
Bottom -> v:%03d!\n", objView);
        r.width++;
        r.height++;
    }

    while (!findCorrespondingPoint(pixInfo,
cv::Point(r.x, r.y), lt_pt, 1) ) {
        printf("No corresponding point of Left Top
-> v:%03d!\n", objView);
        r.x++;
        r.y++;
    }

//-----ok
collImgRes = collImg.clone();
cv::rectangle(collImgRes, lt_pt, rb_pt,
CV_RGB(255, 0, 4);
sprintf(fn,
"Results/%s/%03d/%03d.collImgRes.png", itemLabel, objView, i);
cv::imwrite(fn, collImgRes);

// K :result cropped color,calib, nir , depth
croppedImg = collImg.clone();
croppedImg = croppedImg(cv::Rect(lt_pt, rb_pt));
sprintf(fn,
"Results/%s/%03d/%03d.croppedImg.png", itemLabel, objView, i);
cv::imwrite(fn, croppedImg);

}
//=====
===== just color
int i = 2;
bin_obj[i] = cv::Mat::zeros(cv::Size(512, 424), CV_8UC1);

```

```

        // Otsu's thresholding after Gaussian filtering
        cv::GaussianBlur(probMaps_obj[i],      probMaps_obj[i],
cv::Size(5, 5), 0);
        cv::threshold(probMaps_obj[i],  bin_obj[i],  100,  255,
cv::THRESH_BINARY + cv::THRESH_OTSU);

        // Erosion and Dilation
        cv::erode(bin_obj[i], bin_obj[i], cv::Mat(), cv::Point(-1, -1),
1, 1, 1);
        cv::dilate(bin_obj[i], bin_obj[i], cv::Mat(), cv::Point(-1, -1),
1, 1, 1);
        cv::erode(bin_obj[i], bin_obj[i], cv::Mat(), cv::Point(-1, -1),
3, 1, 1);
        cv::erode(bin_obj[i], bin_obj[i], cv::Mat(), cv::Point(-1, -1),
1, 1, 1);
        cv::dilate(bin_obj[i], bin_obj[i], cv::Mat(), cv::Point(-1, -1),
4, 1, 1);
        cv::dilate(bin_obj[i], bin_obj[i], cv::Mat(), cv::Point(-1, -1),
2, 1, 1);

        // Find rectangle
        auto r = cv::boundingRect(bin_obj[i]);
        cv::cvtColor(bin_obj[i], colRes, cv::COLOR_GRAY2BGR);
        cv::rectangle(colRes, r, CV_RGB(255, 0, 0), 2);

        sprintf(fn,          "Results/%s/%03d/%03d.bin_obj.png",
itemLabel, objView, i);
        cv::imwrite(fn, bin_obj[i]);

        sprintf(fn, "Results/%s/%03d/%03d.colRes.png", itemLabel,
objView, i);
        cv::imwrite(fn, colRes);

        // TO DO: Find corresponding points in color image (Full-
HD)
        cv::Point lt_pt, rb_pt;

```

```

        sprintf(fn, "Results/%s/%03d/%03d.rect.txt", itemLabel,
objView, i);
        FILE* fp = fopen(fn, "w");
        fprintf(fp, "%d\t%d\t%d\t%d\n", r.x, r.y, r.x + r.width, r.y +
r.height);
        fprintf(fp, "%d\t%d\t%d\t%d\n", lt_pt.x, lt_pt.y, rb_pt.x,
rb_pt.y);
        fclose(fp);

        calibColImgRes = calibColImg.clone();
        cv::rectangle(calibColImgRes, r, CV_RGB(255, 0, 0), 2);
        sprintf(fn, "Results/%s/%03d/%03d.calibColImgRes.png",
itemLabel, objView, i);
        cv::imwrite(fn, calibColImgRes);

        croppedCalibImg = calibColImg.clone();
        croppedCalibImg = croppedCalibImg(r);
        sprintf(fn, "Results/%s/%03d/%03d.croppedCalibImg.png",
itemLabel, objView, i);
        cv::imwrite(fn, croppedCalibImg);
    }

void objectExtractionUsingSOI(char* foldername, std::string
itemLabel, int objView)
{
    char fn[1024];
    sprintf(fn, "Results/%s/%03d/000.probMaps_soi.png",
itemLabel, objView);
    cv::Mat bin_obj = cv::imread(fn,
cv::IMREAD_UNCHANGED);
    if (bin_obj.empty())
    {
        printf("Error: In objectExtractionUsingSOI(),
cannot open %s \n", fn);
        return;
    }

    sprintf(fn, "%s/%03d.calib.png", foldername, objView);

```



```

        cv::Mat      calibCollImgRes      =      cv::imread(fn,
cv::IMREAD_UNCHANGED);
        if (calibCollImgRes.empty())
        {
                printf("Error:  In  objectExtractionUsingSOI(),
cannot open %s \n", fn);
                return;
        }

        cv::Mat  colRes  =  cv::Mat::zeros(cv::Size(512, 424),
CV_8UC3);

        // Find rectangle
        auto r = cv::boundingRect(bin_obj);
        cv::cvtColor(bin_obj, colRes, cv::COLOR_GRAY2BGR);
        cv::rectangle(colRes, r, CV_RGB(255, 0, 0), 2);

        sprintf(fn,          "Results/%s/%03d/%03d.bin_obj.png",
itemLabel, objView, 3);
        cv::imwrite(fn, bin_obj);

        sprintf(fn, "Results/%s/%03d/%03d.colRes.png", itemLabel,
objView, 3);
        cv::imwrite(fn, colRes);

        cv::rectangle(calibCollImgRes, r, CV_RGB(255, 0, 0), 2);
        sprintf(fn,  "Results/%s/%03d/%03d.calibCollImgRes.png",
itemLabel, objView, 3);
        cv::imwrite(fn, calibCollImgRes);

        sprintf(fn,  "Results/%s/%03d/%03d.rect.txt",  itemLabel,
objView, 3);
        FILE* fp = fopen(fn, "w");
        fprintf(fp, "%d\t%d\t%d\t%d\n", r.x, r.y, r.x + r.width, r.y +
r.height);
        fclose(fp);
}

```

```

void fObjExt(char foldername[512], std::string itemLabel, int v,
cv::Point center, float tilt) {
    calcProbMaps(foldername, itemLabel, v, center, tilt);

    objectExtraction(foldername, itemLabel, v, tilt);

    objectExtractionUsingSOI(foldername, itemLabel, v);
}

int main()
{
    setupBlueSerial6();
    setupBlueSerial1();

    CKinectV2 kv2;

    cv::Mat halfSizedColorMat = cv::Mat::zeros(cv::Size(1920 /
2, 1080 / 2), CV_8UC3);
    cv::Mat visAcqDataMat = cv::Mat::zeros(cv::Size(512 * 2,
1080 / 2 + 424), CV_8UC3);
    cv::Mat halfVisAcqDataMat =
cv::Mat::zeros(cv::Size(visAcqDataMat.cols *
0.5,
visAcqDataMat.rows * 0.5), CV_8UC3);

    cv::Mat calibColorMat;
    CameraSpacePoint* XYZInDepthSpace_p = nullptr;
    CameraSpacePoint* XYZInColorSpace_p = nullptr;
    ColorSpacePoint* colorSpacePointInDepthSpace_p =
nullptr;

    CreateDirectoryA("photo", NULL);
    CreateDirectoryA("photo/ObjectData", NULL);
    CreateDirectoryA("photo/SOI", NULL);
    CreateDirectoryA("photo/Results", NULL);

    while (true)
    {

```

```

cv::Point center1, center; // berubah" tiap frame

kv2.grabAll();

cv::Mat rawColor = kv2.getRawColorMat().clone();
cv::Mat          rawDepth          =
kv2.getRawDepthMat().clone();
cv::Mat rawNir = kv2.getRawInfraredMat().clone();

cv::Mat color = kv2.getColorMat().clone();
cv::Mat depth = kv2.getDepthColorMat().clone();
cv::Mat nir = kv2.getInfraredColorMat().clone();

cv::resize(color, halfSizedColorMat, cv::Size(), 0.5,
0.5);

//deteksi meja biru
detectBlue(halfSizedColorMat, halfSizedColorMat,
center1);

//imshow biasa

halfSizedColorMat.copyTo(visAcqDataMat(cv::Rect(0, 0,
halfSizedColorMat.cols, halfSizedColorMat.rows)));
depth.copyTo(visAcqDataMat(cv::Rect(0,
halfSizedColorMat.rows, depth.cols, depth.rows)));
nir.copyTo(visAcqDataMat(cv::Rect(depth.cols,
halfSizedColorMat.rows, nir.cols, nir.rows)));
cv::resize(visAcqDataMat, halfVisAcqDataMat,
cv::Size(), 0.5, 0.5);

imshow("visAcqDataMat", halfVisAcqDataMat);
char key = cv::waitKey(10);

switch (mode)
{
case 0:
if (Serial1.available())
{
datas = receiveDataSerial1();
}
}
}

```

```

std::cout << "receiveData:" <<
datas << std::endl;

if (datas == "belajar")
{
    datas = "";
    mode = 1;
}
else if (datas == "lihat meja")
{
    datas = "";
    mode = 4;
}
else if (datas == "tebak")
{
    datas = "";
    mode = 5;
}
else mode = 44;
}
break;

case 1:        //mode belajar
mode = 2;
std::cout << "Proses Belajar" << std::endl;
Serial1.println("Obyek apa ini?");
break;

case 2:        // proses labeling
if (Serial1.available()) {
    itemLabel = receiveDataSerial1();
    std::cout << "label:" << itemLabel
<< std::endl;

    // buat new folder
    mode = 3;
}
break;

```

```

        case 3:          // proses belajar
            flag_captObj = true;
            sprintf(word, "Tunggu sebentar ya. Master
akan mempelajari '%s' dahulu!", itemLabel);
            Serial1.println(word);
            mode = 0;
            break;

        case 4:          // lihat meja
            mode = 0;
            flag_captSoi = true;
            break;

        case 5:          // pengenalan objek
            mode = 0;
            Serial1.println("Tunggu sebentar. Aku
coba mengingatnya");
            flag_recog = true;
            itemLabel = "objectRecognize";
            break;

        case 44:
            Serial1.println("Maaf. Master belum
mengerti apa maksud anda.");
            mode = 0;
            break;

        default:
            break;
    }

    //CAPTURE SOI
    if (flag_captSoi) {
        flag_captSoi = false;
        std::cout << "CAPTURE SOI..." <<
std::endl;

        //1. BUAT FILE BARU (SOI)
        sprintf(foldernameSOI, "photo/SOI");

```

```

NULL);
CreateDirectoryA(foldernameSOI,
(COLOR,DEPTH,NIR) //2. CAPTURE SOI
foldernameSOI);
sprintf(pathname, "%s/000.color.png",
cv::imwrite(pathname, color);
foldernameSOI);
sprintf(pathname, "%s/000.depth.png",
cv::imwrite(pathname, rawDepth);
foldernameSOI);
sprintf(pathname, "%s/000.nir.png",
cv::imwrite(pathname, rawNir);

//3. calib.png + calibinfo.txt
calibColorMat, XYZInDepthSpace_p, color,
colorSpacePointInDepthSpace_p);
detectBlue(calibColorMat,
halfSizedColorMat, center);

//=====
===== MUST CHECK
//flag_detectCenter = true;

//=====
===== MUST CHECK
foldernameSOI);
sprintf(pathname, "%s/000.calib.png",
cv::imwrite(pathname, calibColorMat);

foldernameSOI);
sprintf(pathname, "%s/000.calibinfo.txt",
{
FILE* fp = fopen(pathname, "w");
if (fp)
{

```

```

        int xyzSize =
calibColorMat.cols * calibColorMat.rows;
        for (int i = 0; i < xyzSize;
i++)
        {
            fprintf(fp,
"%f\t%f\t%f\t%f\t%f\t%f\n",
                XYZInDepthSpace_p[i].X, XYZInDepthSpace_p[i].Y,
XYZInDepthSpace_p[i].Z,
                colorSpacePointInDepthSpace_p[i].X,
colorSpacePointInDepthSpace_p[i].Y);
        }
        fclose(fp);
    }
}

// 3. INISIASI SOI / DETECT SOI

std::cout << "TRAIN COLOR NIR GMM"
<< std::endl;

// 4. TRAIN COLOR NIR GMM
trainColorNirGMMs(foldernameSOI, 0,
center, tilt); // EDIT FOLDERNAME HANYA SOI
}

//CAPTURE OBJECT
//key = cv::waitKey(10);
if (flag_captObj)
{
    if (isNewFolder)
    {
        // 1. folder baru

        sprintf(foldername,
"photo/ObjectData/%s", itemLabel);

```

```

NULL);
CreateDirectoryA(foldername,

isNewFolder = false;
Serial6.println("go");

}

if (cntFrame == NUM_OF_FRAMES)
{
    //cntObject++;
    flag_captObj = false;
    cntFrame = 0;
    isNewFolder = true;
    Serial1.println("D");

}
else
{
    std::cout << "CAPTURE
OBJECT..." << std::endl;
// 1. CAPTURE SOI
(COLOR,DEPTH,NIR)
    sprintf(pathname,
"%s/%03d.color.png", foldername, cntFrame);
    cv::imwrite(pathname, color);
    sprintf(pathname,
"%s/%03d.depth.png", foldername, cntFrame);
    cv::imwrite(pathname,
rawDepth);
    sprintf(pathname,
"%s/%03d.nir.png", foldername, cntFrame);
    cv::imwrite(pathname, rawNir);

// 2. CALIB.PNG +
CALIBINFO.TXT
    kv2.calcCorrPoints(rawDepth,
color, calibColorMat, XYZInDepthSpace_p, XYZInColorSpace_p,
colorSpacePointInDepthSpace_p);

```



```

        sprintf(pathname,
"%s/%03d.calib.png", foldername, cntFrame);
        cv::imwrite(pathname,
calibColorMat);

        detectBlue(calibColorMat,
halfSizedColorMat, center);

        sprintf(pathname,
"%s/%03d.calibinfo.txt", foldername, cntFrame);
        {
            FILE*      fp      =
fopen(pathname, "w");
            if (fp)
            {
                int xyzSize =
calibColorMat.cols * calibColorMat.rows;
                for (int i = 0; i <
xyzSize; i++)
                    {
                        fprintf(fp, "%f\t%f\t%f\t%f\t%f\n",
XYZInDepthSpace_p[i].X,      XYZInDepthSpace_p[i].Y,
XYZInDepthSpace_p[i].Z,
colorSpacePointInDepthSpace_p[i].X,
colorSpacePointInDepthSpace_p[i].Y);
                    }
                fclose(fp);
            }
        }

        // 3. obj extraction (cropping)
        fObjExt(foldername, itemLabel,
cntFrame, center, tilt);

        Serial6.println("go");
        cntFrame++;
    }

```

```

    }

    if (flag_recog) {
        flag_recog = false;

        std::cout << "CAPTURE OBJECT..." <<
std::endl;

        // 1. folder baru
        sprintf(foldername, "photo/%s",
itemLabel);

        CreateDirectoryA(foldername, NULL);

        //      1.      CAPTURE      SOI
(COLOR,DEPTH,NIR)
        sprintf(pathname, "%s/%03d.color.png",
foldername, cntFrame);
        cv::imwrite(pathname, color);
        sprintf(pathname, "%s/%03d.depth.png",
foldername, cntFrame);
        cv::imwrite(pathname, rawDepth);
        sprintf(pathname, "%s/%03d.nir.png",
foldername, cntFrame);
        cv::imwrite(pathname, rawNir);

        //      2.      CALIB.PNG      +
CALIBINFO.TXT
        kv2.calcCorrPoints(rawDepth, color,
calibColorMat, XYZInDepthSpace_p, XYZInColorSpace_p,
colorSpacePointInDepthSpace_p);
        sprintf(pathname, "%s/%03d.calib.png",
foldername, cntFrame);
        cv::imwrite(pathname, calibColorMat);
        detectBlue(calibColorMat,
halfSizedColorMat, center);
        sprintf(pathname,
"%s/%03d.calibinfo.txt", foldername, cntFrame);

```

```

        {
            FILE* fp = fopen(pathname, "w");
            if (fp)
            {
                int xyzSize =
calibColorMat.cols * calibColorMat.rows;
                for (int i = 0; i < xyzSize;
i++)
                    {
                        fprintf(fp,
"%f\t%f\t%f\t%f\t%f\n",
                XYZInDepthSpace_p[i].X, XYZInDepthSpace_p[i].Y,
XYZInDepthSpace_p[i].Z,
                colorSpacePointInDepthSpace_p[i].X,
colorSpacePointInDepthSpace_p[i].Y);
                    }
                fclose(fp);
            }
        }
        // 3. obj extraction (cropping)
        fObjExt(foldername, itemLabel, cntFrame,
center, tilt);

        Serial1.println("U");
    }
    //else if (key == 'q' || key == 'Q') break;
}
}

```

Class Create GMM Model

```
#pragma once

#include <Windows.h>
#include "Utils.h"
#include "armadillo"

////////////////////////////////////

// Modeling Color and NIR

void getPixInfoOfTableTop(char* foldername, int v, cv::Point center,
float tilt, std::vector<PixelInfo>& pixInfoOfTableTop, cv::Mat&
soiBinMat);
void trainColorNirGMMs(char* foldername, int v, cv::Point center,
float tilt);
void calcProbMaps(char* foldername, int v, cv::Point center, float
tilt);

////////////////////////////////////

void getPixInfoOfTableTop(char* foldername, int v, cv::Point center,
float tilt, std::vector<PixelInfo>& pixInfoOfTableTop, cv::Mat&
soiBinMat)
{
    char filename[1024];

    // In: CamPosX, Y, Z, Pan, Tilt; Out: TF
    Eigen::Matrix4f tf_all;
    createTransformMatrix(0, 0, 0.36, 0, tilt, tf_all, false);

    std::vector<PixelInfo> pixInfo;
    createPixelsInfo(tf_all, pixInfo, foldername, v);

    sprintf(filename, "%s/%03d.depth.png", foldername, v);
    cv::Mat rawDepthMat, colorDepthMat;
    createDepthImg(filename, rawDepthMat, colorDepthMat);
    extractSOIFromCamPos(pixInfo, rawDepthMat, center,
soiBinMat, pixInfoOfTableTop);
```

```

}

void trainColorNirGMMs(char* foldername, int v, cv::Point center,
float tilt)
{
    std::vector<PixelInfo> pixInfoOfTableTop;
    cv::Mat soiBinMat;
    getPixInfoOfTableTop(foldername, v, center, tilt,
pixInfoOfTableTop, soiBinMat);

    // Create training data
    arma::mat data1; // 1. Mix of scaled R, G, B, and norm NIR
-> Dim = 4
    arma::mat data2; // 2. Mix of scaled H, S, and norm NIR ->
Dim = 3

    // Fill data1 and data2
    int n = 0;
    double scale = 1.0 / 255.0;
    for (auto p : pixInfoOfTableTop)
    {
        if (p.bgr.val[0] == 0 && p.bgr.val[1] == 0 &&
p.bgr.val[2] == 0) continue;

        arma::vec4 v1;
        v1[0] = (double)p.bgr.val[0] * scale;
        v1[1] = (double)p.bgr.val[1] * scale;
        v1[2] = (double)p.bgr.val[2] * scale;
        v1[3] = p.norm_nir;
        data1.insert_cols(n, v1);

        arma::vec3 v2;
        v2[0] = (double)p.hsv.val[0] * scale;
        v2[1] = (double)p.hsv.val[1] * scale;
        v2[2] = p.norm_nir;
        data2.insert_cols(n, v2);

        n++;
    }
}

```

```

// Model the data as a diagonal GMM with 5 Gaussians
arma::gmm_diag model1, model2;
bool status = false;

// 1. Learning of data1

status = model1.learn(data1, 5, arma::maha_dist,
arma::random_subset, 25, 100, 1e-10, true);
if (status == false) { std::cout << "learning of model1 was
failed" << std::endl; }
model1.save("model1.gmm");

// 2. Learning of data2
status = model2.learn(data2, 5, arma::maha_dist,
arma::random_subset, 25, 100, 1e-10, true);
if (status == false) { std::cout << "learning of model2 was
failed" << std::endl; }
model2.save("model2.gmm");

//// For debugging
//FILE* fp1 = fopen("learn_model1_wo_black.txt", "w");
//FILE* fp2 = fopen("learn_model2_wo_black.txt", "w");

//for (int i = 0; i < data1.n_cols; i++)
//{
//    auto v1 = data1.col(i);
//    auto v2 = data2.col(i);

//    fprintf(fp1, "%lf\t%lf\t%lf\t%lf\t%lf\n", v1[0],
v1[1], v1[2], v1[3], model1.log_p(v1));
//    fprintf(fp2, "%lf\t%lf\t%lf\t%lf\n", v2[0], v2[1],
v2[2], model2.log_p(v2));
//}

//fclose(fp1);
//fclose(fp2);
}

```

```

void calcProbMaps(char* foldername, std::string itemLabel, int v,
cv::Point center, float tilt)
{
    char filename[1024];

    // In: CamPosX, Y, Z, Pan, Tilt; Out: TF
    Eigen::Matrix4f tf_all;
    createTransformMatrix(0, 0, 0, 0, tilt, tf_all, false);

    std::vector<PixelInfo> pixInfo;
    createPixelsInfo(tf_all, pixInfo, foldername, v);

    //additional
    //SavePixelInfo("pixelinfo12.txt", pixInfo);

    sprintf(filename, "%s/%03d.depth.png", foldername, v);
    cv::Mat rawDepthMat, colorDepthMat;
    createDepthImg(filename, rawDepthMat, colorDepthMat);

    std::vector<PixelInfo> pixInfoOfTableTop;
    cv::Mat soiBinMat;

    extractSOIFromCamPos(pixInfo, rawDepthMat, center,
soiBinMat, pixInfoOfTableTop);

    // Load gmms
    arma::gmm_diag model1, model2;
    bool status = false;

    status = model1.load("model1.gmm");
    if (status == false) { std::cout << "loading of model1 was
failed" << std::endl; }

    status = model2.load("model2.gmm");
    if (status == false) { std::cout << "loading of model2 was
failed" << std::endl; }
}

```

```

        cv::Mat probMapsRaw_soi[2], probMapsRaw_colnir[2],
probMapsRaw_colnir_norm[2], probMapsRaw_colnir_obj[2],
probMapsRaw_obj[2];
        cv::Mat probMaps_soi[2], probMaps_colnir[2],
probMaps_colnir_obj[2], probMaps_obj[2];
        for (int i = 0; i < 2; i++)
        {
                probMapsRaw_soi[i] =
cv::Mat::zeros(cv::Size(512, 424), CV_64FC1);
                // ProbMap of Space of Interest
                probMapsRaw_colnir[i] =
cv::Mat::zeros(cv::Size(512, 424), CV_64FC1);           //
                ProbMap of TableTop (Non-Object)
                probMapsRaw_colnir_norm[i] =
cv::Mat::zeros(cv::Size(512, 424), CV_64FC1);           // Normalized
                ProbMap of TableTop (Non-Object)
                probMapsRaw_colnir_obj[i] =
cv::Mat::zeros(cv::Size(512, 424), CV_64FC1);           // PropMap of
                Object
                probMapsRaw_obj[i] =
cv::Mat::zeros(cv::Size(512, 424), CV_64FC1);
                // Integrated ProbMap = P_soi x P_obj

                probMaps_soi[i] = cv::Mat::zeros(cv::Size(512,
424), CV_8UC1);
                probMaps_colnir[i] = cv::Mat::zeros(cv::Size(512,
424), CV_8UC1);
                probMaps_colnir_obj[i] =
cv::Mat::zeros(cv::Size(512, 424), CV_8UC1);
                probMaps_obj[i] = cv::Mat::zeros(cv::Size(512,
424), CV_8UC1);
        }

        double scale = 1.0 / 255.0;
        for (auto p : pixInfo)
        {
                if (soiBinMat.at<uchar>(p.depPt) == 255)
probMapsRaw_soi[0].at<double>(p.depPt) = 1.0;
                arma::vec4 v1;

```



```

        v1[0] = (double)p.bgr.val[0] * scale;
        v1[1] = (double)p.bgr.val[1] * scale;
        v1[2] = (double)p.bgr.val[2] * scale;
        v1[3] = p.norm_nir;
        auto likelihood1 = model1.log_p(v1);
        probMapsRaw_colnir[0].at<double>(p.depPt) =
likelihood1;

        probMapsRaw_soi[1].at<double>(p.depPt) =
probMapsRaw_soi[0].at<double>(p.depPt);
        arma::vec3 v2;
        v2[0] = (double)p.hsv.val[0] * scale;
        v2[1] = (double)p.hsv.val[1] * scale;
        v2[2] = p.norm_nir;
        auto likelihood2 = model2.log_p(v2);
        probMapsRaw_colnir[1].at<double>(p.depPt) =
likelihood2;
    }

    for (int i = 0; i < 2; i++)
    {
        double minVal, maxVal;
        cv::minMaxLoc(probMapsRaw_colnir[i],
&minVal, &maxVal);
        scale = 1.0 / (maxVal - minVal);

        //printf("model-%d -> min:%lf, max:%lf, scale:%lf
\n", i, minVal, maxVal, scale);
        for (int y = 0; y < probMapsRaw_colnir[i].rows;
y++)
        {
            for (int x = 0; x <
probMapsRaw_colnir[i].cols; x++)
            {
                int index = y * 512 + x;
                auto p = pixInfo[index];
                if (p.bgr.val[0] == 0 &&
p.bgr.val[1] == 0 && p.bgr.val[2] == 0)
                {

```

```

//probMapsRaw_soi[i].at<double>(y, x) = 0;

probMapsRaw_colnir_norm[i].at<double>(y, x) = 0;

probMapsRaw_colnir_obj[i].at<double>(y, x) = 0;

probMapsRaw_obj[i].at<double>(y, x) = 0;

                                continue;
                                }

    probMapsRaw_colnir_norm[i].at<double>(y, x) =
(probMapsRaw_colnir[i].at<double>(y, x) - minVal) * scale;

    probMapsRaw_colnir_obj[i].at<double>(y, x) = 1.0 -
probMapsRaw_colnir_norm[i].at<double>(y, x);

    probMapsRaw_obj[i].at<double>(y, x) =
probMapsRaw_soi[i].at<double>(y, x) *
probMapsRaw_colnir_obj[i].at<double>(y, x);

                                probMaps_soi[i].at<uchar>(y, x)
= static_cast<uchar>(255.0 * probMapsRaw_soi[i].at<double>(y, x));
                                // denorm saja
                                probMaps_colnir[i].at<uchar>(y,
x) = static_cast<uchar>(255.0 *
probMapsRaw_colnir_norm[i].at<double>(y, x));
                                // denorm saja

    probMaps_colnir_obj[i].at<uchar>(y, x) =
static_cast<uchar>(255.0 *
probMapsRaw_colnir_obj[i].at<double>(y, x));
                                //
denorm saja

                                probMaps_obj[i].at<uchar>(y, x)
= static_cast<uchar>(255.0 * probMapsRaw_obj[i].at<double>(y,
x));
                                // denorm saja
                                }

```

```

    }
}

// Results
CreateDirectoryA("Results", NULL);
sprintf(filename, "Results/%s", itemLabel);
CreateDirectoryA(filename, NULL);
sprintf(filename, "Results/%s/%03d", itemLabel, v);
CreateDirectoryA(filename, NULL);

for (int i = 0; i < 2; i++)
{
    sprintf(filename,
"Results/%s/%03d/%03d.probMaps_soi.png", itemLabel, v, i);
    cv::imwrite(filename, probMaps_soi[i]);

    sprintf(filename,
"Results/%s/%03d/%03d.probMaps_colnir.png", itemLabel, v, i);
    cv::imwrite(filename, probMaps_colnir[i]);

    sprintf(filename,
"Results/%s/%03d/%03d.probMaps_colnir_obj.png", itemLabel, v,
i);
    cv::imwrite(filename, probMaps_colnir_obj[i]);

    sprintf(filename,
"Results/%s/%03d/%03d.probMaps_obj.png", itemLabel, v, i);
    cv::imwrite(filename, probMaps_obj[i]);
}
}

```

Class SOI Init

```
#pragma once

#include "Utils.h"

void extractSOIFromCamPos(std::vector<PixelInfo>&
pixInfo, cv::Mat rawDepthMat, cv::Point center,
cv::Mat& soiBinMat, std::vector<PixelInfo>&
pixInfoOfTableTop);

void extractSOIFromCamPos(std::vector<PixelInfo>&
pixInfo, cv::Mat rawDepthMat, cv::Point center,
cv::Mat& soiBinMat, std::vector<PixelInfo>&
pixInfoOfTableTop)
{
    pixInfoOfTableTop.clear();
    // center = titik tengah nya meja

    soiBinMat = cv::Mat::zeros(rawDepthMat.size(),
CV_8UC1);
    cv::Point center_pt =
cv::Point(rawDepthMat.cols * 0.5, rawDepthMat.rows *
0.5);

    //auto centerTableY =
rawDepthMat.at<ushort>(center.y, center.x);
    // SOI = r/2
    float soiY = 0.15;
    float soiX = 0.15;
    float soiZ = 0.15;

    int iCenter = center.y * rawDepthMat.cols +
center.x;
    //std::cout << center.y << std::endl;
    //std::cout << center.x << std::endl;
    auto centerTableY =
pixInfo[iCenter].wldPos.val[0];
    auto centerTableX =
pixInfo[iCenter].wldPos.val[1];
```

```

        auto centerTableZ =
pixInfo[iCenter].wldPos.val[2];

        for (int y = 0; y < rawDepthMat.rows; y++)
        {
            for (int x = 0; x < rawDepthMat.cols;
x++)

                {
                    //auto v =
rawDepthMat.at<ushort>(y, x);
                    //int i = y * rawDepthMat.cols +
x;

                    int i = y * rawDepthMat.cols + x;
                    auto wy =
pixInfo[i].wldPos.val[0];
                    auto wx =
pixInfo[i].wldPos.val[1];
                    auto wz =
pixInfo[i].wldPos.val[2];

                    if (wy > centerTableY - soiY &&
wy < centerTableY + soiY &&
                    wx > centerTableX - soiX
&& wx < centerTableX + soiX &&
                    wz > centerTableZ - soiZ *
2 && wz < centerTableZ + 0.01) //0.1 = error
ketelitian tinggi
                    {

                        soiBinMat.at<uchar>(pixInfo[i].depPt) = 255;

                        pixInfoOfTableTop.push_back(pixInfo[i]);
                    }

                }
        }

```

-----*Halaman ini sengaja dikosongkan*-----

E.2 Program Arduino

Program Arduino Stepper Bluetooth

```
#include<SoftwareSerial.h>
#define stepPin 6
#define sleepPin 7
#define dirPin 5
#define rxPin 4
#define txPin 3
SoftwareSerial hc05(rxPin, txPin);
#define baudrate 9600

String receivedString;
String msg;

int i1 = 0;

void setup() {
    pinMode(stepPin, OUTPUT);
    pinMode(sleepPin, OUTPUT);
    pinMode(dirPin, OUTPUT);
    pinMode(rxPin, INPUT);
    pinMode(txPin, OUTPUT);
    hc05.begin(baudrate);
    Serial.begin(baudrate);
    Serial.println("READY ON");
}

void loop() {

    readSerialPort();
    if (msg != "") hc05.println(msg);
    if (hc05.available() > 0)
    {
        lcd.setCursor(0, 1);
        lcd.print("RECEIVED");
        receivedString = hc05.readStringUntil('\n');
        Serial.println(receivedString);
        mode_on();
        delay(10);
    }
}
```

```

    }
}

void mode_on() {
    int n_step = 5;
    digitalWrite(sleepPin, HIGH);
    delay(100);
    if (i1 * n_step * 1.8 > 359)
    {
        i1 = 0;
        digitalWrite(sleepPin, LOW);
    }
    oneMove(HIGH, n_step, i1);
    i1 = i1 + 1;
}

void send_go() {
    hc05.write('5');
    delay(1);
}

void readSerialPort()
{
    msg = "";
    while (Serial.available()) {
        delay(10);
        if (Serial.available() > 0) {
            char c = Serial.read();
            msg += c;
        }
    }
}

void oneMove(bool dir, int n_step, int i)
{
    // function : oneMove=1.8deg*step
    // dir: direction
    digitalWrite(dirPin, dir);
}

```



```
    for (int x = 0; x < n_step; x++) {
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(16000);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(16000);
    }
    delay(50);
}

void fullRotate(bool dir, int n_step, int delayMS)
{
    int n_loop = 200 / n_step;
    for (int i = 0; i < n_loop; i++) {
        float degree = n_step * (i + 1) * 1.8;
        digitalWrite(sleepPin, HIGH);
        digitalWrite(dirPin, dir);
        delay(100);
        for (int x = 0; x < n_step; x++) {
            digitalWrite(stepPin, HIGH);
            delayMicroseconds(16000);
            digitalWrite(stepPin, LOW);
            delayMicroseconds(16000);
        }
        delay(delayMS);
    }
}
```

-----*Halaman ini sengaja dikosongkan*-----

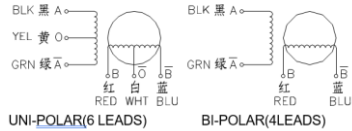
F. Datasheet Stepper Motor

2 Phase Hybrid Stepper Motor

17HK series-Size 42mm(0.9 degree)



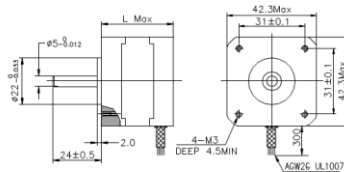
Wiring Diagram:



Series Model	Step Angle (deg)	Motor Length (mm)	Rated Current (A)	Phase Resistance (ohm)	Phase Inductance (mH)	Holding Torque (N.cm Min)	Detent Torque (N.cm Max)	Rotor Inertia (g.cm ²)	Lead Wire (No.)	Motor Weight (g)
17HK2408	0.9	28	0.6	9	9	10	0.9	20	4	130
17HK3401	0.9	34	1.3	2.4	3.6	24	1.2	34	4	220
17HK3410	0.9	34	1.7	1.2	2.2	24	1.2	34	4	220
17HK3430	0.9	34	0.4	30	45	24	1.2	34	4	220
17HK3630	0.9	34	0.4	30	22	20	1.2	34	6	220
17HK3616	0.9	34	0.16	75	54	12	1.2	34	6	220
17HK4401	0.9	40	1.7	1.5	3.8	36	1.8	54	4	280
17HK4402	0.9	40	1.3	2.5	6.2	36	1.8	54	4	280
17HK4602	0.9	40	1.2	3.2	3.8	26	1.8	54	6	280
17HK4630	0.9	40	0.4	30	34	26	1.8	54	6	280
17HK8401	0.9	48	1.7	1.8	4.0	42	2.2	68	4	350
17HK8402	0.9	48	1.3	3.0	7.0	36	2.2	68	4	350
17HK8403	0.9	48	2.3	1.2	2.0	42	2.2	68	4	350
17HK8630	0.9	48	0.4	30	28	27	2.2	68	6	350

****Note:** We can manufacture products according to customer's requirements.

Dimensions: unit=mm



Motor Length:

Model	Length
17HS2XXX	28 mm
17HS3XXX	34 mm
16HS4XXX	40 mm
16HS8XXX	48 mm

-----Halaman ini sengaja dikosongkan-----

BIODATA PENULIS



Kelvin Liusiani lahir di Bekasi pada tanggal 4 Oktober 1998. Penulis memulai kehidupan perkuliahan pada tahun 2016 di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya. Selama masa perkuliahan, penulis aktif dalam berbagai kegiatan kepanitiaan dan organisasi khususnya di bidang keagamaan. Selain itu, penulis juga aktif menjadi asisten praktikum di bidang studi elektronika. Penulis dapat dihubungi melalui e-mail: kelvinliu04@gmail.com