

**PERANCANGAN DAN PEMODELAN *AIRCRAFT VISUAL DOCKING*
GUIDANCE SYSTEM BERBASIS ARDUINO**

TUGAS AKHIR

diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

Oleh:

Nama: Octafian Firdaus

NPM: 157002001



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS SILIWANGI
TASIKMALAYA
OKTOBER, 2021**

HALAMAN JUDUL

**PERANCANGAN DAN PEMODELAN *AIRCRAFT VISUAL DOCKING*
GUIDANCE SYSTEM BERBASIS ARDUINO**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

Oleh:

Nama: Octafian Firdaus

NPM: 157002001



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS SILIWANGI
TASIKMALAYA
OKTOBER, 2021**

HALAMAN PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

Nama : Octafian Firdaus

NPM : 157002001

Fakultas : Teknik

Jurusan : Elektro

Bersama ini saya menyatakan dengan sebenar-benarnya, bahwa laporan skripsi ini merupakan hasil harya saya sendiri dan saya pribadi bertanggung jawab secara penuh terhadap hasil karya ini.

Tasikmalaya, 18 Oktober 2021

Yang menyatakan,

Matrei Rp.6000

Octafian Firdaus

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Octafian Firdaus

NIM : 157002001

Program Studi : Teknik Elektro

Judul Skripsi : Perancangan dan Pemodelan *Aircraft Visual Docking Guidance System* Berbasis Arduino

DEWAN PENGUJI

Pembimbing I : Asep Andang, S.T.,M.T (.....)

Pembimbing II : Nundang Busaeri, ir.,M.T. (.....)

Penguji I : Nurul Hiron, S.T.,M.Eng. (.....)

Penguji II : Firmansyah Maulana,S, S.T.,MT (.....)

Ditetapkan di : Tasikmalaya

Tanggal : 18 Oktober 2021

Mengetahui,

Dekan

Ketua Program Studi

Fakultas Teknik

Teknik Elektro

Prof. Aripin, Ph. D.

Nurul Hiron,S.T.,M. Eng.

NIP 196708161996031001

NIDN 419087504

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Siliwangi. Tak lupa saya ucapkan terimakasih kepada pihak-pihak yang atas kontribusi secara langsung maupun tidak langsung membantu kelancaran skripsi ini, diantaranya kepada:

1. Ayahanda Edy Sujono, Ibunda Ummy Khomisah, dan Kakak penulis Naufal Ariq Ramadhan yang selalu memberikan doa, dukungan dan dorongan semangat serta menjadi inspirasi dan motivasi bagi saya.
2. Prof. Arifin selaku dekan Fakultas Teknik atas segala dukungan, kebijakan dan motivasinya selama penulis menempuh pendidikan di Universitas Siliwangi.
3. BapK Nurul Hiron selaku Kepala Prodi Teknik Elektro atas segala dukungan, kebijakan dan motivasinya selama penulis menempuh pendidikan di Universitas Siliwangi.
4. Bapak Asep Andang selaku pembimbing I yang telah banyak meluangkan waktu dan tenaga untuk membimbing penulis selama penyusunan tugas akhir ini.
5. Bapak Nundang Busaeri selaku pembimbing II dan wali dosen saya yang telah banyak meluangkan waktu dan tenaga untuk membimbing saya selama saya berkuliah disini ini.

6. Bapak Ateung selalu pembimbing skripsi saya yang telah memberikan saya support, pencerahan dan bimbingan hingga saya menyelesaikan skripsi ini dengan baik.
7. Bapak / Ibu dosen khususnya Jurusan Teknik Elektro Universitas Siliwangi yang telah membekali penulis dengan disiplin ilmu yang sangat berguna bagi saya.
8. Kepada teman-teman Elektro 2015 yang bersama-sama berjuang dari awal perkuliahan hingga menyelesaikan skripsi ini.
9. Kepada para pejuang skripsi seangkatan saya yaitu: Daus, Agni, Peja, Fredo, Irpan, Fachrim, Viki, Aldy, Rizky, Ivan, Ikhsan dan yang lainnya yang menjadi tanda bukti perjuangan kita selama ini, yang tiada lelah selalu mendukung saya baik tenaga, pikiran, waktu serta selalu-mendoakan saya dalam usaha menyelesaikan skripsi ini.

Rasa hormat dan terimakasih bagi semua pihak atas segala dukungan dan doanya semoga Allah SWT., membalas segala kebaikan yang telah mereka berikan kepada penulis. Amin.

Tasikmalaya, 18 Oktober 2021

Penulis

Octafian Firdaus

HALAMAN PERNYATAAN PERSETUJUAN MENYERAHKAN HAK MILIK ATAS TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Siliwangi, saya yang bertanda tangan di bawah ini:

Nama : Octafian Firdaus

NIM : 157002001

Program Studi : Teknik Elektro

Departemen :

Fakultas : Teknik

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Siliwangi Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty Free Right) atas karya ilmiah saya yang berjudul:

PERANCANGAN DAN PEMODELAN *AIRCRAFT VISUAL DOCKING GUIDANCE SYSTEM* BERBASIS ARDUINO

Dengan hak bebas Royalti Noneksektif ini Universitas Siliwangi berhak menyimpan, mengalihmedia/formatkan, mengembangkan, mengubah, mengelola dalam bentuk pangkalan data (*data base*), merawat, dan memublikasikan tugas akhir saya selama tetap mencabtumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Tasikmalaya

Pada tanggal: 18 Oktober 2021

Yang menyatakan

Matri Rp.6000

(Octafian Firdaus)

ABSTRAK

Nama : Octafian Firdaus

Program Studi : Teknik Elektro

Judul : Perancangan dan Pemodelan *Aircraft Visual Docking Guidance System* Berbasis Arduino

Pertumbuhan transportasi udara dan perkembangan teknologi semakin tahun semakin pesat, hal ini menyebabkan pelayanan bandar udara melampaui kemampuan dalam menyediakan fasilitas untuk memenuhi pertumbuhan secara mencukupi. Salah satu penyebab menurunnya kualitas di bandara adalah kurangnya kemampuan bandara mengelola proses parkir pesawat terbangnya, di setiap bandar udara memiliki seorang *Marshaller* sebagai pemandu parkir pesawat yang dikerjakan hanya oleh satu orang operator yang berada di lintasan bandara. Hingga saat ini beberapa bandara di Indonesia masih menggunakan beberapa orang *Marshaller* sebagai *staff*-nya. Hal ini jika dilihat dari sisi manusiawi terasa kurang tepat. Berdasarkan hal tersebut, dirasa perlu dirancang metode pemanduan parkir pesawat yang dilakukan oleh sebuah sistem yang di rancang sedemikian rupa agar menjamin keamanan dan keselamatan para penumpang pesawat. Metode yang dimaksud adalah *Visual Docking Guidance System*(VDGS). VDGS adalah peralatan yang memandu pesawat terbang secara *visual* menuju ke tempat parkir di *parking stand* secara otomatis. Sistem otomatisasi VDGS dirancang menggunakan Arduino. Untuk mengatur pengukuran jarak pesawat maka VDGS perlu sebuah sensor jarak yaitu sensor laser jarak VL53L0X.

Kata Kunci: *Visual Docking Guidance System, Marshaller, Parkir pesawat, Arduino, Sensor Jarak VL53L0X*

ABSTRACT

Name : Octafian Firdaus

Study Program : Electrical Engineering

Title : Design and Modeling of Aircraft Visual Docking Guidance System with Arduino Based

The evolution of air transportation and technological developments are getting more rapid year to year, this has caused airport services to exceed the ability to provide facilities to meet sufficient growth. One of the causes of the decreasing quality at the airport is the lack of the airport's ability to manage the aircraft parking process, in each airport having a Marshaller as a guide for parking the aircraft which is done by only one operator who is on the airport track. Until now, several airports in Indonesia still use some Marshallers as their staff. This matter if seen from humanity viewpoint feels inappropriate. Based on this, it is deemed necessary to design aircraft parking guidance methods carried out by a system designed in such a way as to ensure the safety and convenience of aircraft passengers. The method that referred is the Visual Docking Guidance System (VDGS). VDGS is equipment that guides the aircraft visually to the parking lot at the parking stand automatically. The VDGS automation system is designed using Arduino. To adjust aircraft distance measurement, VDGS needs a range measurement sensor, the VL53L0X range laser sensor.

Keywords: *Visual Docking Guidance System, Marshaller, Aircraft Parking, Arduino, Laser Range Sensor VL53L0X*

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR	iv
HALAMAN PERNYATAAN PERSETUJUAN AKADEMIS	vi
ABSTRAK.....	vii
<i>ABSTRACT</i>	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xv
DAFTAR RUMUS	xvii
Daftar Lampiran.....	xviii
BAB I PENDAHULUAN	I-1
1.1 Latar Belakang	I-1
1.2 Rumusan Masalah.....	I-3
1.3 Tujuan Penelitian	I-3
1.4 Manfaat Penelitian	I-3
1.5 Batasan Masalah	I-4
1.6 Metodologi Perancangan Dan Pemodelan.....	I-4
1.7 Sistematika Penulisan	I-5
BAB II TINJAUAN PUSTAKA.....	II-1
2.1 Pesawat Terbang Komersil	II-1
2.1.1 <i>Jenis Pesawat Komersil</i>	II-1

2.2	Airport Apron.....	II-6
2.2.1	Standarisasi <i>Apron</i>	II-7
2.2.2	Perencanaan <i>Apron</i>	II-9
2.3	Marshalls.....	II-11
2.4	Visual Docking Guidance System	II-12
2.5	Arduino	II-12
2.5.1	Arduino Mega 2560.....	II-13
2.5.2	Software Arduino.....	II-14
2.6	Sensor VL53L0X.....	II-17
2.7	Laser VCSEL	II-19
2.8	LCD (Liquid Crystal Display)	II-20
2.9	Pengertian Perbandingan atau <i>Ratios</i>	II-20
BAB III METODE PENELITIAN.....		III-1
3.1	Flowchart Penelitian	III-1
3.2	Lokasi Penelitian.....	III-3
3.3	Identifikasi Kebutuhan.....	III-3
3.4	Perancangan dan Pembuatan.....	III-3
3.4.1	Perancangan Prosedur Kerja Sistem.....	III-4
3.4.2	Flowchart Sistem	III-4
3.4.3	Blok Diagram.....	III-5
3.4.4	Diagram Blok Kerja Sistem.....	III-6
3.5	Subjek dan Objek Penelitian.....	III-7
3.6	Metode Pengumpulan Data.....	III-7
BAB IV HASIL DAN PEMBAHASAN		IV-1

4.1	Rancang Bangun Sistem	IV-1
4.1.1	Rancang Bangun Hardware Arduino.....	IV-2
4.1.2	Rancang Bangun Software Arduino	IV-17
4.2	Pengujian Unit	IV-24
4.2.1	Pengujian Arduino	IV-24
4.2.2	Pengujian Sensor jarak VL53L0X.....	IV-27
4.2.3	Pengujian Layar LCD I2C	IV-31
4.2.4	Pengujian Push Button.....	IV-32
4.2.5	Hasil Pengujian Per Unit	IV-33
4.3	Pengujian Sistem.....	IV-35
4.3.1	Pengujian Fungsi Push Button Untuk Mode	IV-36
4.3.2	Pengujian Fungsi Sensor Jarak	IV-36
4.4	Analisis	IV-61
4.5	Kelebihan dan Kekurangan.....	IV-61
4.5.1	Kelebihan	IV-62
4.5.2	Kekurangan.....	IV-63
BAB V KESIMPULAN DAN SARAN.....		V-1
5.1	Kesimpulan	V-1
5.2	Saran	V-2
DAFTAR PUSTAKA		xiv

DAFTAR GAMBAR

Gambar 2.1 Bagian Pesawat Terbang (www.nasa.gov 2010).....	II-1
Gambar 2.2 Wide Body Aircraft(www.nasa.gov 2010)	II-2
Gambar 2.3 Narrow Body Aircraft(www.nasa.gov 2010).....	II-4
Gambar 2.4 Pesawat Cessna 72 (R. Flaugh 2008).....	II-5
Gambar 2.5 Aircraft Parking Limit Lines (Andriani 2015).....	II-7
Gambar 2.6 Komponen Karakteristik Pesawat Terbang(Andriani 2015).....	II-9
Gambar 2.7 Ilustrasi <i>Apron</i> untuk satu pesawat(Andriani 2015).....	II-10
Gambar 2.8 Aircraft Marshaller (Dell and Al 2003).....	II-11
Gambar 2.9 Arduino Mega 2560 (Jin 2015)	II-13
Gambar 2.10 Tampilan Program Arduino Uno(Santoso 2015)	II-15
Gambar 2.11 Tampilan percobaan setup(Santoso 2015)	II-16
Gambar 2.12 Tampilan percobaan loop(Santoso 2015).....	II-17
Gambar 2.13 Sensor VL53L0X (Microelectronics 2018).....	II-18
Gambar 2.14 Pin pada sensor VL53L0X (Maria and Susianti 2019)	II-18
Gambar 2.15 Bagian-bagian laser VCSEL(Amaral et al. 2013).....	II-19
Gambar 2.16 LCD (Liquid Crystal Display)(Fitriandi 2016).	II-20
Gambar 3.1 <i>Flowchart</i> Penelitian	III-1
Gambar 3.2 <i>Flowchat</i> Sistem.....	III-5
Gambar 3.3 Diagram Blok Sistem	III-6
Gambar 3.4 Diagram Blok Kerja Sistem	III-7
Gambar 4.1 Arsitektur Sistem.....	IV-1
Gambar 4.2 Ilustrasi Gambar Boeing 777-300 ER.....	IV-3
Gambar 4.3Sketsa Ukuran Boeing 777-300 ER	IV-4

Gambar 4.4 Ilustrasi Gambar Airbus A330	IV-4
Gambar 4.5 Ilustrasi Ukuran Airbus A330-300.....	IV-6
Gambar 4.6 Ilustrasi Gambar Pesawat Cessna 208 Caravan.	IV-6
Gambar 4.7 Gambar Design Posisi Alat	IV-9
Gambar 4.8 Perancangan Penempatan Sensor pada Boeing 777 – 300 ER.....	IV-9
Gambar 4.9 Perancangan Penempatan Sensor pada Airbus A330-300	IV-9
Gambar 4.10 Perancangan Penempatan Sensor pada Cessna 208 Caravan.....	IV-10
Gambar 4.11 Sketsa Pengukuran Panel.	IV-11
Gambar 4.12 Rancangan 3 Dimensi Panel VDGS.....	IV-12
Gambar 4.13 Wiring Diagram Sistem VDGS dengan Arduino Mega.....	IV-13
Gambar 4.14 <i>Wiring Diagram</i> LCD I2C (<i>Display</i>)	IV-14
Gambar 4.15 <i>Wiring Diagram</i> Sensor Jarak VL5310X.....	IV-15
Gambar 4.16 <i>Wiring Diagram Push Button</i>	IV-16
Gambar 4.17 Program <i>sketch</i> alat VDGS pada Arduino IDE.....	IV-24
Gambar 4.18 Pengujian Arduino ke perangkat Komputer.....	IV-25
Gambar 4.19 Percobaan program <i>blink</i>	IV-25
Gambar 4.20 <i>Sketch</i> untuk pemrograman <i>Test Blink</i>	IV-26
Gambar 4.21 Menyalakan LED pada Arduino Uno.....	IV-26
Gambar 4.22 Percobaan <i>sketch</i> untuk sensor jarak VL53L0X.	IV-27
Gambar 4.23 Pengukuran Sensor Jarak	IV-28
Gambar 4.24 Ilustrasi Percobaan Pengukuran Batas Minimum.	IV-28
Gambar 4.25 Percobaan LCD dengan kalimat “Hello World!”.....	IV-32
Gambar 4.26 Sketch & Percobaan Push Button pada Arduino.....	IV-32
Gambar 4.27 Keberhasilan dalam meng- <i>Upload</i> data.	IV-35

Gambar 4.28 Pengujian Alat VDGS.	IV-35
Gambar 4.29 Monitoring sensor jarak dengan program Arduino IDE.	IV-36
Gambar 4.30 Proses pemarkiran modeling pesawat	IV-37
Gambar 4.31 Skema Pengujian Perintah “LURUS”	IV-37
Gambar 4.32 Sketsa perintah “LURUS”	IV-39
Gambar 4.33 Sketsa Perintah “STOP”	IV-41
Gambar 4.34 Sketsa perintah “KANAN”	IV-44
Gambar 4.35 Skema Pemanduan Perintah “KANAN”	IV-47
Gambar 4.36 Sketsa Perintah “KIRI”.	IV-48
Gambar 4.37 Skema Proses Pemanduan Perintah “KIRI”	IV-51
Gambar 4.38 Sketsa <i>Out of Range</i>	IV-52
Gambar 4.39 Percobaan Pesawat Miring Kekiri.....	IV-54
Gambar 4.40 Percobaan Pesawat Miring Kekanan.....	IV-56
Gambar 4.41 Percobaan Pesawat Datang dari Samping Kiri.....	IV-57
Gambar 4.42 Percobaan Pesawat Datang dari Samping Kanan.....	IV-58
Gambar 4.43 Sketsa Kemiringan Pesawat Berdasarkan Sudut.....	IV-59

DAFTAR TABEL

Tabel 2.1 Data Spesifikasi Pesawat Boeing 777 – 300 ER.....	II-3
Tabel 2.2 Data Spesifikasi Pesawat Airbus A330 - 300	II-5
Tabel 2.3 Spesifikasi Pesawat Cessna 208 Caravan	II-6
Tabel 2.4 Spesifikasi Arduino Mega 2560(Jin 2015)	II-13
Tabel 3.1 Kondisi Aktivitas yang Diharapkan.....	III-4
Tabel 4.1 Hasil Perhitungan Skala pada 3 Jenis Pesawat.	IV-8
Tabel 4.2 Hasil Perhitungan Ukuran Purwarupa Alat VDGS.....	IV-11
Tabel 4.3 Wiring Pin Kaki Modul LCD I2C Pada Arduino	IV-13
Tabel 4.4 Wiring Pin Sensor Jarak VL53L0X.....	IV-15
Tabel 4.5 <i>Wiring Pin Push Button</i>	IV-16
Tabel 4.6 Pengukuran Batas Minimum Sensor.....	IV-29
Tabel 4.7 Pengukuran Batas Maksimum Sensor.....	IV-31
Tabel 4.8 Hasil Percobaan pada Tombol Push Button.....	IV-33
Table 4.9 Tabel Hasil Kinerja Komponen	IV-34
Tabel 4.10 Data Penentuan Mode pada Push Button.....	IV-36
Tabel 4.11 Pengujian Perintah “LURUS”.....	IV-38
Tabel 4.12 Percobaan Pemanduan Perintah “LURUS”	IV-40
Tabel 4.13 Pengujian Perintah “STOP”	IV-42
Tabel 4.14 Pengujian Batas Jarak Perintah “KANAN”.....	IV-45
Tabel 4.15 Percobaan Pemanduan Pesawat pada Perintah “KANAN”	IV-47
Tabel 4.16 Hasil Pengujian Perintah “KIRI”	IV-49
Tabel 4.17 Percobaan Pemanduan Perintah “KIRI”.	IV-51
Tabel 4.18 Percobaan <i>Out of Range</i>	IV-53

Tabel 4.19 Sketsa Percobaan Kemiringan.	IV-54
Tabel 4.20 Data Sensor Percobaan Pesawat Miring Kekiri.	IV-55
Tabel 4.21 Data Sensor Percobaan Pesawat Miring Kekanan.	IV-56
Tabel 4.22 Data Sensor Percobaan Pesawat Datang dari Samping Kiri.	IV-57
Tabel 4. 22 Data Sensor Percobaan Pesawat Datang dari Samping Kanan.	IV-58
Tabel 4.23 Pengujian Kemiringan Sudut.	IV-60

DAFTAR RUMUS

Rumus <i>Aeroplane Reference Field Length</i> (2.1).....	II-8
Rumus Kebutuhan Luas dari <i>Apron</i> (2.2)	II-10
Rumus Perbandingan (2.3).....	II-20
Rumus Perbandingan Senilai 1 (2.4).....	II-21
Rumus Perbandingan Senilai 2 (2.5).....	II-21
Rumus Perbandingan Senilai dengan Konsep Pembagi (2.6).....	II-21
Rumus Perbandingan Senilai 3 (2.7).....	II-21
Perbandingan Berbalik Nilai 1 (2.8)	II-21
Perbandingan Berbalik Nilai 2 (2.9)	II-22
Rumus Perbandingan Berbalik Nilai dengan Konsep Pembagi (2.10).....	II-22
Rumus Perbandingan Berbalik Nilai 3 (2.11).....	II-22
Rumus Menentukan Ukuran Lebar Kabin Pesawat (4.1).....	IV-3
Rumus Menentukan Ukuran Lebar Badan Pesawat (4.2).....	IV-4

DAFTAR LAMPIRAN

Gambar 1 Percobaan Alat VDGS	xvii
Gambar 2 <i>Wiring</i> Alat VDGS.....	xvii
Gambar 3 Percobaan LCD I2C pada Alat VDGS.....	xviii
Gambar 4 Pemasangan Sensor VL53L0X.	xviii
Gambar 5 Tampilan Alat VDGS.....	xix

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pertumbuhan transportasi udara dan perkembangan teknologi semakin tahun semakin pesat, hal ini menyebabkan pelayanan bandar udara melampaui kemampuan dalam menyediakan fasilitas untuk memenuhi pertumbuhan secara mencukupi. Sebagian besar sistem transportasi udara di bandar udara ditekan melebihi kemampuan kapasitas rancangan bandar udara yang telah ada, sehingga mengakibatkan menurunnya kualitas penerbangan di bandar udara. Dalam transportasi udara, perhatian khusus diberikan pada pergerakan pesawat terbang, penumpang, dan barang, baik melalui bandar udara maupun sistem penerbangan(Denny Dermawan; M. Jalu Purnomo 2016).

Salah satu penyebab menurunnya kualitas di bandara adalah kurangnya kemampuan bandara mengelola proses pemarkiran pesawat terbangnya, di setiap bandar udara memiliki seorang *Marshaller* sebagai pemandu pemarkiran pesawat. Seorang *Marshaller* yang berguna untuk mengatur pergerakan dengan tujuan menghindari tabrakan antara pesawat udara dengan halangan, mengatur jalur masuk pesawat udara ke *parking stand*, menjamin keselamatan, dan pengaturan kegiatan lainnya dikerjakan hanya oleh satu orang operator yang berada di lintasan bandara(PT Angkasa Pura II (Persero) 2013).

Hingga saat ini beberapa bandara di Indonesia masih menggunakan beberapa orang *Marshaller* sebagai *staff*-nya. Hal ini jika dilihat dari sisi manusiawi terasa kurang tepat, mengingat lingkungan bandara yang terpengaruh cuaca panas dan hujan, juga polusi udara dan suara yang cukup bising, dimana pesawat harus

berhenti dengan tepat berada di jangkauan boarding bridge yang hanya bisa berbelok dengan sudut -15° sampai 15° dari sudut awalnya. Panjang badan, tinggi dan posisi pintu pesawat berbeda-beda ukuran disetiap jenis pesawat. Namun hal tersebut masih terbilang tidak cukup efisien untuk menanggulangi kapasitas yang di butuhkan di tiap landasannya dimana seorang *Marshaller* hanya bisa berada di satu tempat di tiap landasan parkirnya(Fatonah 2013).

Berdasarkan hal tersebut maka dipandang perlu dibuatkan *Visual Docking Guidance System* (VDGS) yang berarti adalah pemandu parkir pesawat visual otomatis yang akan mempermudah proses pemandun parkir pesawat dan juga meringankan *Marshaller*. Maka dari itu judul dari penelitian ini adalah **“PERANCANGAN DAN PEMODELAN AIRCRAFT VISUAL DOCKING GUIDANCE SYSTEM”** ini untuk memudahkan proses parkir hingga kenyamanan dan keamanan seluruh penumpang tanpa menggunakan tenaga manusia.

1.2 Rumusan Masalah

Adapun perumusan masalah dari alat ini adalah sebagai berikut:

1. Bagaimana kinerja VDGS dalam proses pemandu parkir pesawat terbang di bandara.
2. Bagaimana cara membedakan jenis pesawat berdasarkan ukurannya.
3. Bagaimana cara mendeteksi kesalahan yang terjadi pada saat proses pemanduan pesawat.

1.3 Tujuan Penelitian

Adapun tujuan dari perancangan dan pemodelan alat ini adalah sebagai berikut:

1. Mengidentifikasi proses pemanduan pesawat dengan menggunakan alat VDGS ini.
2. Merancang pengaturan yang membedakan jenis pesawat berdasarkan ukuran pesawatnya.
3. Menentukan kesalahan yang kemungkinan akan terjadi pada proses pemanduan pesawat.

1.4 Manfaat Penelitian

Perancangan dan Pemodelan ini dapat bermanfaat sebagai berikut:

1. Bermanfaat untuk membantu pilot pesawat untuk mensimulasikan proses pemanduan parkir pesawat yang lebih akurat.
2. Penelitian ini bisa bermanfaat bagi bandara untuk mengurangi tingkat kesalahan yang terjadi pada saat proses pemanduan pesawat.

3. Dapat menjadi bahan acuan dalam pemanduan alat transportasi besar lainnya yang membutuhkan akurasi dan presisi yang baik.

1.5 Batasan Masalah

Untuk membantu proses pemarkiran pesawat terbang, maka dirancang alat VDGS yang dapat berguna di *Parking Stand* di tiap bandara. Untuk membatasi permasalahan pada alat ini, maka ada beberapa batasan masalah dalam penelitian ini, yaitu:

1. Penelitian akan dilakukan dengan menggunakan simulasi alat sederhana dan berupa purwarupa.
2. Berfokus pada pemanduan pesawat yang terdeteksi oleh alat VDGS ini.
3. Dibatasi untuk pesawat terbang komersil saja.
4. Sistem yang di buat menentukan tipe pesawat secara manual.

1.6 Metodologi Perancangan Dan Pemodelan

Metodologi Perancangan dan Pemodelan yang digunakan untuk mencapai keberhasilan Perancangan dan Pemodelan ini berupa:

1. Studi Literatur, yaitu dengan mencari dan mengumpulkan kajian-kajian dan literature-literatur yang berkaitan dengan Perancangan dan Pemodelan ini, berupa artikel, buku referensi, jurnal perancangan dan pemodelan terkait, dan sumber lainnya yang berhubungan.
2. Perancangan Sistem, merancang sistem berdasarkan tujuan Perancangan dan Pemodelan.
3. Pemodelan Sistem, memodelkan implementasi sistem yang telah dirancang.

4. Perakitan model alat, merakit alat dan bahan menjadi 1 bagian.
5. Pengujian Pemodelan Sistem, menguji pemodelan sistem yang telah dibuat.
6. Menganalisis Hasil, menganalisis hasil yang diperoleh dari pengujian sistem yang telah dirancang.
7. Kesimpulan, memberikan Kesimpulan berdasarkan hasil dan analisa yang telah dilakukan.

1.7 Sistematika Penulisan

Sistematika penulisan yang akan digunakan untuk menyelesaikan Perancangan dan Pemodelan sebagai tugas akhir ini meliputi:

BAB 1 PENDAHULUAN: Bab ini menguraikan mengenai latar belakang, perumusan masalah, tujuan Perancangan dan Pemodelan, manfaat Perancangan dan Pemodelan, batasan masalah, metodologi Perancangan dan Pemodelan, dan sistematika penulisan.

BAB II LANDASAN TEORI: Bab ini membahas mengenai dasar-dasar teori dari komponen/perangkat dan system yang digunakan.

BAB III METODOLOGI PERANCANGAN DAN PEMODELAN: Bab ini akan menguraikan metode-metode dan tahap-tahap yang dilakukan dalam melakukan Perancangan dan Pemodelan ini.

BAB IV PERANCANGAN APLIKASI: Bab ini berisikan analisa dan perancangan aplikasi perangkat lunak yang mendukung keseluruhan modul, sistem, dan implementasi hasil dan membahas sistem yang dirancang dan dibangun.

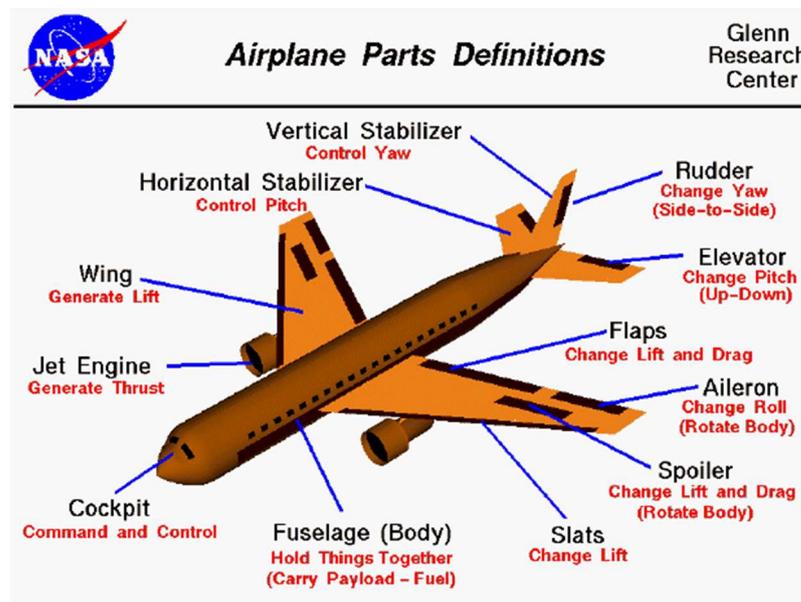
BAB V HASIL DAN PEMBAHASAN: Bab ini berisikan Kesimpulan dari keseluruhan Perancangan dan Pemodelan sistem sebagai tugas akhir ini.

BAB II

TINJAUAN PUSTAKA

2.1 Pesawat Terbang Komersil

Pesawat terbang komersil adalah pesawat udara yang digunakan untuk mengangkut penumpang sipil beserta barang-barang di bagasi. Syarat-syarat mengenai pengangkutan sipil diatur dalam undang-undang, baik pemerintah maupun internasional melalui lembaga PBB bernama ICAO, Gambar 2.1 memberikan contoh pesawat terbang komersil yang biasanya di gunakan. (*International Civil Aviation Organization*)(Kesuma, Silaban, and Firmansyah 20AD).



Gambar 2.1 Bagian Pesawat Terbang (www.nasa.gov 2010)

2.1.1 Jenis Pesawat Komersil

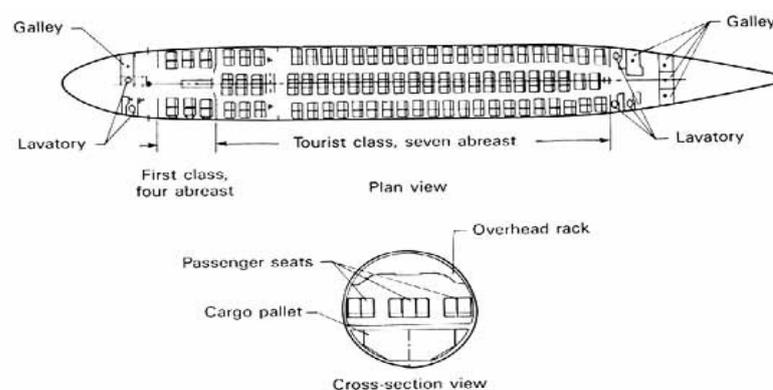
Pesawat penerbangan sipil memiliki banyak jenis tipe pesawat yang diproduksi oleh berbagai negara, dengan spesifikasi dan karakteristik yang berbeda, dalam segi ukuran dan batas penumpang yang di tampung. Jenis pesawat penerbangan sipil serta spesifikasi dan karakteristik pesawat yang sudah ada

memiliki tiga tipe, yakni *wide body craft*, *narrow body craft* dan pesawat perintis(Kesuma, Silaban, and Firmansyah 20AD).

1. Wide Body Aircraft

Gambar 2.2 menunjukkan sketsa dari *Wide Body Aircraft* yang adalah pesawat dengan lebar lebih dari 20 kaki, mempunyai dua lorong atau biasa juga dikenal dengan *twin-aisles*, dengan tujuh kursi atau lebih sejajar. Diameter pesawat ini biasanya mencapai lima atau enam meter. Dalam kabin bagian ekonomi, dapat menampung tempat duduk dengan susunan 3-4-3 atau 2-3-2, dengan total kapasitas mencapai 200 hingga 850 penumpang. Jenis ini memiliki pesawat terlebar yang mencapai enam meter dan dapat menampung hingga 11 penumpang sejajar. Rata-rata *wide body aircraft* memiliki izin terbang *trans-atlantik* dan *trans-kontinental* sehingga biasanya digunakan untuk penerbangan jarak menengah dan jarak jauh. Pesawat jenis ini juga memerlukan landasan yang jauh lebih panjang(Kesuma, Silaban, and Firmansyah 20AD).

Model-model *wide body aircraft* diantaranya adalah Airbus A300 seperti pada Gambar 2.2, Airbus A330, Airbus A340, Airbus A350, Airbus A380, Boeing 747, Boeing 767, Boeing 777, Boeing 787 Dreamliner, Ilyushin Il-86, Ilyushin Il-96, L1011 Tristar, MD DC-10(Kesuma, Silaban, and Firmansyah 20AD).



Gambar 2.2 Wide Body Aircraft(www.nasa.gov 2010)

Boeing 777 – 300 ER

Boeing 777 adalah pesawat berbadan lebar Amerika yang dikembangkan dan diproduksi oleh Boeing Commercial Airplanes. Ini adalah twinjet terbesar di dunia. 777 dirancang untuk menjembatani kesenjangan antara Boeing 767 dan 747, dan untuk menggantikan DC-10 dan L-1011 yang lebih tua. Dikembangkan melalui konsultasi dengan delapan maskapai besar, dengan pertemuan pertama pada Januari 1990, program ini diluncurkan pada 14 Oktober 1990 dengan pesanan dari United Airlines. Prototipe diluncurkan pada 9 April 1994, dan pertama kali terbang pada 12 Juni 1994. 777 memasuki layanan dengan pelanggan peluncuran, United Airlines, pada 7 Juni 1995. Varian jarak yang lebih jauh diluncurkan pada 29 Februari 2000 dan telah pertama kali dikirim pada tanggal 29 April 2004. Berikut adalah data spesifikasi pesawat Boeing 777 – 300 ER pada Tabel 2.1 (Grantham, Russell 2008).

Tabel 2.1 Data Spesifikasi Pesawat Boeing 777 – 300 ER

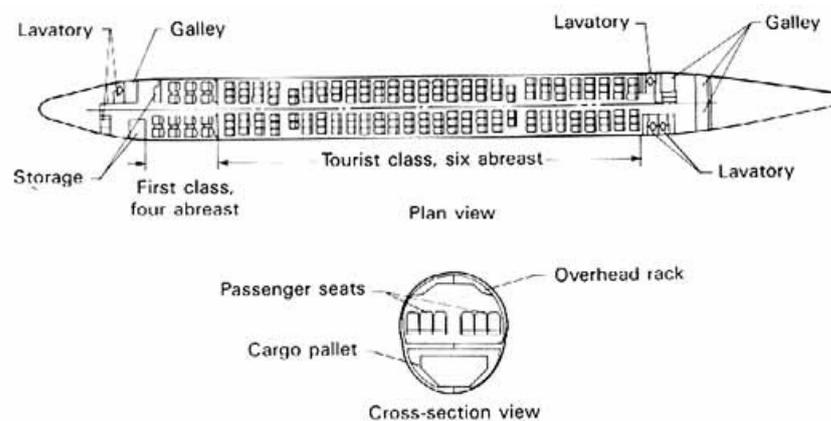
Boeing 777- 300 ER		
Kru Kokpit	2	Orang
Panjang Pesawat	73.86	Meter
Bentang Sayap	64.80	Meter
Tinggi	18.5	Meter
Lebar Badan (diameter)	6.20	Meter
Lebar Kabin (diameter)	5.86	Meter

2. Narrow Body Aircraft

Narrow body aircraft yang biasanya juga dikenal dengan sebutan pesawat lorong tunggal – adalah pesawat dengan lebar kabin yang biasanya mencapai tiga sampai empat meter seperti pada Gambar 2.3. Hanya memiliki satu lorong,

penempatan tempat duduk pesawat ini biasanya 3-3, 2-2 atau terkadang 2-1 bahkan 1-1 untuk private jet dengan kapasitas kurang dari 250 orang. *Narrow body aircraft* umumnya tidak memiliki izin terbang *trans-atlantik* atau *trans-kontinental* dan hanya digunakan untuk penerbangan *regional* (Kesuma, Silaban, and Firmansyah 20AD).

Model-model *narrow body aircraft* diantaranya adalah: Airbus A320 family, Boeing 707, Boeing 720, Boeing 727, Boeing 737, Boeing 757, Douglas DC-8, Tupolev Tu-154, Tupolev Tu-204, Tupolev Tu-334, Sukhoi SSJ 100, Vickers VC10, BAC One-Eleven, BAe 146, Boeing 717, Bombardier C Series, Convair 880 dan 990, Fokker F28, Fokker 70, Fokker 100, McDonnell Douglas DC-9, McDonnell Douglas MD-80/MD-90, Antonov 148, Antonov 158, ATR-72, MA-60, dsb (Kesuma, Silaban, and Firmansyah 20AD).



Gambar 2.3 Narrow Body Aircraft (www.nasa.gov 2010)

Airbus A330 – 300

Airbus A330 adalah pesawat berbadan lebih sempit dibandingkan jenis Boeing 777 yang dirancang dan diproduksi oleh Airbus. Airbus membuat beberapa turunan dari A300, pesawat pertamanya pada pertengahan 1970-an. Kemudian perusahaan mulai mengembangkan twinjet A330 secara paralel dengan quadjet A340 dan

meluncurkan kedua desain tersebut dengan pesanan pertama pada Juni 1987. A330-300, varian pertama, melakukan penerbangan perdananya pada November 1992 dan mulai beroperasi dengan Air Inter pada Januari 1994. Varian A330-200 yang sedikit lebih pendek menyusul pada tahun 1998. Berikut adalah data spesifikasi dari Airbus A330 – 300(Tabel 2.2) (Norris & Wagner 2001).

Tabel 2.2 Data Spesifikasi Pesawat Airbus A330 - 300

Airbus A330-300		
Kru Kokpit	2	Orang
Panjang Pesawat	63.67	Meter
Bentang Sayap	60.3	Meter
Tinggi	16.79	Meter
Lebar Badan (diameter)	5.64	Meter
Lebar Kabin (diameter)	5.26	Meter

3. Pesawat Perintis

Seperti yang terlihat pada gambar 2.4, pesawat ini berukuran kecil dengan berat kurang dari 6 ton. Pesawat jenis ini biasanya digunakan untuk menjangkau daerah-daerah terpencil. Karena ukuran badan yang kecil, pesawat ini dapat mendarat di landasan pendek dan landasan tanah. Pesawat yang digunakan biasanya jenis Cessna, Beechcraft, Grumman, dan lain-lain(Kesuma, Silaban, and Firmansyah 20AD).



Gambar 2.4 Pesawat Cessna 72 (R. Flaugh 2008)

Cessna 208 Caravan

Cessna 208 Caravan adalah pesawat utilitas yang diproduksi oleh Cessna. Proyek ini dimulai pada tanggal 20 November 1981, dan prototipe pertama kali terbang pada tanggal 9 Desember 1982. Model produksi disertifikasi oleh FAA pada bulan Oktober 1984 dan varian kargo *Cargomaster* dikembangkan untuk FedEx. *208B Super Cargomaster* yang berukuran 4 kaki (1,2 m) lebih panjang pertama kali terbang pada tahun 1986 dan dikembangkan menjadi *208B Grand Caravan* penumpang (Goyer, Robert 2008).

Tabel 2.3 menunjukkan Cessna 208 yang bisa menopang biasanya menampung sembilan penumpang di kabinnya yang tidak bertekanan, ditenagai oleh turboprop traktor *Pratt & Whitney Canada PT6A* tunggal dan memiliki roda pendaratan roda tiga tetap, pelampung atau ski. Hingga November 2017, 2.600 telah dikirimkan dan 20 juta jam penerbangan telah dicatat. Karavan digunakan untuk pelatihan penerbangan, maskapai komuter, transportasi VIP, kargo udara, dan misi penyelamatan (Goyer, Robert 2008).

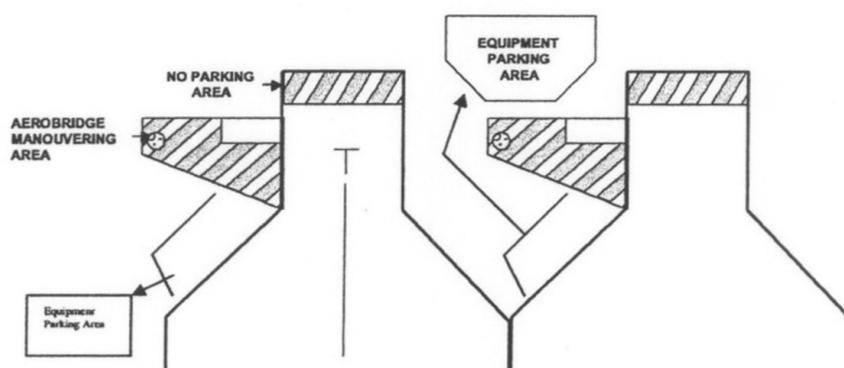
Tabel 2.3 Spesifikasi Pesawat Cessna 208 Caravan

Cessna 208 Caravan		
Kru Kokpit	1 atau 2	Orang
Panjang Pesawat	11.46	Meter
Bentang Sayap	15.87	Meter
Tinggi	4.5	Meter
Tinggi Kabin	1.37	Meter
Lebar Kabin	1.63	Meter

2.2 Airport Apron

Apron adalah bagian dari lapangan gerak darat suatu Bandar udara yang berfungsi untuk menaikkan dan menurunkan penumpang dan muatan, pengisian bahan bakar, parkir, dan persiapan pesawat terbang sebelum melanjutkan

penerbangan. Ukuran dan letak *gate* harus direncanakan dengan memperhatikan karakter pesawat yang menggunakan *gate* tersebut seperti lebar sayap, panjang, dan radius belok pesawat, dan juga area yang diperlukan oleh kendaraan-kendaraan yang menyediakan servis untuk pesawat selama berada di *gate* (Gambar 2.5). Untuk menjamin keselamatan pesawat di daratan, ICAO dan FAA juga menetapkan persyaratan jarak minimum antara pesawat terbang yang sedang parkir di *apron* satu sama lainnya, dengan bangunan, atau obyek-obyek tetap lainnya yang ada di *apron* berdasarkan jarak sayap pesawat/*wingtip clearance* (Denny Dermawan; M. Jalu Purnomo 2016).



Gambar 2.5 Aircraft Parking Limit Lines (Andriani 2015)

2.2.1 Standarisasi *Apron*

Berdasarkan karakteristik pesawat, Menurut Horonjeff (1993) bentang sayap dan panjang badan pesawat mempengaruhi ukuran *apron* parkir, yang akan mempengaruhi susunan gedung-gedung terminal (Gambar 2.6). Ukuran pesawat juga menentukan lebar *runway*, *taxiway* dan jarak antara keduanya, serta mempengaruhi jari-jari putar yang dibutuhkan pada kurva-kurva perkerasan (SKEP 77-VI-2005 Dirjen Perhubungan Udara 2005).

ARFL (*Aeroplane Reference Field Length*) adalah panjang landasan minimum yang diperlukan untuk lepas landas suatu pesawat terbang dengan ketentuan yang ada, setiap jenis pesawat terbang mempunyai ARFL sendiri yang tercantum dalam manual yang dibuat oleh pabrik yang memproduksi pesawat tersebut(SKEP 77-VI-2005 Dirjen Perhubungan Udara 2005).

$$ARFL = (ARFL_{rencana} * Ft * Fe * Fs) + Fw \dots \dots \dots (2.1)$$

Dimana:

ARFL_{rencana} = Panjang runway rencana (m)

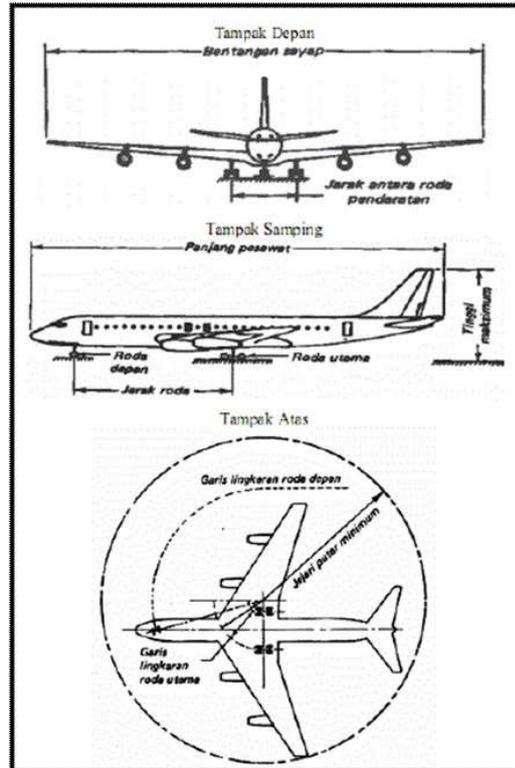
Ft =Faktor koreksi temperatur

Fe = Faktor koreksi ketinggian

Fs = Faktor koreksi kemiringan

Fw = Faktor koreksi angin permukaan

Pesawat rencana yang akan digunakan sebagai percobaan pengukuran adalah Airbus A – 380 dengan karakteristik teknis(Gambar 2.6):



Gambar 2.6 Komponen Karakteristik Pesawat Terbang (Andriani 2015)

Jarak antara roda pendaratan = 3.350 m

Bentangan sayap = 79,8 m

Jarak antar roda depan dan utama = 7,14 m

Rata-rata Panjang Pesawan = 73 m

Berat *take off* maksimum = 590.000 kg

2.2.2 Perencanaan *Apron*

Seperti pada ilustrasi Gambar 2.7, *Apron* merupakan tempat pesawat parkir yang konstruksinya serupa dengan *Runway* dan *Taxiway*. *Apron* berfungsi sebagai tempat naik dan turunnya penumpang atau barang dari maupun ke pesawat. Tempat pelataran parkir pesawat harus tidak melanggar pembatas rintangan yang berada dipermukaan dan terutama didalam. Ukuran pelataran parkir pesawat harus cukup untuk dapat melayani arus lalu lintas maksimum yang diperlukan. Untuk

perencanaan apron, diambil nilai berdasarkan wingspan jenis pesawat Airbus A380(Andriani 2015), dimana:

Panjang sayap = 79,8 m

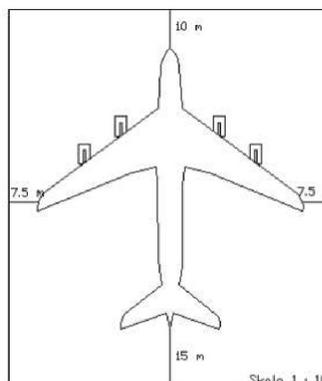
Panjang badan = 72,73 m

Kode tipe pesawat = Airbus A-380.

Jarak Bebas Pesawat dan Bangunan Terminal Berdasarkan SKEP/77/VI/2005, jarak antara pesawat dengan bangunan lain adalah 10m. Sehingga dalam desain ini digunakan jarak bebas 10 m. Maka luas apron yang dibutuhkan untuk satu pesawat dengan lebar sayap A380 (SKEP 77-VI-2005 Dirjen Perhubungan Udara 2005) yaitu:

$$\begin{aligned}
 \text{Apron} &= (\text{wing span} + \text{jarak ruang}) \\
 &\quad * (\text{panjang badan pesawat} + \text{jarak bebas}) \dots \dots \dots (2.2) \\
 &= (79,8 + 15) * (72,73 + 10 + 15) \\
 &= 9170 \text{ m}^2
 \end{aligned}$$

Maka luas kebutuhan *Apron* untuk pesawat jenis Airbus A-380 adalah 9170 m².



Gambar 2.7 Ilustrasi *Apron* untuk satu pesawat(Andriani 2015)

2.3 Marshallers

Marshaller adalah seorang yang memberikan komando kepada pilot pesawat terbang untuk memarkirkan kendaraannya ketika pesawat terbang mendarat di landasan pacu (Gambar 2.8). Seorang *marshaller*, harus benar dan tepat dalam memberikan tanda kepada pilot. Karena untuk memarkirkan pesawat terbang yang memiliki ukuran besar harus berhati-hati, dan tidak boleh lengah (Quarta Aviation Training Center 2016).

Pada tahun 1990, Kantor Federal Swiss untuk Penerbangan Sipil (FOCA) dan Organisasi Asuransi Kecelakaan Nasional Swiss (Suva) membuat proyek "*Training of Marshalls*". Hal ini dilakukan sebagai tanggapan terhadap peningkatan dramatis dalam gerakan helikopter sejak penerbangan helikopter diperkenalkan pada tahun 1983 dan mengakibatkan peningkatan serius dalam jumlah kecelakaan (Dell and Al 2003).

Marshalling adalah satu-satunya komunikasi visual dan bagian dari penanganan pesawat udara di landasan. Peralatan yang biasa digunakan marshaller adalah rompi keselamatan, helm dengan penutup telinga akustik yang berfungsi melindungi telinga dari kebisingan mesin pesawat, dan sarung tangan atau tongkat *Marshalling* (Dell and Al 2003).



Gambar 2.8 Aircraft Marshaller (Dell and Al 2003)

2.4 Visual Docking Guidance System

Visual Docking Guidance System atau bisa di sebut juga sebagai VDGS adalah peralatan yang memandu pesawat terbang secara *visual* menuju ke tempat parkir di *parking stand* secara otomatis. Sistem ini dirancang untuk membantu meringankan proses pemanduan parkir *docking* dengan cepat halus dan presisi sampai pada gate terminal. Sensor yang bekerja mendeteksi adanya pesawat yang masuk landasan mengontrol pesawat terbang untuk memastikan apakah sesuai dengan informasi yang diberikan untuk *docking*(Minohara 2005).

Alat ini menampilkan informasi yang jelas dan terlihat oleh pilot pada layar intensitas tinggi LED untuk mengarahkan proses *docking* yang benar. Prosedur *docking* di bandara-bandara Indonesia saat ini sebagian besar masih menggunakan *Marshaller* sebagai pemandu utamanya, namun proses ini mampu lebih cepat dengan menggunakan peralatan VDGS ini. Parkir tanpa menggunakan *Marshaller* akan lebih efektif dikarenakan penggunaan *Marshaller* akan terkendala pada saat cuaca buruk dan tidak memungkinkan untuk staff yang sama mengatasi tugas-tugas lain di landasan lainnya, serta meminimalkan pergerakan jumlah petugas bandara udara yang bekerja di sekitar landasan parkir dengan alasan safety (Denny Dermawan; M. Jalu Purnomo 2016).

2.5 Arduino

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardware dalam arduino memiliki prosesor Atheros AR9331 dan menggunakan software dan bahasa sendiri. Ini adalah papan mikrokontroler yang dikembangkan

oleh Arduino.cc dan berdasarkan pada Atmega. Perangkat elektronik menjadi kompak, fleksibel yang mampu melakukan lebih banyak fungsi (Lee 2018).

2.5.1 Arduino Mega 2560

Tabel 2.4 menunjukkan bahwa Arduino Mega 2560 adalah papan mikrokontroler berdasarkan ATmega2560 (*data sheet*). Arduino Mega 2560 (Gambar 2.9) memiliki 54 pin input / output digital (14 di antaranya dapat digunakan sebagai output PWM), 16 input analog, 4 UART (port serial perangkat keras), osilator kristal 16 MHz, koneksi USB, colokan listrik, header ICSP, dan tombol reset. Ini berisi semua yang diperlukan untuk mendukung mikrokontroler; cukup sambungkan ke komputer dengan kabel USB atau daya dengan adaptor AC-to-DC atau baterai untuk memulai. Mega kompatibel dengan sebagian besar perisai yang dirancang untuk Arduino Duemilanove atau Diecimila (Jin 2015).



Gambar 2.9 Arduino Mega 2560 (Jin 2015)

Tabel 2.4 Spesifikasi Arduino Mega 2560(Jin 2015)

Spesifikasi Teknis	
Mikrokontroler	ATmega2560
Operate Voltage	5 V
Input Voltage	7 V – 12 V
Input Voltage (Limits)	6 V- 20 V
Digital I/O Pins	54 (14 Pin PWM)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3,3 V Pin	50 mA

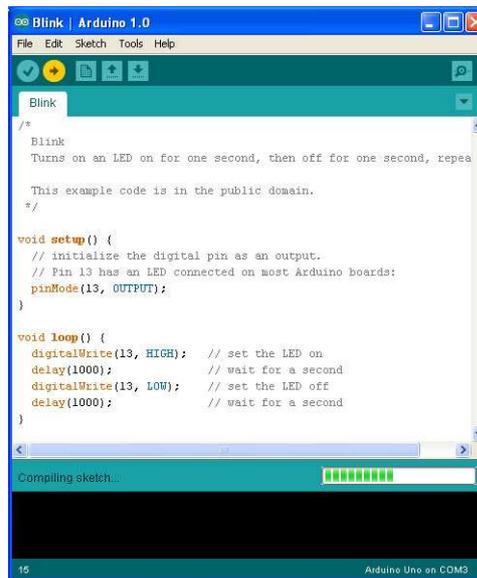
Flash Memory	256 KB (8 KB used by bootloader)
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

2.5.2 Software Arduino

Software arduino yang digunakan adalah driver dan IDE, walaupun masih ada beberapa software lain yang sangat berguna selama pengembangan arduino. *Integrated Development Environment* (IDE), suatu program khusus untuk suatu komputer agar dapat membuat suatu rancangan atau sketsa program untuk papan Arduino. IDE arduino merupakan *software* yang sangat canggih ditulis dengan menggunakan java (Gustomo 2015).

1. *Editor* Program Sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.
2. *Compiler* Berfungsi untuk kompilasi *sketch* tanpa unggah ke board bisa dipakai untuk pengecekan kesalahan kode sintaks *sketch*. Sebuah modul yang mengubah kode program menjadi kode biner bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *processing*.
3. *Uploader* Berfungsi untuk mengunggah hasil kompilasi *sketch* ke *board target*. Pesan *error* akan terlihat jika *board* belum terpasang atau alamat *port* COM belum terkonfigurasi dengan benar. Sebuah modul yang memuat kode biner dari komputer ke dalam *memory* didalam papan arduino.

2.5.2.1 Program Arduino IDE



Gambar 2.10 Tampilan *Sketch* Program Arduino Uno(Santoso 2015)

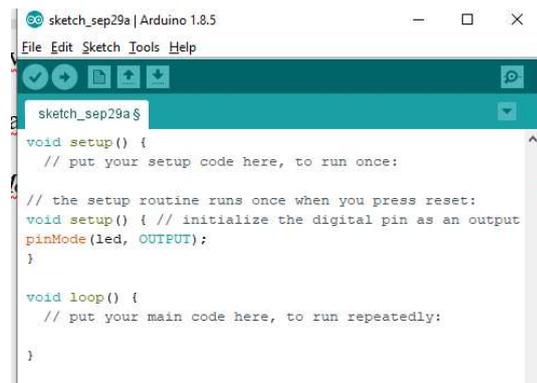
Kode *Program* Arduino biasa disebut *Sketch*(Gambar 2.10) dan dibuat menggunakan bahasa pemrograman C. Program atau *sketch* yang sudah selesai ditulis di Arduino IDE bisa langsung dicompile dan diupload ke Arduino Board. Secara sederhana, *sketch* dalam Arduino dikelompokkan menjadi 3 blok (lihat gambar di atas)(Santoso 2015):

1. Header

Pada bagian ini biasanya ditulis definisi-definisi penting yang akan digunakan selanjutnya dalam program, misalnya penggunaan library dan pendefinisian variable. Kode dalam blok ini dijalankan hanya sekali pada waktu *compile*. Di bawah ini contoh code untuk mendeklarasikan variable led (integer) dan sekaligus di isi dengan angka 13 `int led = 13`(Santoso 2015).

2. Setup

Di saat awal, atau ketika *power on Arduino board*. Biasanya di blok ini diisi penentuan apakah suatu pin digunakan sebagai *input* atau *output*, menggunakan perintah *pinMode*(Gambar 2.11) (Santoso 2015).



```

sketch_sep29a | Arduino 1.8.5
File Edit Sketch Tools Help
sketch_sep29a $
void setup() {
  // put your setup code here, to run once:
  // the setup routine runs once when you press reset:
void setup() { // initialize the digital pin as an output
pinMode(led, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
}

```

Gambar 2.11 Tampilan percobaan setup(Santoso 2015)

OUTPUT adalah suatu makro yang sudah didefinisikan Arduino yang berarti = 1. Jadi perintah di atas sama dengan `pinMode(led, 1);` Suatu pin bisa difungsikan sebagai *OUTPUT* atau *INPUT*. Jika difungsikan sebagai output, dia siap mengirimkan arus listrik (maksimum 100 mA) kepada beban yang disambungkannya. Jika difungsikan sebagai *INPUT*, pin tersebut memiliki *impedance* yang tinggi dan siap menerima arus yang dikirimkan kepadanya(Saghoa, Somppie, and Tulung 2018).

3. Loop

Blok ini akan dieksekusi secara terus menerus. Apabila program sudah sampai akhir blok, maka akan dilanjutkan dengan mengulang eksekusi dari awal blok. Program akan berhenti apabila tombol *power Arduino* di matikan(Gambar 2.12) (Saputro and Sukmadi 2013).



```

sketch_sep29a | Arduino 1.8.5
File Edit Sketch Tools Help
sketch_sep29a $
void setup() {
  // put your setup code here, to run once:

  // the setup routine runs once when you press reset:
void setup() { // initialize the digital pin as an output
pinMode(led, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
void loop() {
digitalWrite(led, HIGH); // nyalakan LED
delay(1000); // tunggu 1000 milidetik
digitalWrite(led, LOW); // matikan LED
delay(1000); // tunggu 1000 milidetik
}

```

Gambar 2.12 Tampilan percobaan loop(Santoso 2015)

Perintah *digital Write* (pinNumber, nilai) akan memerintahkan arduino untuk menyalakan atau mematikan tegangan di *pinNumber* tergantung nilainya. Jadi perintah di atas *digital Write (led, HIGH)* akan membuat pin nomor 13 (karena di *header* dideklarasikan led = 13) memiliki tegangan = 5V (*HIGH*). Hanya ada dua kemungkinan nilai *digitalWrite* yaitu *HIGH* atau *LOW* yang sebetulnya adalah nilai integer 1 atau 0. Pasang kabel USB yang diikutsertakan pada saat membeli Arduino, pasang ke komputer dan board arduino, dan upload programnya. Lampu LED yg ada di Arduino board kita akan kelap-kelip. Sekedar informasi, sebuah LED telah disediakan di board Arduino Uno dan disambungkan ke pin 13. Selain blok *setup()* dan *loop()* di atas kita bisa mendefinisikan sendiri blok fungsi sesuai kebutuhan(Santoso 2015).

2.6 Sensor VL53L0X

VL53L0X adalah modul pengukur jarak *Time-of-Flight* (ToF) generasi baru dengan ukuran terkecil saat ini, memberikan pengukuran jarak yang akurat apa pun target yang dipantulkan tidak seperti teknologi konvensional. VL53L0X dapat mengukur jarak absolut hingga 1,3 m (Maria and Susianti 2019).

VL53L0X(Gambar 2.13) mengintegrasikan array SPAD terdepan (*Single Photon Avalanche Diodes*). Emitor VCSEL 940 nm VL53L0X (Laser Pemancar Permukaan Vertikal) ini tidak bisa dilihat oleh mata manusia, ditambah dengan filter inframerah fisik internal, memungkinkan jarak yang lebih jauh, kekebalan yang lebih tinggi terhadap cahaya sekitar, dan ketahanan yang lebih baik untuk menutupi *Cross-Talk* optik kaca.



Gambar 2.13 Sensor VL53L0X (Microelectronics 2018)

Sensor VL53L0X(Gambar 2.14) memiliki pin-pin yang dapat digunakan untuk memasukan database ke Arduino(Microelectronics 2018):

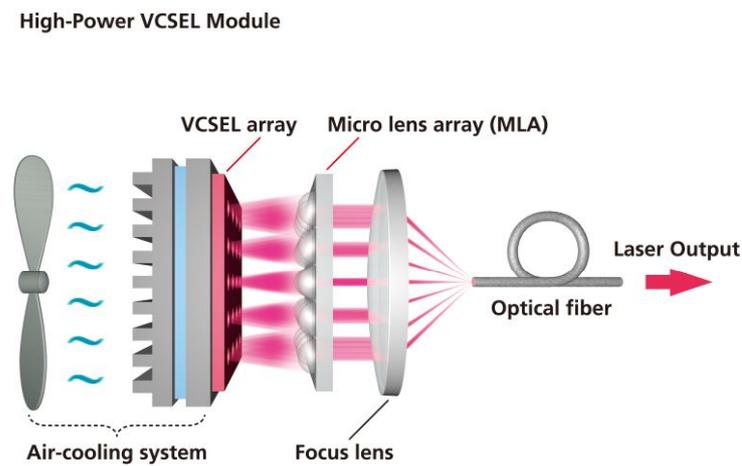
1. VIN = 5V Pin *Power Supply* sumber tegangan positif sensor.
2. GND = Pin *Ground /0V Power Supply* sumber tegangan negatif sensor.
3. SCL & SDA = Pin SCL dan SDA adalah pin I2C atau IIC. I2C adalah singkatan dari *Inter Integrated Circuits*. SCL adalah singkatan dari *Serial Clock*. SDA adalah singkatan dari *Serial Data*.
4. GPIO1 = GPIO (*General Purpose, Input, Output*) digunakan untuk mendesain Arduino agar bisa membaca sensor VL53L0X.
5. XSHUT = *XShutdown* digunakan untuk mengatur ulang I/O.



Gambar 2.14 Pin pada sensor VL53L0X (Maria and Susianti 2019)

2.7 Laser VCSEL

Nama lengkap VCSEL adalah *Vertical Cavity Surface Emitting Laser* (VCSEL)(Gambar 2.15), yang disebut laser pemancar permukaan. Ini adalah salah satu sumber cahaya yang digunakan dalam komunikasi serat optik.



Gambar 2.15 Bagian-bagian laser VCSEL(Amaral et al. 2013).

Laser memancarkan permukaan rongga vertikal (VCSEL) adalah jenis laser baru yang memancarkan cahaya dari permukaan vertikal. Struktur yang berbeda dari laser edge-emitting konvensional memiliki banyak keuntungan yaitu sudut divergensi yang kecil dan simetri melingkar dari distribusi medan jauh dan dekat sangat meningkatkan efisiensi kopling dengan serat tanpa sistem pembentukan balok yang rumit dan mahal. Terbukti bahwa efisiensi kopling dengan serat multimode bisa lebih dari 90%; panjang rongga sangat pendek, yang mengarah ke jarak mode longitudinal yang besar, yang dapat mewujudkan operasi mode longitudinal tunggal dalam kisaran suhu yang luas, dan frekuensi modulasi dinamis yang tinggi; Small membuat faktor emisi spontannya beberapa kali lipat lebih tinggi daripada laser biasa, yang menyebabkan banyak sifat fisik ditingkatkan, hal tersebut dapat diuji dalam chip yang sangat mengurangi biaya pengembangan(Amaral et al. 2013).

2.8 LCD (Liquid Crystal Display)

LCD adalah lapisan dari campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan seven segment dan lapisan elektroda pada kaca belakang (Gambar 2.16). Ketika elektroda diaktifkan dengan medan listrik (tegangan). Lapisan *sandwich* memiliki polarizer cahaya vertikal depan dan polarizer cahaya horizontal belakang yang diikuti dengan lapisan reflektor. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan segmen yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan (Saghoa, Somppie, and Tulung 2018).



Gambar 2.16 LCD (Liquid Crystal Display) (Fitriandi 2016).

2.9 Pengertian Perbandingan atau *Ratios*

Perbandingan adalah ukuran yang digunakan untuk membandingkan suatu nilai terhadap nilai lainnya dengan satuan sejenis. Setiap nilai yang dibandingkan harus mempunyai satuan yang sama, umpamanya satuan panjang, berat, dan waktu. Dalam bahasa Inggris, perbandingan juga disebut dengan "*ratios*".

Dalam penerapannya, perbandingan digunakan sebagai bahasa di lingkungan sehari-hari. Perbandingan sama artinya dengan menjadikan satu.

$$\text{Hasil } X = \frac{\text{Perbandingan } X}{\text{Total Perbandingan}} \times \text{Total Hasil} \dots \dots \dots (2.3)$$

1. Perbandingan Senilai

Perbandingan senilai adalah jenis perbandingan dua atau lebih satuan, saat salah satu nilai meningkat maka nilai lainnya ikut meningkat begitu juga saat salah satu nilai menurun maka nilai lainnya ikut menurun. Rumus perbandingan senilai adalah sebagai berikut:

Ket.

a_1 = Ukuran Pesawat Skala Asli

b_1 = Ukuran Pesawat Sketsa

a_2 = Ukuran VDGS Asli

b_2 = Ukuran VDGS Sketsa

$$a_1 : b_1 = a_2 : b_2 \dots \dots \dots (2.4)$$

atau

$$\frac{a_1}{b_1} = \frac{a_2}{b_2} \dots \dots \dots (2.5)$$

Jika menggunakan konsep pembagi maka:

$$a_1 \times b_2 = b_1 \times a_2 \dots \dots \dots (2.6)$$

Rumus menghitung perbandingan senilai dapat dirumuskan sebagai berikut:

$$a_1 = \frac{b_1 \times a_2}{b_2}, \dots \dots \dots (2.7)$$

$$b_1 = \frac{a_1 \times b_2}{a_2}, \quad a_2 = \frac{a_1 \times b_2}{b_1}, \quad b_2 = \frac{b_1 \times a_2}{a_1}$$

2. Perbandingan Berbalik Nilai

Perbandingan berbalik nilai adalah jenis perbandingan dua atau lebih satuan, saat salah satu nilai meningkat maka nilai lainnya menurun begitu juga saat salah satu nilai menurun maka nilai lainnya meningkat.

$$a_1 : b_2 = a_2 : b_1 \dots \dots \dots (2.8)$$

atau

$$\frac{a_1}{b_2} = \frac{a_2}{b_1} \dots \dots \dots (2.9)$$

Jika menggunakan konsep pembagi maka:

$$a_1 \times b_1 = b_2 \times a_2 \dots \dots \dots (2.10)$$

Rumus menghitung perbandingan berbalik nilai dapat dirumuskan sebagai berikut:

$$a_1 = \frac{b_2 \times a_2}{b_1}, \dots \dots \dots (2.11)$$

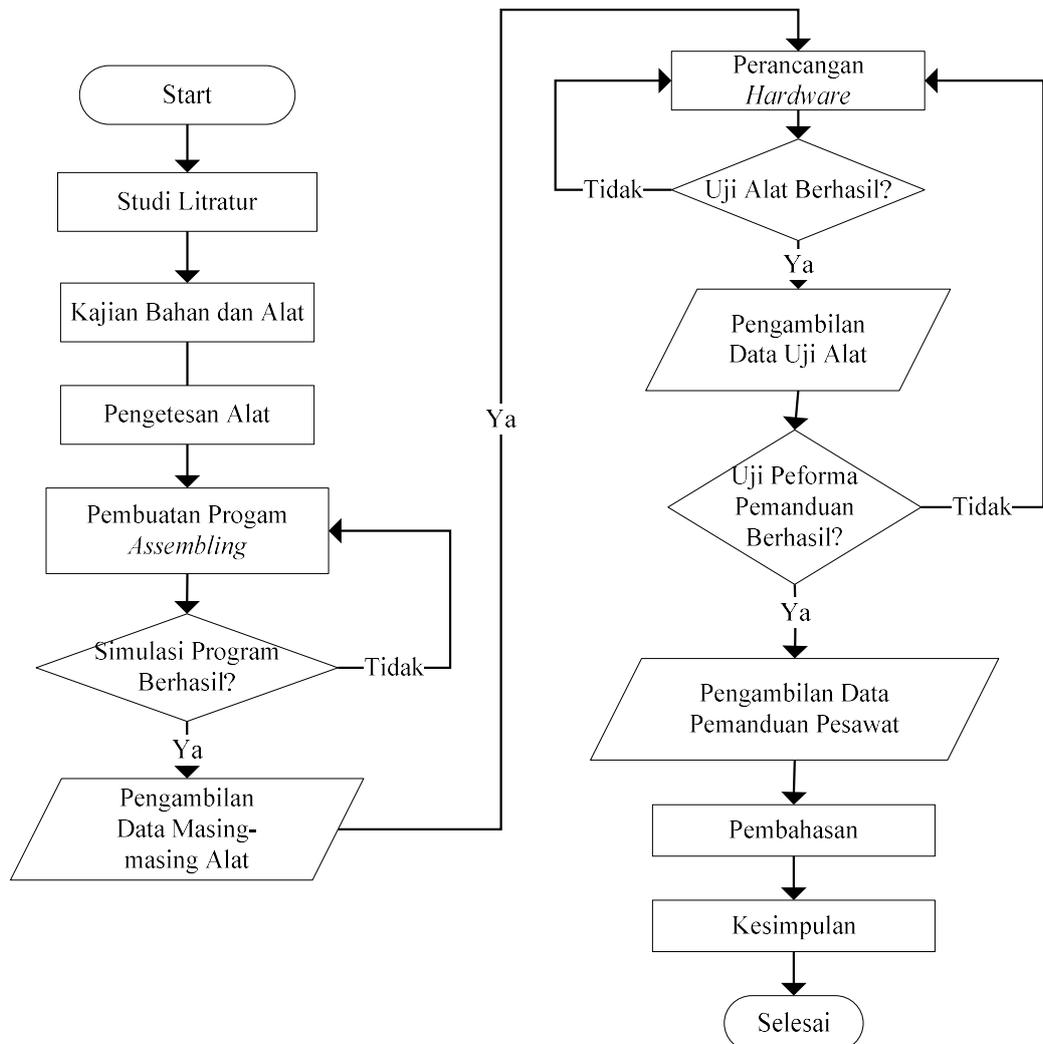
$$b_1 = \frac{b_2 \times a_2}{a_1}, \quad a_2 = \frac{a_1 \times b_1}{b_2}, \quad b_2 = \frac{a_1 \times b_1}{a_2}$$

BAB III

METODE PENELITIAN

3.1 Flowchart Penelitian

Ada beberapa tahapan dalam perancangan dan pembuatan Pemodelan Aircraft Visual Docking Guidance System dengan berbasis Arduino Uno yang dijelaskan dalam Gambar 3.1. Tahapan tersebut harus dilakukan secara berurutan agar diperoleh hasil yang sesuai.



Gambar 3.1 *Flowchart* Penelitian

1. Tahapan pertama adalah memulai penelitian.
2. Tahapan kedua adalah mengumpulkan pengumpulan dasar teori yang berkaitan dengan alat yang dibuat. Hal ini dilakukan untuk mempermudah dalam perancangan alat.
3. Tahapan ketiga adalah mengumpulkan data-data dan bahan-bahan yang dibutuhkan dalam pembuatan alat.
4. Tahapan keempat adalah mencoba menjalankan masing-masing alat yang telah di siapkan beserta programnya.
5. Tahapan kelima adalah merancang program pada alat yang sudah dibuat.
6. Tahapan keenam adalah mensimulasi program yang telah dibuat dan melacak letak permasalahannya.
7. Tahapan ketujuh adalah pengambilan data hasil dari pengujian masing-masing alat dan tingkat keberhasilannya.
8. Tahapan kedelapan adalah merancang alat dari bahan-bahan yang sudah dikumpulkan.
9. Tahapan kesembilan adalah pengujian program dan alat, pengujian alat ini dilakukan untuk mengetahui adanya kesalahan dan kekurangan pada alat yang dibuat.
10. Tahapan ke sepuluh adalah pengambilan data uji performa dari keseluruhan alat.
11. Tahapan ke sebelas adalah pengujian alat terhadap proses pemanduan parkir pesawat.

12. Tahapan kedubelas adalah pengambilan data uji peforma pemanduan parkir pesawat dengan tingkat keberhasilannya.
13. Tahapan ketigabelas adalah pembahasan, membahas masalah dari kekurangan yang ada pada alat.
14. Tahapan keempatbelas adalah kesimpulan, menyimpulkan keseluruhan alat.
15. Tahapan kelimabelas adalah selesai penelitian.

3.2 Lokasi Penelitian

Kegiatan penelitian ini akan dilaksanakan di lokasi studi yaitu di Laboratorium Teknik Elektro Universitas Siliwangi, Jalan Siliwangi No.24, Kota Tasikmalaya.

3.3 Identifikasi Kebutuhan

Bahan dan alat yang dibutuhkan meliputi:

1. Arduino Mega 2560
2. LCD (Liquid Crystal Display)
3. Sensor VL53L0X
4. Push Button
5. Kabel, dll.

3.4 Perancangan dan Pembuatan

Untuk mempermudah dalam memahami model sistem maka dibuat flowchart sistem, dan diagram blok sistem.

3.4.1 Perancangan Prosedur Kerja Sistem

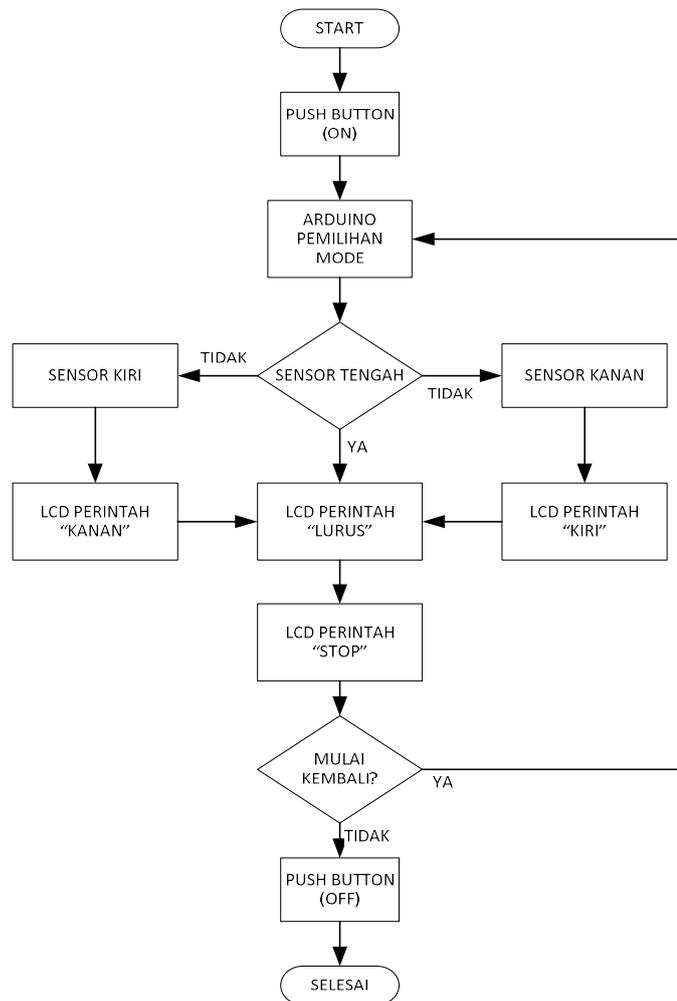
Perancangan Visual Docking Guidance System ini diharapkan mampu mengatasi beberapa kondisi yang di kutip pada Tabel 3.1.

Tabel 3.1 Kondisi Aktivitas yang Diharapkan

	VDGS bekerja jika:	Aktivitas
Kondisi Normal	1. Mendapatkan Arus	Sensor akan membantu sang pilot untuk parkir hingga berhenti sesuai standar landasan <i>Apron</i>
	2. Sensor mendeteksi badan pesawat	
	3. Operator memasukkan kode jenis pesawat dengan benar	
	4. LCD bekerja dan bisa dilihat oleh pilot	
	VDGS btidak bekerja jika:	
Kondisi Gangguan	1. Tidak ada daya	Sistem VDGS tidak akan bisa berjalan dengan baik jika salah satu faktor tersebut ada, dan pilot tidak akan bisa memarkirkan pesawatnya dengan benar.
	2. sensor tidak bekerja	
	3. operator salah memasukkan kode pesawat	
	4. Adanya objek yang menghalangi sensor	
	5. LCD tidak terlihat oleh pilot	

3.4.2 Flowchart Sistem

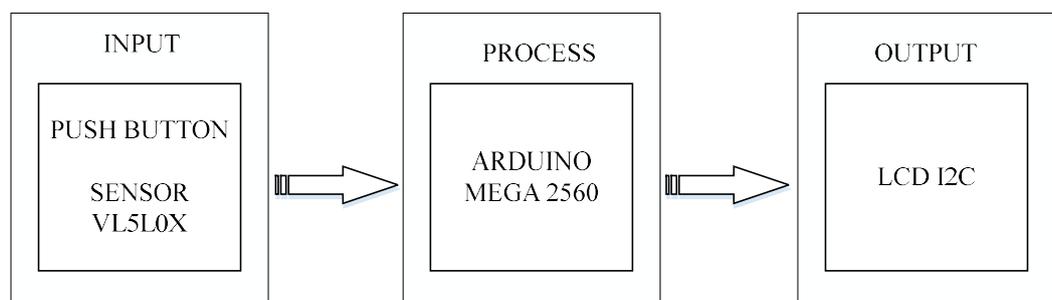
Sistem yang di rancang adalah sebuah sistem yang dapat dioperasikan secara otomatis. Untuk Mempermudah dalam memahami cara kerja dari sistem yang dibuat maka dibuat *flowchart* pada Gambar 3.2.



Gambar 3.2 *Flowchat* Sistem

3.4.3 Blok Diagram

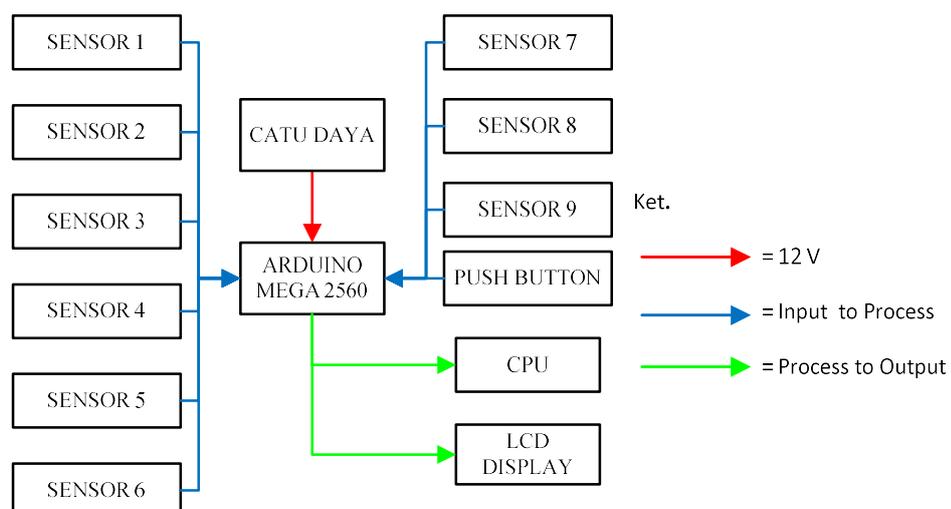
Rancangan sistem yang dibuat digambarkan pada Gambar 3.3 dalam Rancangan Diagram Blok Sistem.



Gambar 3.3 Diagram Blok Sistem

1. *Input*. Pada Gambar 3.3 bagian *input* terdapat sensor VL53L0X, dan *Push Button* yang terpasang pada komponen mekanika. *Push Button* berfungsi untuk memberi pengaturan kepada *Arduino* untuk mengatur sitem kerja sistem VDGS, dan Sensor VL53L0X disini berfungsi untuk mendeteksi jarak dari moncong pesawat terbang agar bisa sejajar dengan lintasan landasan.
2. *Process*. Pada Gambar 3.3 block proses ini, *Arduino Mega 2560* berfungsi sebagai *control* utama untuk menerima sinyal sesuai dengan perintah program yang sudah dibuat.
3. *Output*. Pada Gambar 3.3 bagian *output* terdapat LCD Display. LCD Display bekerja untuk memberikan arahan berupa jenis pesawat yang masuk ke *Apron*, arahan untuk meluruskan badan pesawat dengan tanda panah, dan jarak antara batas pemberhentian kepada pilotnya.

3.4.4 Diagram Blok Kerja Sistem



Gambar 3.4 Diagram Blok Kerja Sistem

3.5 Subjek dan Objek Penelitian

Subjek penelitian yang akan digunakan sebagai penelitian dalam pembuatan tugas akhir ini adalah software Arduino IDE yang terkoneksi pada Arduino Uno (Gambar 3.4). Sedangkan Objek penelitian yang akan digunakan adalah *Visual Docking Guidance System (VDGS)* yang terkoneksi dengan Arduino Uno.

3.6 Metode Pengumpulan Data

Metode-metode yang digunakan dalam pengumpulan data pada tugas akhir ini adalah dengan cara berikut:

1. Studi Literatur

Merupakan metode untuk pengumpulan kajian-kajian teori yang dapat menunjang dalam tugas akhir sehingga dapat menjadi dasar dalam pembuatan tugas akhir.

2. Observasi

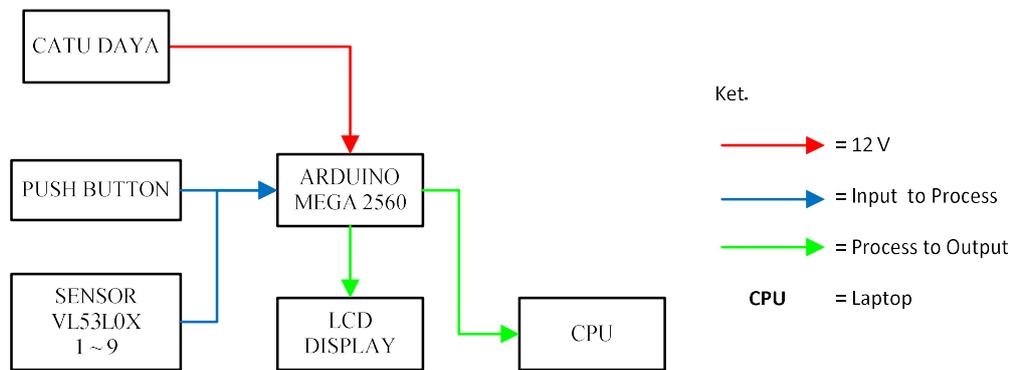
Metode ini adalah metode dengan melakukan pengamatan langsung terhadap objek penelitian atau percobaan. Adapun tujuan penggunaan metode ini adalah untuk membuktikan studi literatur dengan melihat hasil dari suatu pengujian atau percobaan.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Rancang Bangun Sistem

Perancangan sistem dengan membagi setiap bagian kedalam sebuah arsitektur sistem sesuai dengan fungsinya masing-masing. Gambar 4.1 ini merupakan gambar arsitektur sistem dari rancang bangun *Visual Docking Guidance System*.



Gambar 4.1 Arsitektur Sistem

Gambar 4.1 menjelaskan arsitektur dari Sistem VDGS. Arsitektur sistem ini terdiri atas beberapa bagian yaitu:

1. Catu Daya
2. Arduino Mega 2560
3. Sensor Pengukur Jarak
4. Layar Display
5. Push Button
6. Laptop Pengguna

4.1.1 Rancang Bangun Hardware Arduino

Ada beberapa bagian dalam rancang bangun *Hardware Arduino* yaitu, perancangan panel dan *wiring diagram*.

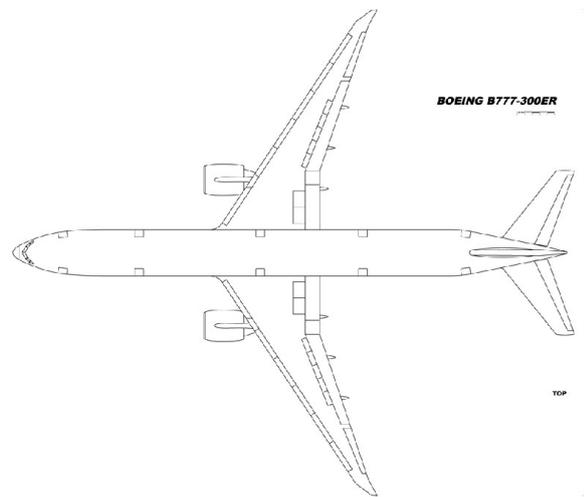
4.1.1.1 Pengukuran Perbandingan Skala Pesawat

Sebelum merancang sebuah panel, diperlukan perhitungan dalam beberapa aspek yaitu sebagai berikut. Pengukuran ini bertujuan untuk membuat sebuah pemodeling agar dapat mengetahui perbandingan ukuran yang asli dengan pemodelingnya.

Data yg di butuhkan adalah Moncong pesawatnya saja karena alat yang di buat hanya membutuhkan bagian moncong pesawatnya untuk di deteksi.

Data pesawat:

1. Boeing 777 – 300 ER



Gambar 4.2 Ilustrasi Gambar Boeing 777-300 ER..

Perbandingan yang dibutuhkan untuk pemodeling alat VDGS ini adalah 1:20 dari ukuran aslinya(Gambar 4.2), berikut adalah perhitungannya.

Lebar kabin: 5.86 m

Lebar badan: 6,2 m

Perhitungan:

1. Menentukan ukuran lebar kabin(Gambar 4.3)

Lebar kabin = 5,86 m = 586 cm

Ket:

A1 = Ukuran Pesawat Asli

A2 = Perbandingan A1

B1 = Ukuran Pesawat Purwarupa

B2 = Perbandingan B1

$$b_1 = \frac{b_2 \times a_1}{a_2} \dots \dots \dots (4.1)$$

$$b_1 = \frac{586 \times 1}{20}$$

$$b_1 = 29,3 \text{ cm}$$

2. Menentukan ukuran lebar badan(Gambar 4.3)

Lebar badan= 6,2 m = 620 cm

Ket:

A1 = Ukuran Pesawat Asli

A2 = Perbandingan A1

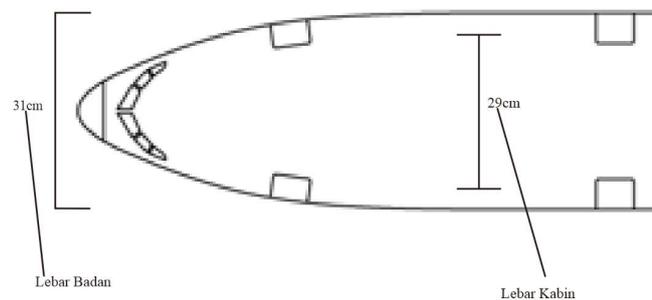
B1 = Ukuran Pesawat Purwarupa

B2 = Perbandingan B1

$$b_1 = \frac{b_2 \times a_1}{a_2} \dots \dots \dots (4.2)$$

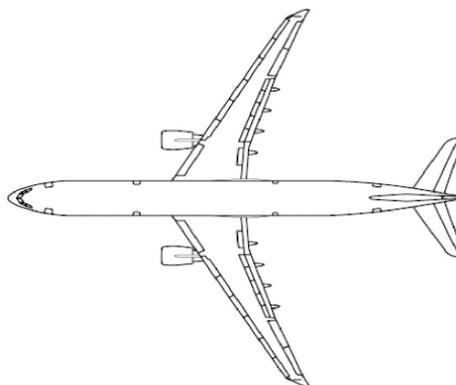
$$b_1 = \frac{620 \times 1}{20}$$

$$b_1 = 31 \text{ cm}$$



Gambar 4.3 Sketsa Ukuran Boeing 777-300 ER

2. Airbus A330 – 300



Gambar 4.4 Ilustrasi Gambar Airbus A330

Perbandingan yang dibutuhkan adalah 1:20 dari ukuran aslinya, berikut adalah perhitungannya(Gambar 4.4).

Lebar kabin: 5,26 m

Lebar badan: 5,64 m

Perhitungan:

1. Menentukan ukuran lebar kabin (Gambar 4.5).

Lebar kabin= 5,26 m = 526 cm

Ket:

A1 = Ukuran Pesawat Asli

A2 = Perbandingan A1

B1 = Ukuran Pesawat Purwarupa

B2 = Perbandingan B1

$$b_1 = \frac{b_2 \times a_1}{a_2} \dots \dots \dots (4.3)$$

$$b_1 = \frac{526 \times 1}{20}$$

$$b_1 = 26,3 \text{ cm}$$

2. Menentukan ukuran lebar badan (Gambar 4.5)

Lebar badan= 5,64 m = 564 cm

Ket:

A1 = Ukuran Pesawat Asli

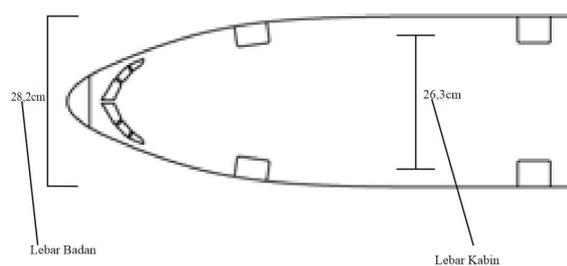
A2 = Perbandingan A1

B1 = Ukuran Pesawat Purwarupa

B2 = Perbandingan B1

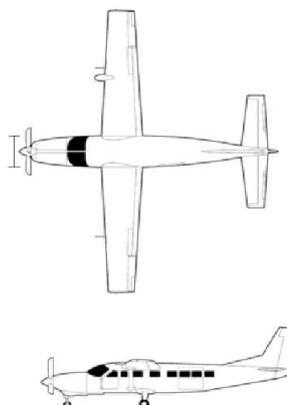
$$b_1 = \frac{b_2 \times a_1}{a_2} \dots \dots \dots (4.4)$$

$$b_1 = \frac{564 \times 1}{20}, b_1 = 28,2 \text{ cm}$$



Gambar 4.5 Ilustrasi Ukuran Airbus A330-300.

3. Cessna 208 Caravan



Gambar 4.6 Ilustrasi Gambar Pesawat Cessna 208 Caravan.

Lebar kabin: 1,37 m

Tinggi kabin: 1,63 m

Perhitungan:

Perbandingan yang dibutuhkan adalah 1:20 dari ukuran aslinya, berikut adalah perhitungannya (Gambar 4.6).

1. Menentukan ukuran lebar kabin

Lebar kabin= 1,37 m = 137 cm

Ket:

A1 = Ukuran Pesawat Asli

A2 = Perbandingan A1

B1 = Ukuran Pesawat Purwarupa

B2 = Perbandingan B1

$$b_1 = \frac{b_2 \times a_1}{a_2} \dots \dots \dots (4.5)$$

$$b_1 = \frac{137 \times 1}{20}$$

$$b_1 = 6,85 \text{ cm}$$

2. Menentukan ukuran lebar badan

Tinggi kabin= 1,63 m = 163 cm

Ket:

A1 = Ukuran Pesawat Asli

A2 = Perbandingan A1

B1 = Ukuran Pesawat Purwarupa

B2 = Perbandingan B1

$$b_1 = \frac{b_2 \times a_1}{a_2} \dots \dots \dots (4.6)$$

$$b_1 = \frac{163 \times 1}{20}$$

$$b_1 = 8,15 \text{ cm}$$

4. Hasil Perhitungan Perbandingan Pesawat

Maka dengan adanya data pesawat purwarupa menjadi acuan untuk terbentuknya purwarupa dari alat VDGS ini, maka bisa di simpulkan pada tabel berikut.

Tabel 4.1 Hasil Perhitungan Skala pada 3 Jenis Pesawat.

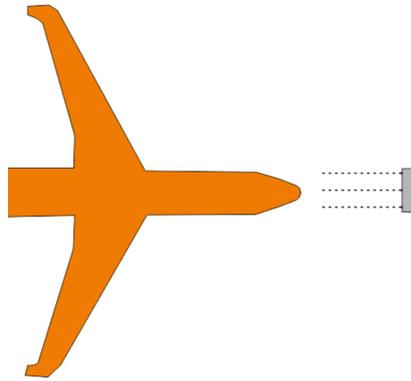
No	Nama Pesawat	Skala Ukuran	
		Purwarupa	Asli
1	Boeing 777 - 300 ER		
	Lebar Kabin	29,3 cm	5,86 m
	Lebar Badan	31 cm	6,2 m
2	Airbus A330 - 300		
	Lebar Kabin	26,3 cm	5,26 m
	Lebar Badan	28,2 cm	5,64 m
3	Cessna 208 Caravan		
	Lebar Kabin	6,8 cm	1,37 m
	Tinggi Kabin	8,15 cm	1,63 m

Tabel 4. 1 menunjukkan perbandingan skala 1:20 pesawat terbang pada ketiga jenis pesawat yang telah di pelajari untuk dijadikan bahan percobaan alat VDGS.

4.1.1.2 Pengukuran Skala Ukuran Panel.

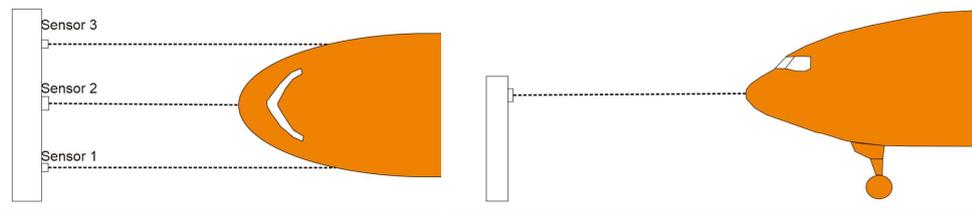
1. Penempatan Sensor

Untuk menentukan posisi titik sensor, kita membutuhkan 3 titik di mana akan di posisikan 1 di titik tengah lingkaran dan 2 lainnya berada di pipi lingkaran, seperti pada Gambar 4.7



Gambar 4.7 Gambar Design Posisi Alat

1. Ukuran untuk Pesawat Mode 1 / Boeing 777 – 300 ER (Gambar 4.8).



Gambar 4.8 Perancangan Penempatan Sensor pada Boeing 777 – 300 ER

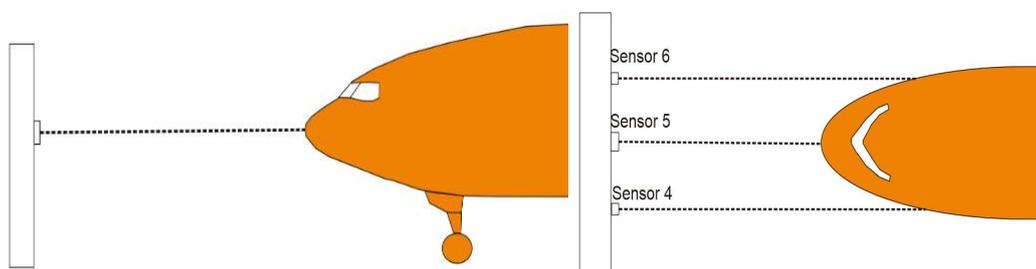
Lebar Pesawat = 31 cm

Titik sensor 1 = Pipi kiri Pesawat

Titik sensor 2 = Hidung Pesawat

Titik sensor 3 = Pipi Kanan Pesawat

2. Ukuran untuk Pesawat Mode 2 / Airbus A330 – 300 (Gambar 4.9).



Gambar 4.9 Perancangan Penempatan Sensor pada Airbus A330-300

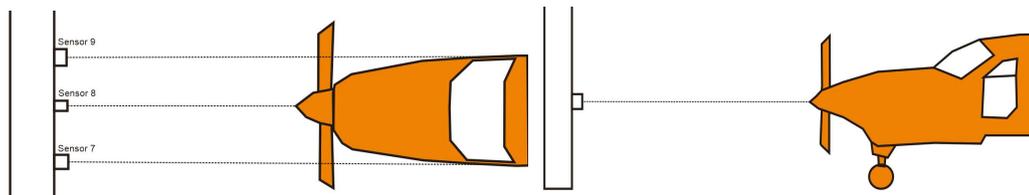
Lebar Pesawat = 28,2 cm

Titik sensor 4 = Pipi kiri Pesawat

Titik sensor 5 = Hidung Pesawat

Titik sensor 6 = Pipi Kanan Pesawat

3. Ukuran untuk Pesawat Mode 3 / Cessna 208 Caravan (Gambar 4.10).



Gambar 4.10 Perancangan Penempatan Sensor pada Cessna 208 Caravan

Lebar Pesawat = 6,86 cm

Titik sensor 7 = Pipi kiri Pesawat

Titik sensor 8 = Hidung Pesawat

Titik sensor 9 = Pipi Kanan Pesawat

2. Ukuran tempat semua Sensor

Berdasarkan ukuran masing-masing jenis pesawat alat akan di buat dengan ukuran perbandingan 1:20 Maka jarak perbandingan alat prototype(Gambar 4.11) yang asli adalah sebagai berikut:

1. Ukuran alat:

Tinggi = 35 cm

Lebar = 34,5 cm

2. Jarak antar Mode:

Mode 1 ke Mode 2 = 8,9 cm

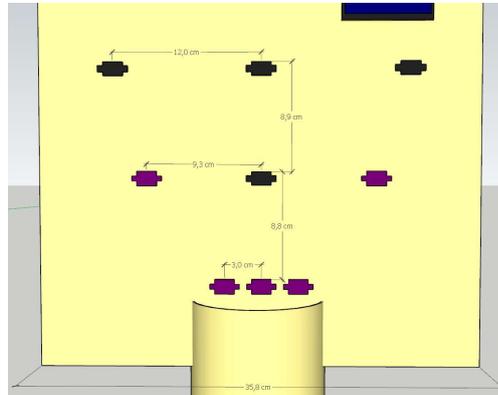
Mode 2 ke Mode 3 = 8,8 cm

3. Jarak antar sensor

Mode 1 =12 cm

Mode 2 = 9,3 cm

Mode 3 = 3 cm



Gambar 4.11 Sketsa Pengukuran Panel.

3. Hasil Perhitungan Perbandingan Skala Panel

Maka setelah perhitungan dengan perbandingan 1:20, data yang dihasilkan adalah data untuk membuat perancangan purwarupa supaya ukurannya cocok dengan ukuran purwarupa pesawatnya. Dengan hasil perhitungan sebelumnya, maka di dapatkan data pada Tabel 4.2.

Tabel 4.2 Hasil Perhitungan Ukuran Purwarupa Alat VDGS.

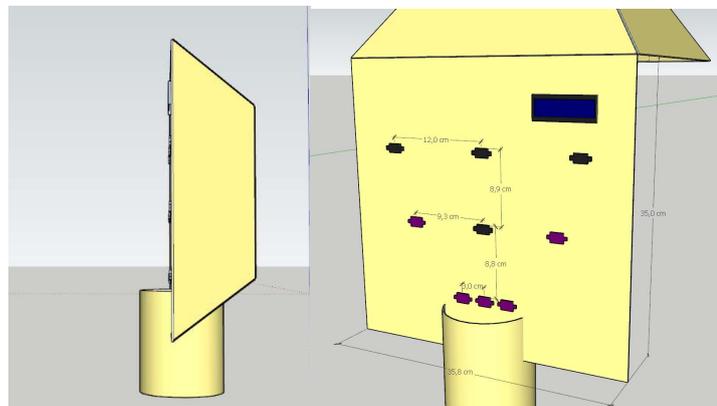
No.	Nama	Skala Ukuran	
		Purwarupa	Asli
1	Ukuran Alat		
	Tinggi	35 cm	7 m
	Lebar	34,5 cm	6,9 m
2	Jarak Antar Sensor		
	Mode 1 - Mode 2	8,9 cm	1,78 m
	Mode 2 - Mode 3	8,8 cm	1,76 m
3	Jarak Sensor		
	Mode 1	12 cm	2,4 m
	Mode 2	9,3 cm	1,86 m
	Mode 3	3 cm	0,6 m

Tabel 4.2 ialah menentukan ukuran perbandingan untuk pembuatan model perancangan alat VDGS yang akan di sandingkan dengan purwarupa model pesawat yang ukuranya sebanding.

4.1.1.3 Perancangan Desain Panel

Setelah perhitungan yang dibuat dan menghasilkan data pada tabel 4.1 dan 4.2, Perancangan panel akan di buat berdasarkan data yang ada dengan skala 1:20 dengan ukuran aslinya seperti pada gambar 4.12. Komponen pada panel Purwarupa VDGS ini diantaranya:

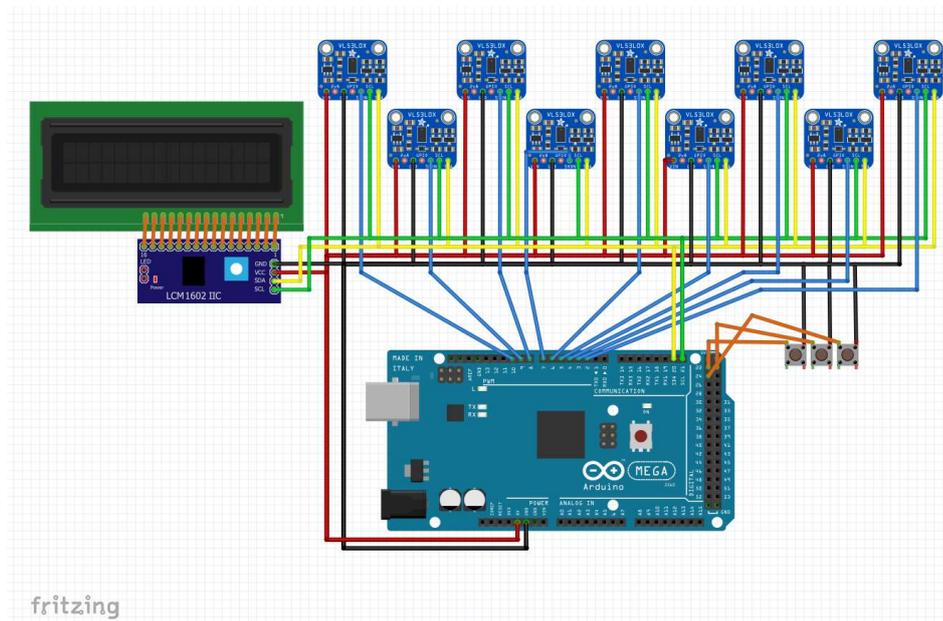
1. Arduino Mega 2620
2. V53L0X Laser Distance Sensor
3. Push Button
4. LCD I2C



Gambar 4.12 Rancangan 3 Dimensi Panel VDGS

1. Perancangan Wiring Diagram

Dibentuklah Perancangan Wiring Diagram rancang bangun sistem VDGS seperti pada Gambar 4.13.



Gambar 4.13 Wiring Diagram Sistem VDGS dengan Arduino Mega

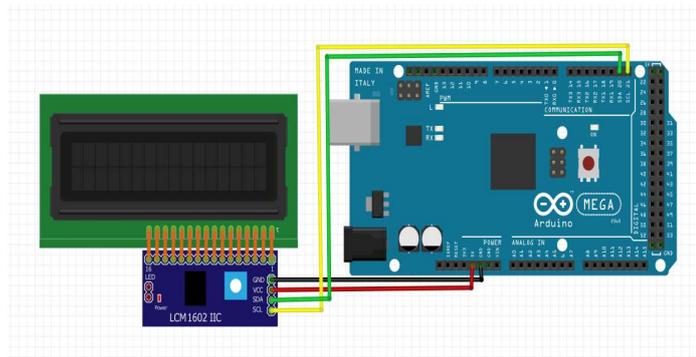
2. Perancangan Hardware Untuk Pemanduan VDGS

Perancangan ini bertujuan untuk merancang sistem kerja pada VDGS untuk memandu sang pilot. LCD I2C digunakan sebagai pemandu parkir yang akan dihubungkan ke Arduino Uno sebagai mikrokontroler. Berikut ini adalah tabel dan gambar *wiring* modul LCD pada Arduino Uno.

Tabel 4.3 Wiring Pin Kaki Modul LCD I2C Pada Arduino

Modul LCD I2C	Arduino Mega
Pin VCC	Arduino (Pin 5V)
Pin SDA	Arduino (Pin 20 SDA)
Pin SCL	Arduino (Pin 21 SCL)
Pin GND	Arduino (Pin GND)

Penggunaan pin-pin pada perancangan alat VDGS bisa di lihat pada Tabel 4.3 yang menunjukkan keterangan penggunaan pada setiap pinnya dan memiliki fungsinya masing-masing.



Gambar 4.14 *Wiring Diagram* LCD I2C (*Display*)

Gambar 4.14 adalah skema *wiring* diagram Modul LCD I2C pada Arduino Uno. Pin VCC sensor dihubungkan pada pin konektor 5V pada pin Arduino Uno yang berfungsi sebagai tegangan masukan modul sebesar 5 V DC. Pin SDA pada modul dihubungkan dengan pin A4 pada Arduino Uno yang berfungsi sebagai komunikasi serial data dari modul. Pin SCL pada modul dihubungkan dengan pin A3 pada Arduino Uno yang berfungsi sebagai komunikasi serial *clock* dari modul. Pin GND pada sensor dihubungkan pada pin GND pada Arduino Uno sebagai ground.

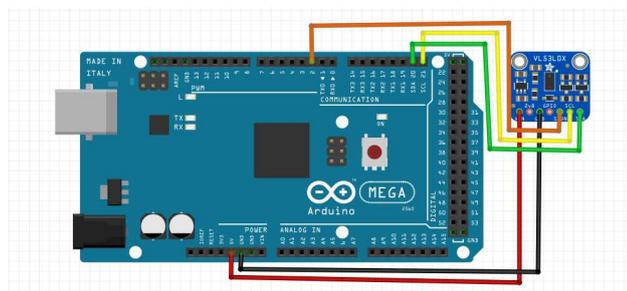
3. Perancangan Sensor Jarak Pada VDGS

Perancangan ini bertujuan untuk merancang sistem untuk mengukur jarak antara pesawat dengan batas parkir *apron*. Sensor yang digunakan adalah sensor jarak yang maksimal yang dapat diukur mencapai 1 Meter. Sensor dihubungkan dengan Arduino Uno sebagai mikrokontrolernya. Berikut ini adalah tabel dan gambar *wiring* sensor jarak pada Arduino Uno.

Tabel 4.4 Wiring Pin Sensor Jarak VL53L0X

Sensor VL53L0X	Arduino Mega
Pin VCC	Arduino (Pin 5V)
Pin XSHUT (Data)	Arduino (Pin 2)
Pin GND	Arduino (Pin GND)
Pin SDA	Arduino (Pin 20 SDA)
Pin SCL	Arduino (Pin 21 SCL)

Tabel 4.4 menunjukkan wiring pin untuk penggunaan sensor jarak VL53L0X yang di sambungkan pada Arduino Mega.

Gambar 4.15 *Wiring Diagram* Sensor Jarak VL53L0X

Gambar 4.15 adalah skema *wiring* diagram sensor jarak VDGS pada Arduino Uno. Pin VCC sensor dihubungkan pada pin konektor 5V pada pin Arduino Uno yang berfungsi sebagai tegangan masukan sensor sebesar 5 V DC. Pin XSHUT sensor dihubungkan dengan pin 2 *digital/PWM* pada Arduino Uno yang berfungsi sebagai pembacaan data dari sensor. Pin GND pada sensor dihubungkan pada pin GND pada Arduino Uno sebagai ground. Pin SDA pada modul dihubungkan dengan pin A4 pada Arduino Uno yang berfungsi sebagai komunikasi serial data dari modul. Pin SCL pada modul dihubungkan dengan pin A3 pada Arduino Uno yang berfungsi sebagai komunikasi serial *clock* dari modul.

4. Pengaturan Mode Jenis Peawat

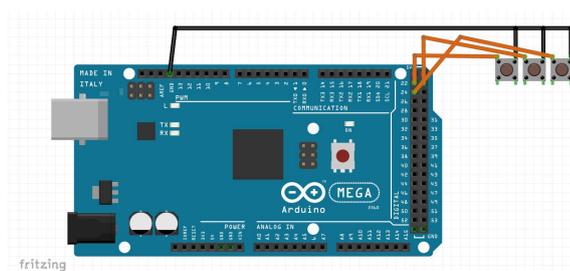
Pengaturan Mode pada alat VDGS akan menggunakan bantuan dari Push Button untuk pemilihan Mode. Pengaturan ini bertujuan untuk merancang sistem

untuk memudahkan sensor untuk membaca dan memandu setiap jenis pesawat yang berbeda ukurannya. Mode ini diatur sedemikian rupa dan menjadi 3 jenis, yaitu mode 1 untuk pesawat berukuran besar dan lebih tinggi, mode 2 untuk pesawat ukuran sedang, dan mode 3 untuk pesawat berukuran kecil. Push Button dihubungkan dengan Arduino Mega 2560 sebagai mikrokontrolernya. Berikut ini adalah tabel dan gambar *wiring Push Button* pada Arduino Mega.

Tabel 4.5 *Wiring Pin Push Button*

Push Button Pin	Arduino Mega
Push Button 1 Pin A	Arduino (GND)
Push Button 1 Pin C	Arduino (Pin 22)
Push Button 2 Pin A	Arduino (GND)
Push Button 2 Pin C	Arduino (Pin 23)
Push Button 3 Pin A	Arduino (GND)
Push Button 3 Pin C	Arduino (Pin 24)

Tabel 4.5 berikut adalah penggunaan pin Push Button yang akan di sambungkan kepada Arduino Mega 2560.



Gambar 4.16 *Wiring Diagram Push Button*

Gambar 4.16 adalah skema *wiring* diagram Push Button pada Arduino Mega. Push Button 1 yang akan diprogram untuk pengaturan Mode 1, Push Button 2 untuk program Mode 2, dan Push Button 3 untuk program Mode 3.

4.1.2 Rancang Bangun Software Arduino

Rancang Bangun *Software Arduino* yaitu dengan membuat kode untuk kontrol otomatis pada sistem *Visual Docking Guidance System*.

Rancang bangun software Arduino yaitu membuat *sketch* untuk sensor pada VDGS menggunakan Arduino IDE yang akan di upload kedalam Arduino Mega 2560.

Program di bawah ini merupakan program dari kerangka utama pada *sketch* Arduino Mega. Program dimulai dengan pemanggilan beberapa library yang dibutuhkan agar dapat terhubung dan diolah dalam memori mikrokontroler.

```
#include <Adafruit_VL53L0X.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

Pada Arduino Mega terhubung dengan LCD I2C, Push Button, dan sensor jarak VL53L0X. Oleh karena itu diperlukan pemanggilan library tersendiri agar dapat terhubung dan diolah dalam memori mikrokontroler. Perintah “`#include`” berfungsi untuk memasukan atau menyertakan file lain dengan program yang kita buat, file yang dimasukan/lampirkan berupa header file.

```
#include <Adafruit_VL53L0X.h>
```

Fungsi include di atas ialah memasukkan *library* untuk sensor VL53L0X, fungsi ini yang nantinya akan berguna sebagai header file utama yang bersangkutan dengan pemrograman terhadap sensor jarak tersebut.

```
#include <LiquidCrystal_I2C.h>
```

Fungsi berikut adalah untuk memasukkan *library* dari LCD, fungsi ini yang nantinya akan menjadi pemandu utama dari sang pilot untuk mengontrol posisi pesawatnya.

```
int inputPin1 = 22; // tombol untuk Mode 1
int inputPin2 = 23; // tombol untuk Mode 2
int inputPin3 = 24; // tombol untuk Mode 3

int kondisiInputPin1 = 0; //kondisi awal tombol 0 = mati
int kondisiInputPin2 = 0; //kondisi awal tombol 0 = mati
int kondisiInputPin3 = 0; //kondisi awal tombol 0 = mati
```

Fungsi diatas adalah untuk memasukkan library untuk Push Button, fungsi ini yang berfungsi sebagai pergantian mode berdasarkan 3 jenis pesawat. Fungsi ini di gunakan oleh operator bantuan agar bisa bekerja.

```
// address we will assign if dual sensor is present
#define LOX1_ADDRESS 0x30
    #define LOX2_ADDRESS 0x31
    #define LOX3_ADDRESS 0x32
    #define LOX4_ADDRESS 0x33
    #define LOX5_ADDRESS 0x34
    #define LOX6_ADDRESS 0x35
    #define LOX7_ADDRESS 0x36
    #define LOX8_ADDRESS 0x37
    #define LOX9_ADDRESS 0x38
// set the pins to shutdown
#define SHT_LOX1 2
#define SHT_LOX2 3
#define SHT_LOX3 4
#define SHT_LOX4 7
#define SHT_LOX5 6
#define SHT_LOX6 5
#define SHT_LOX7 8
#define SHT_LOX8 9
#define SHT_LOX9 10
```

Sedangkan Perintah “`#define`” digunakan untuk mendefinisikan sebuah variable dengan nama variable baru, fungsi mendefinisikan variable ini agar dalam pembuatan program kita lebih mudah mengingat nama variabelnya. Perintah “`#define`” disini digunakan untuk mendefinisikan nama variabel untuk masing-masing sensor jarak dari 1 sampai dengan 9, seperti pada “`#define LOX1_ADDRESS`

0x30” yang memberikan variabel nama LOX1 menggunakan *Address* 0x30 dan “`#define SHT_LOX1 2`” untuk memberikan perintah *shutdown* pada SHT_LOX1 yaitu sensor jarak 1 ke pin 2 Digital pada Arduino Uno.

Library yang dibutuhkan pada sketch1 diantaranya adalah *library* untuk LCD untuk menampilkan karakter pada LCD 16x2, Push Button dan *sensor VL53L0x* sebagai fungsi utama dari sistem VDGS. Setelah pemanggilan *library* yaitu program untuk menginisialisasi nilai dan port yang bertugas sebagai input maupun output pada Arduino diantaranya LCD, Push Button dan sensor.

```
LiquidCrystal_I2C lcd(0x25, 16, 2);
```

Pada bagian ini LCD I2C diberi variabel *address* yaitu 0x25 yang memiliki 16 kolom dan 2 baris. Kemudian untuk Push Button menggunakan inisialisasi sebagai berikut:

Setelah itu, untuk sensor jarak VL53L0X menggunakan inisialisasi sebagai berikut.

```
#define LOX1_ADDRESS 0x30
#define SHT_LOX1 2
Adafruit_VL53L0X lox1 = Adafruit_VL53L0X();
VL53L0X_RangingMeasurementData_t measure1;
...
```

Pada bagian ini, dimulai dengan menginisialisasi sensor jarak dengan nama “lox1” dan menginisiasinya menjadi pin 2 Digital dan di berikan *Address* 0x30, “VL53L0X_RangingMeasurementData_t measure1” di maksud adalah menjadikan sensor sebagai fungsi pengukur jarak.

```
void setup() {
  Serial.begin (115200);
  pinMode(SHT_LOX1, OUTPUT);
  m1=false;
}
```

```

m2=false;
m3=false;
setID();
lcd.begin();
lcd.backlight();
lcd.print("Siap Di Mulai");
}

```

Bagian ini merupakan void setup berfungsi sebagai salah satu fungsi yang hanya satu kali eksekusi ketika awal program berjalan. Dalam void setup ini biasanya berisikan inisialisasi fungsi-fungsi yang digunakan dalam program. pinMode berfungsi untuk mendeklarasikan bahwa "SHT_LOX1" sebagai output. Disini didefinisikan "m1=false" sebagai perintah salah untuk memberikan fungsi "If" sebagai pergantian Mode 1, 2, dan 3. Selain itu pada bagian ini juga berisikan program perintah "lcd.print("Siap Di Mulai");" untuk mengatur posisi kursor untuk menampilkan karakter pada LCD dan perintah lcd.print(); untuk menampilkan karakter pada LCD untuk pertama kali.

```

void loop() {
  if (m1 == false && m3 == false && m2 == false)
    kondisiInputPin1 = digitalRead(inputPin1);
    kondisiInputPin2 = digitalRead(inputPin2);
    kondisiInputPin3 = digitalRead(inputPin3);

  if (kondisiInputPin1 == HIGH)
  {
    m1=true;; // Pindah ke Mode 1
  }
  else if (kondisiInputPin2 == HIGH)
  {
    m2=true; // Mode 2
  }
  else if (kondisiInputPin3 == HIGH)
  {
    m3=true; // Mode 3
  }
  else
  {
    void setID(), HIGH;
    digitalWrite(LCD16()), HIGH);
  }
}
if (m1 == true || m3 == true || m2 == true)
{

```

```

Serial.println("=====");
Serial.println("Status");
Serial.print("M1 : ");
Serial.println(m1);
Serial.print("M2 : ");
Serial.println(m2);
    Serial.print("M3 : ");
Serial.println(m3);
read_sensors();
Serialbos();
LCD16();
delay(1000);
}
}

```

Fungsi ini void loop(){} akan dijalankan setelah fungsi void setup selesai.

Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya (power) dilepaskan. Pada program diatas menjelaskan bahwa program tersebut merupakan program untuk memanggil perintah untuk mengetahui waktu melalui *library* dan menampilkannya dalam bentuk jarak milimeter, dan perintah pengarahannya pada LCD. Fungsi “if (m1 == true || m3 == true || m2 == true)” disini digunakan untuk memberikan pilihan perintah menjadi 3 pilihan “” dan akan di tampilkan pada serial monitor di CPU “Serial.println("Status"); ("M1:"); (m1);”. Fungsi “read_sensors(); Serialbos();LCD16();” disini adalah sebuah set program yang sudah dibuat dan set program tersebut akan di *Loop* secara terus menerus sesuai dengan urutan perintahnya.

```

void setID() {
// all reset
Serial.println("Shutdown pins inited...");
digitalWrite(SHT_LOX1, LOW);
digitalWrite(SHT_LOX1, HIGH);
if(!lox1.begin(LOX1_ADDRESS)) {
    Serial.println(F("Failed to boot first VL53L0X"));
    ...
}
}

```

Program diatas adalah sebuah program yang membuat fungsi baru “void setID” yang berguna untuk meng-set seluruh ID dari sensor jarak yang di gunakan,

fungsi ini bertujuan apabila terjadi kendala pada salah satu sensor maka program akan memberikan peringatan berupa kegagalan perangkat seperti “Failed to boot first VL53L0X”.

```
void read_sensors() {
  if (m1==true)
  {
    lox1.rangingTest(&measure1, false);
    lox2.rangingTest(&measure2, false);
    lox3.rangingTest(&measure3, false);
    if(measure1.RangeStatus != 4) {
      sensor1 = measure1.RangeMilliMeter;
    } else {
      Serial.print("Sensor 1 Out of range");
    }
    if(measure2.RangeStatus != 4) {
      sensor2 = measure2.RangeMilliMeter;
    } else {
      Serial.print("Sensor 2 Out of range");
    }
    if(measure3.RangeStatus != 4) {
      sensor3 = measure3.RangeMilliMeter;
    } else {
      Serial.print("Sensor 3 Out of range");
    }
  }
}
```

Program diatas adalah sebuah program yang membuat fungsi baru untuk di panggil kembali pada perintah *Loop* utama, fungsi nya adalah memberikan progam pada setiap mode “void read_sensors()” untuk membedakan tiap jenis pesawatnya.

```
void LCD16()
{
  if (m1==true)
  {
    if(sensor1 < sensor2 )
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("KIRI");
    }
    if(sensor3 < sensor2 )
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("KANAN");
    }
  }
}
```

```

if (sensor2 < sensor1 && sensor2 < sensor3)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("LURUS");
}
if ( sensor2 <80)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("STOP");
  m1=false;
  delay(2000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("SIAP DI MULAI");
  return;
} }

```

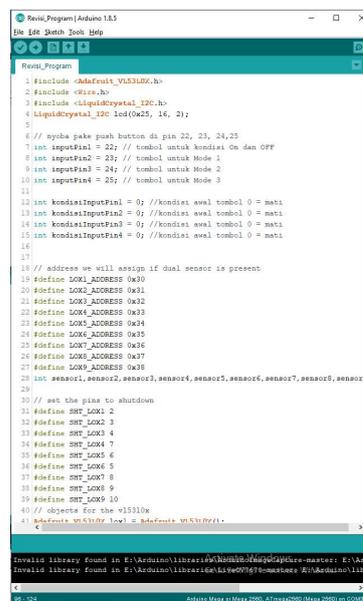
Program diatas menunjukkan program yang berfungsi untuk mengatur perintah untuk mode 1, 2 & 3. Ini bertujuan untuk mengatur sistem kerja utama dari VDGS, program ini dibentuk untuk mengatur jarak, posisi, dan waktu tempat untuk berhentinya pesawat agar berhenti di posisi yang presisi. Fungsi “if (sensor2 < sensor1 && sensor2 < sensor3)” ini adalah pengaturan posisi antara sensor kanan, kiri, dan tengah yang berkerjasama untuk mengatur agar sensor tengah selalu menjadi acuan posisi yang presisi yang akan di tampilkan pada layar LCD I2C. Jika “if (sensor2 <80)” sensor tengah mendeteksi benda pada jarak tertentu, maka akan langsung memberi perintah yang akan di tampilkan pada LCD I2C untuk berhenti di tempat yang tepat. Setelah pesawat sudah berhenti di posisinya maka sistem akan berhenti dan kembali ke awal ketika pemilihan mode pesawatnya.

```

void Serialbos()
{
  Serial.println("=====");
  if ( m1 ==true)
  {
    Serial.print("Sensor 1 : ");
    Serial.print(sensor1);
    Serial.println(" mm");
    ...
  }
}

```

Program di atas adalah sebuah fungsi baru yang berfungsi untuk *monitoring operator*, yaitu sebuah program yang menampilkan status keadaan sensor dan memantau sistem kerja dari VDGS pada Gambar 4.17.



```

1 #include <Adafruit_VDGS.h>
2 #include <Wire.h>
3 #include <liquidcrystal_i2c.h>
4 #include <liquidcrystal_i2c.h>;
5
6 // nyoba pake push button di pin 22, 23, 24,25
7 int inputPin = 22; // tombol untuk kondisi On dan OFF
8 int inputPin = 23; // tombol untuk Mode 1
9 int inputPin = 24; // tombol untuk Mode 2
10 int inputPin = 25; // tombol untuk Mode 3
11
12 int kondisiInputPin = 0; //kondisi awal tombol 0 = mati
13 int kondisiInputPin = 0; //kondisi awal tombol 0 = mati
14 int kondisiInputPin = 0; //kondisi awal tombol 0 = mati
15 int kondisiInputPin = 0; //kondisi awal tombol 0 = mati
16
17
18 // address we will assign if dual sensor is present
19 #define LOCK_ADDRESS 0x30
20 #define LOCK_ADDRESS 0x31
21 #define LOCK_ADDRESS 0x32
22 #define LOCK_ADDRESS 0x33
23 #define LOCK_ADDRESS 0x34
24 #define LOCK_ADDRESS 0x35
25 #define LOCK_ADDRESS 0x36
26 #define LOCK_ADDRESS 0x37
27 #define LOCK_ADDRESS 0x38
28 int sensor1,sensor2,sensor3,sensor4,sensor5,sensor6,sensor7,sensor8,sensor9
29
30 // set the pins to shutdown
31 #define SHT_LOCK1 2
32 #define SHT_LOCK2 3
33 #define SHT_LOCK3 4
34 #define SHT_LOCK4 7
35 #define SHT_LOCK5 6
36 #define SHT_LOCK6 5
37 #define SHT_LOCK7 8
38 #define SHT_LOCK8 9
39 #define SHT_LOCK9 10
40 // objects for the i2c bus
41 #define Wire1 Wire1 = Adafruit_VDGS(I2C);
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Gambar 4.17 Program *sketch* alat VDGS pada Arduino IDE.

4.2 Pengujian Unit

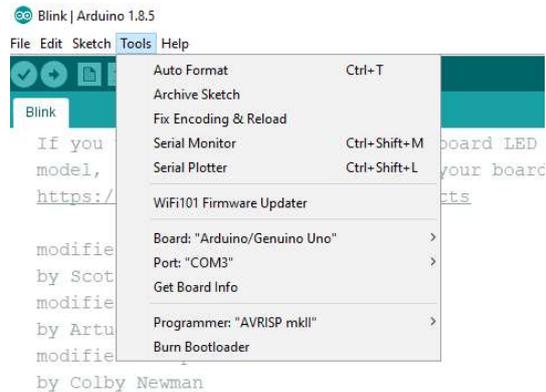
Pengujian unit dilakukan untuk mengetahui apakah komponen yang akan dirancang dapat bekerja dengan baik dan sesuai dengan yang diinginkan.

Dari hasil pengujian program yang dilakukan dengan *software* Arduino IDE, proses *compiling* pada *sketch2* yang di uji secara terpisah tidak ditemukan error pada program sehingga program dapat di jalankan di waktuyang bersamaan dengan sistem utama.

4.2.1 Pengujian Arduino

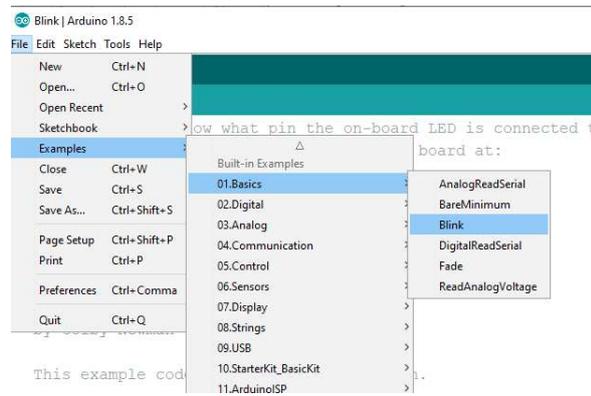
Pengujian Arduino Mega dan Arduino Uno bertujuan untuk mengetahui apakah *board* Arduino dapat bekerja dengan baik atau tidak. Pengujian dilakukan dengan menggunakan program *test blink* dimana program tersebut dibuat untuk

menyalakan LED yang ada pada *board* Arduino Pro Mini. Berikut adalah cara untuk pengujian program *test blink*



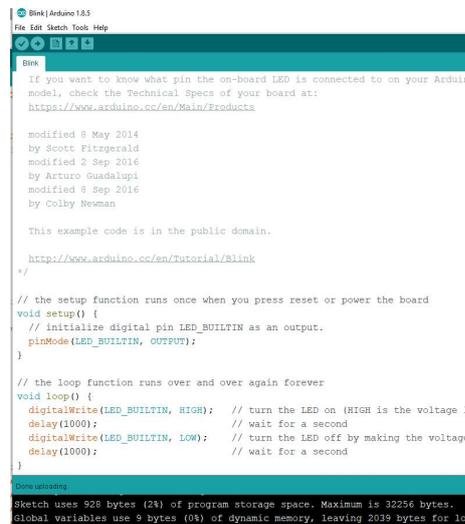
Gambar 4.18 Pengujian Arduino ke perangkat Komputer.

Untuk menghubungkan Arduino dengan PC maka pilih menu *tools*, lalu atur jenis Arduino dan *port* yang digunakan seperti pada gambar 4.18



Gambar 4.19 Percobaan program *blink*.

Pada menu bar software Arduino IDE pilih menu *file>example>basic>blink* terlihat pada gambar 4.19



```

Blink

If you want to know what pin the on-board LED is connected to on your Arduino
model, check the Technical Specs of your board at:
https://www.arduino.cc/en/Main/Products

modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

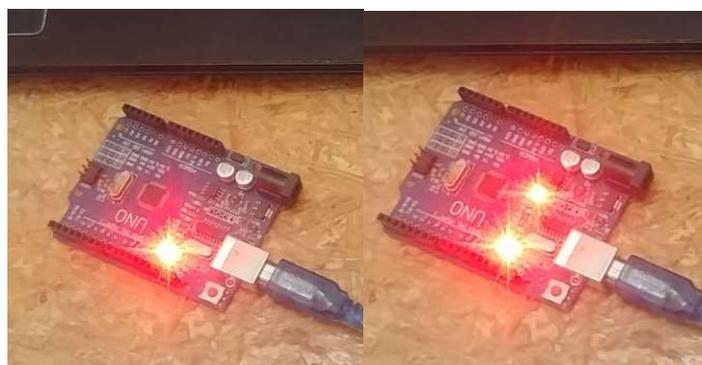
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage L
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage
  delay(1000); // wait for a second
}

```

Sketch uses 528 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables.

Gambar 4.20 *Sketch* untuk pemrograman *Test Blink*.

Gambar 4.20 memperlihatkan tampilan program yang di upload ke board Arduino. Pada bagian void setup() terdapat perintah pinMode (LED_BUILTIN, OUTPUT) dimana pinMode difungsikan sebagai untuk mendeklarasikan suatu pin yang mana pada program ini mendeklarasikan LED_BUILTIN sebagai output. LED_BUILTIN adalah led yang tertanan pada board Arduino Pro Mini. Pada bagian void loop terdapat 2 perintah yaitu digitalWrite (LED_BUILTIN, HIGH) yang bertujuan untuk memberikan nilai high untuk pin LED_BUILTIN dan waktu tunda (delay) selama 1 detik (1000ms). Untuk digitalWrite(LED_BUILTIN, LOW) bertujuan untuk memberikan nilai low untuk pin LED_BUILTIN selama 1 detik.

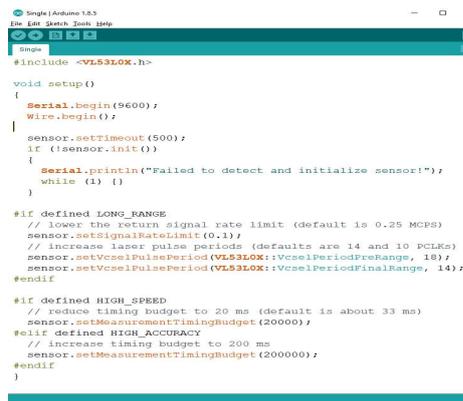


Kondisi Mati (Low) Kondisi Menyala (High)
Gambar 4.21 Menyalakan LED pada Arduino Uno.

Dengan pengujian diatas dapat dilihat bahwa papan arduino bekerja dengan baik. Hal ini dikarenakan board Arduino dapat menjalankan program test blink dan led pada board Arduino dapat menyala (high) dan mati (low) pada Gambar 4.21.

4.2.2 Pengujian Sensor jarak VL53L0X

Pengujian pada sensor jarak bertujuan untuk mengetahui jarak dari sensor dengan benda yang ada di depannya. Pengujian dilakukan dengan program *test single measurement*, dimana program tersebut memberikan perintah pada sensor jarak untuk mengukur satuan jarak antara benda dan ditampilkan pada perangkat lainnya.



```

Single [Arduino 1.8.3]
File Edit Sketch Tools Help

Single
#include <VL53L0X.h>

void setup()
{
  Serial.begin(9600);
  Wire.begin();
}

sensor.setTimeout(500);
if (!sensor.init())
{
  Serial.println("Failed to detect and initialize sensor!");
  while (1) {}
}

#ifdef LONG_RANGE
// lower the return signal rate limit (default is 0.25 MCPS)
sensor.setSignalRateLimit(0.1);
// increase laser pulse periods (defaults are 14 and 10 PCLKs)
sensor.setVcselPulsePeriod(VL53L0X::VcselPeriodPreRange, 18);
sensor.setVcselPulsePeriod(VL53L0X::VcselPeriodFinalRange, 14);
#endif

#ifdef HIGH_SPEED
// reduce timing budget to 20 ms (default is about 33 ms)
sensor.setMeasurementTimingBudget(20000);
#endif
#ifdef HIGH_ACCURACY
// increase timing budget to 200 ms
sensor.setMeasurementTimingBudget(200000);
#endif
}

```

Gambar 4.22 Percobaan *sketch* untuk sensor jarak VL53L0X.

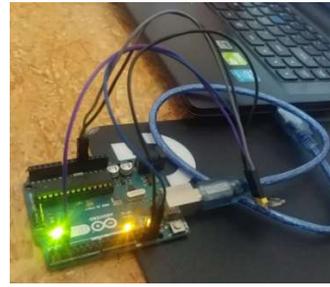
Gambar 4.22 adalah salah satu program untuk sensor jarak VL53L0X yang di upload kepada Arduino. Pada bagian “void setup()” terdapat perintah “Serial.begin(9600);” yang digunakan untuk menampilkan status yang di tampilkan pada “Serial Monitor” pada Arduino. Pada bagian “void loop()” terdapat perintah “Serial.print(sensor.readRangeSingleMillimeters());” perintah ini bertujuan untuk mengetahui data yang di terima oleh sensor VL53L0X yang berupa jarak milimeter dari benda di depannya seperti ada pada gambar 4.23(a).

```

COM12 (Arduino/Genuino Uno)
|
Reading a measurement... Distance (mm): 52
Reading a measurement... Distance (mm): 49
Reading a measurement... Distance (mm): 47
Reading a measurement... Distance (mm): 45
Reading a measurement... Distance (mm): 48
Reading a measurement... Distance (mm): 48
Reading a measurement... Distance (mm): 40
Reading a measurement... Distance (mm): 48
Reading a measurement... Distance (mm): 86
Reading a measurement... Distance (mm): 83
Reading a measurement... Distance (mm): 60

```

(a). Serial Monitor Arduino



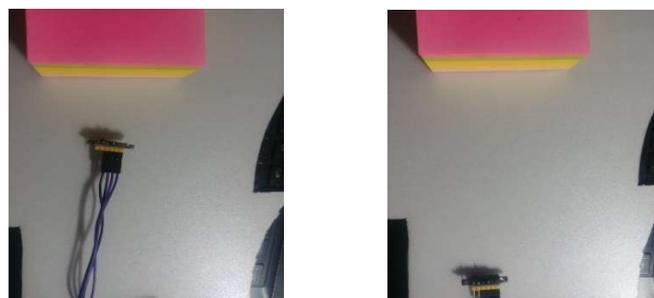
(b). Arduino Uno dan Sensor VL53L0X

Gambar 4.23 Pengukuran Sensor Jarak

Percobaan ini bertujuan untuk mengetahui batasan jarak dari masing-masing sensor seperti pada Gambar 4.23(b), percobaan ini dilakukan dengan mengukur dari batas minimum sensor bisa mendeteksi hingga batas maksimum sensor bisa mendeteksi.

1. Pengukuran Jarak Minimum

Percobaan pada Gambar 4.24 adalah untuk menghitung tingkat terendah dari jarak minimum dari sensor. Percobaan ini melakukan percobaan pada tiap sensor agar bisa menentukan batas pemberhentian yang akurat pada percobaan berikutnya dengan cara mendekatkan benda terhadap sensornya sedekat mungkin. Cara menghitung tingkat erornya adalah dengan melihat data yang di tangkap oleh sensor lalu membandingkan dengan jarak yang aslinya, dari sini kita akan melihat tingkat error dari masing-masing sensor.



Gambar 4.24 Ilustrasi Percobaan Pengukuran Batas Minimum.

Jarak minimum pada Mode 1 untuk sensor dikatakan berhasil ketika sensor bisa mendeteksi pada jarak batas kurang dari 10 cm, jika sensor tidak bisa kurang dari 10 cm maka sistem tidak akan bekerja, sedangkan pada Mode 2 yaitu pada 8 cm, dan Mode 3 membutuhkan batas minimum yang lebih pendek yaitu di di bawah 5 cm berdasarkan ukuran pesawat pada mode ini lebih kecil.

Tabel 4.6 Pengukuran Batas Minimum Sensor
Percobaan Batas Pengukuran Jarak Minimum (cm)

No.	Mode 1			Mode 2			Mode 3		
	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8	Sensor 9
1	10,6	9,9	10,7	4,1	8,1	4,1	3,6	4,2	3,8
2	10,7	9,8	10,6	3,9	7,8	3,9	4	3,8	3,8
3	10,7	9,9	10,7	4,2	7,9	3,8	4	4	4,2
4	10,8	9,8	10,5	4,3	8,1	4,1	4,1	3,9	3,7
5	10,5	10,1	10,7	4	7,9	3,8	3,8	4,1	3,8
6	10,7	10	10,6	4	8	3,7	4,2	4	4
7	10,6	10	10,7	3,9	7,9	4,1	3,8	4,1	3,8
8	10,8	9,9	10,8	4,2	8	3,7	3,7	4,2	3,9
9	10,8	10,1	10,8	3,9	7,8	4	4,1	4	3,8
10	10,8	9,8	10,7	4,3	8,1	4,1	4	3,8	3,9

Dari hasil tabel 4.6 bisa di simpulkan bahwa tingkat keberhasilan masing-masing sensor memiliki perbedaan berdasarkan Mode-nya masing-masing. Dapat di simpulkan bahwa:

1. Pada Mode 1 hasil rata-rata error tertinggi adalah pada sensor 1 dan sensor 3 dengan tingkat kegagalan yaitu 100%, dan pada sensor 2 memiliki error yaitu 20% dari 10 kali percobaan. kegagalan disini dapat di maklumi, dikarenakan sensor 1 dan sensor 3 tidak di gunakan sebagai sensor tengah sebagai fungsi utama dari perintah untuk berhenti di percobaan berikutnya. maka dari itu sensor 1 dan 3 tidak akan bisa di gunakan pada fungsi perintah "STOP".

2. Pada mode 2, tingkat error pada sensor 5 memiliki tingkat error yaitu sebesar 30%, lalu pada sensor 4 dan 6 tidak memiliki tingkat error dikarenakan 2 sensor tersebut memiliki perbedaan jenis sensor. Kedua sensor tersebut menggunakan jenis yang sama seperti pada Mode 3 yang membutuhkan jarak yang lebih pendek.
3. Mode 3 diharuskan menggunakan sensor yang mampu mendeteksi kurang dari 5 cm dikarenakan ukuran pesawat yang lebih kecil dari Mode 1 dan Mode 2, tingkat keberhasilan alat pada Mode ini adalah pada tingkat yang lebih tinggi dari Mode 1 dan Mode 3 yaitu 100% di setiap sensornya.

2. Pengukuran Jarak Maksimum

Pengukuran jarak maksimum bertujuan untuk mengetahui jarak yang diberikan sensor pada pesawat yang akan datang, data pengukuran maksimum sensor. Percobaan ini dilakukan dengan menentukan jarak terjauh dari sensor hingga sensor tidak bisa membaca lagi benda yang dideteksi hingga error akan terjadi. Maka dari itu cara menghitung tingkat errornya adalah dengan menjauhkan objek hingga sensor tidak bisa lagi mendeteksi lalu akan terlihat dari monitor yaitu jarak maksimum dan jarak yang tidak terdeteksi yaitu seperti menunjukkan pada angka "8989". Maka dari itu, hasil batas akhir dari masing-masing sensor adalah pada tabel 4.7 berikut.

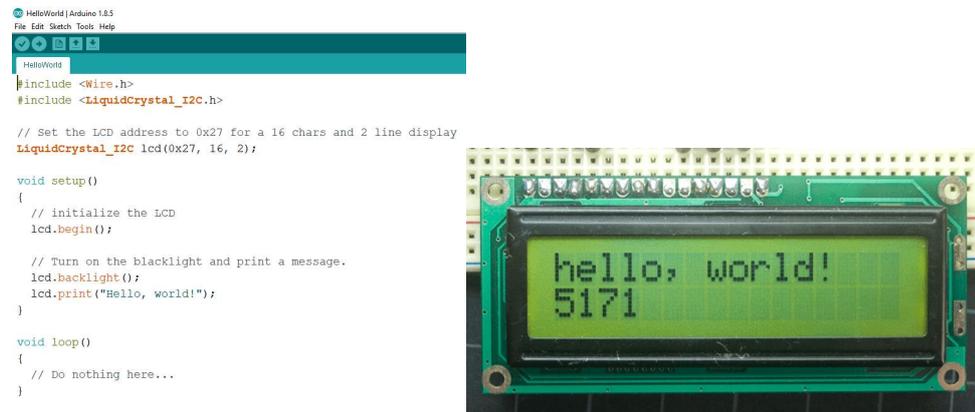
Tabel 4.7 Pengukuran Batas Maksimum Sensor.

Percobaan Batas Pengukuran Jarak Maksimum (cm)									
No.	Mode 1			Mode 2			Mode 3		
	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8	Sensor 9
1	73,5	84,7	85,4	61,1	72,9	53,7	60,9	57,4	64,2
2	76,4	83,8	83,9	66,8	78,1	60,2	58,5	63,1	65
3	72,9	84,1	83,5	56,5	71,1	62,3	56,1	63,8	63,5
4	74,7	81,3	79,8	59,5	72,2	57,1	51,4	65,3	59,9
5	73,1	79,3	80	64,8	69,6	55,6	63,3	59,5	63,9
6	73,8	76,9	80,3	64,5	78,9	62	59,3	53,4	67,6
7	82,7	75,3	78	56,6	70,4	60,1	68,1	68,1	61,5
8	73,4	77,3	88	57,2	73,7	60,6	55,7	59	58
9	75,5	82,6	84,1	64,6	63,6	52,6	57,3	57,5	69,7
10	75,7	78,9	86,3	58,7	73,5	60,4	56,2	59,2	62,9

Dari hasil tabel 4.7 bisa di lihat bahwa rata-rata pengujian masing-masing sensor memiliki batasan tersendiri, nilai terendah di miliki oleh sensor 7 yaitu dengan jarak maksimum rata-nya adalah 58,68, sedangkan nilai tertinggi dimiliki oleh sensor 3 yaitu dengan rata- jarak maksimumnya adalah 82,94. Mode 1 memiliki tingkat jarak lebih tinggi dengan nilai rata-rata 79,93, nilai yang cukup tinggi ini harus di miliki oleh sensor pada Mode 1 dikarenakan kebutuhan jarak yang lebih jauh dibandingkan dengan mode lain karena pesawat pada mode ini lebih besar.

4.2.3 Pengujian Layar LCD I2C

Pengujian sensor LCD I2C dilakukan dengan memberikan program yang akan menampilkan data yang menggunakan krsital cair yang mengubahnya menjadi tampik pada layar LCD. Hasil pengujian dapat dilihat pada Gambar 4.25.

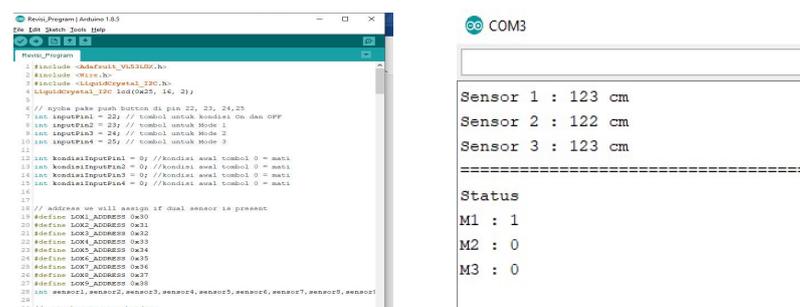


Gambar 4.25 Percobaan LCD dengan kalimat “Hello World!”

Layar LCD akan di gunakan sebagai pemandu pilot pesawat selama proses pemarkiran pesawat, alat ini akan memberikan perintah kepada sang pilot dengan berupa kalimat seperti “LURUS” adalah kalimat yang memerintah sang pilot untuk tetap melaju. Perintah “KANAN” dan “KIRI” yang memerintahkan sang pilot untuk sedikit berbelok ke kanan maupun kiri agar pesawat berada di posisi yang di butuhkan sesuai dengan standar. Perintah “STOP” adalah perintah untuk sang pilot agar berhenti, ini pertanda bahwa pesawat sudah dalam posisi yang benar.

4.2.4 Pengujian Push Button

Pengujian Push Button dilakukan dengan cara melakukan pergantian Mode disaat program VDGS berjalan dan akan di tampilkan pada serial monitor. Berikut adalah contoh dari pengujian alat Push Button pada Gambar 4.26.



Gambar 4.26 Sketch & Percobaan Push Button pada Arduino

Tabel 4.8 Hasil Percobaan pada Tombol Push Button.

No.	Percobaan Push Button		
	1	2	3
1	Ya	Ya	Ya
2	Ya	Ya	Ya
3	Ya	Ya	Ya
4	Ya	Ya	Ya
5	Ya	Ya	Ya
6	Ya	Ya	Ya
7	Ya	Ya	Ya
8	Ya	Ya	Ya
9	Ya	Ya	Ya
10	Ya	Ya	Ya

Percobaan ini dilakukan dengan cara menekan seluruh tombol, dan dilakukan berulang kali sebanyak 10 kali percobaan di setiap tombolnya. Hasil dari Tabel 4.8 adalah 100% keberhasilan di setiap tombolnya dan waktu antar penekanan tombol dan hasil keluarnya cukup memuaskan karna waktu keluarnya adalah dengan waktu kurang lebih 0,1 per detik. Dari hasil data tersebut menunjukkan bahwa tidak ada error yang terjadi pada percobaan ini.

4.2.5 Hasil Pengujian Per Unit

Untuk mempermudah pemrograman dibuat *list* I/O beserta fungsinya, berikut adalah *list* I/O yang digunakan pada sistem *Visual Docking Guidance System* pada Tabel 4.9.

4.3 Pengujian Sistem

Pengujian yang dilakukan adalah melakukan proses *compiling* pada *sketch* yang telah dibuat melalui *software* Arduino IDE.

```

20 #define SHT_LOX2 3
21 #define SHT_LOX3 4
Done uploading.
Sketch uses 23906 bytes (9%) of program storage space. Maximum is 262144 bytes.
Global variables use 5523 bytes (67%) of dynamic memory, leaving 27477 bytes for the global variable table.
Invalid library found in E:\Arduino\libraries\Arduino
Invalid library found in E:\Arduino\libraries\Live
Invalid library found in E:\Arduino\libraries\Arduino
Invalid library found in E:\Arduino\libraries\Live
Exception in thread "AWT-EventQueue-0" java.lang.
31

```

Gambar 4.27 Keberhasilan dalam meng-Upload data.

Dari hasil pengujian program yang dilakukan pada Gambar 4.27 dengan *software* Arduino IDE, proses *compiling* pada *sketch* tidak ditemukan eror pada program sehingga program/*sketch* dapat di *upload* ke Arduino Mega 2560.



Gambar 4.28 Pengujian Alat VDGS.

Gambar 4.28 berikut adalah kondisi ketika *Software* berhasil di jalankan dan alat mulai bekerja.

4.3.1 Pengujian Fungsi Push Button Untuk Mode

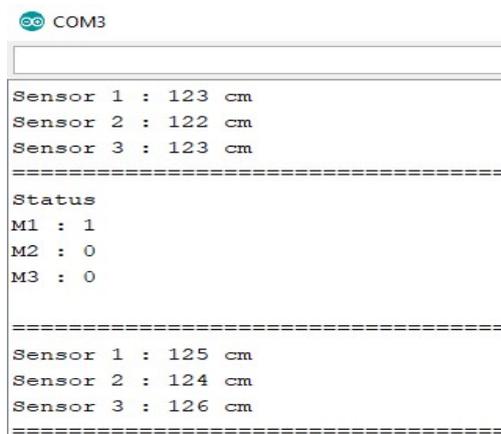
Tabel 4.10 Data Penentuan Mode pada Push Button

Percobaan	Pemilihan Mode		
	Mode 1	Mode 2	Mode 3
1	Ya	Ya	Ya
2	Ya	Ya	Ya
3	Ya	Ya	Ya
4	Ya	Ya	Ya
5	Ya	Ya	Ya
6	Ya	Ya	Ya
7	Ya	Ya	Ya
8	Ya	Ya	Ya
9	Ya	Ya	Ya
10	Ya	Ya	Ya

Tabel 4.10 adalah pengujian terhadap kerja Push Button kepada arduino untuk mengatur Mode/Jenis Pesawat yang akan masuk ke apron. Pengujian ini bertujuan untuk mensimulasikan sistem kerja antar Mode 1, 2, dan 3.

4.3.2 Pengujian Fungsi Sensor Jarak

Tabel pengujian kedua adalah pengujian terhadap kerja alat sensor VL53L0X, sensor ini mengukur jarak yang dibutuhkan pesawat agar tetap lurus dan hingga pesawat berhenti, percobaan ini bisa di liat pada Gambar 4.29 dan Gambar 4.30.

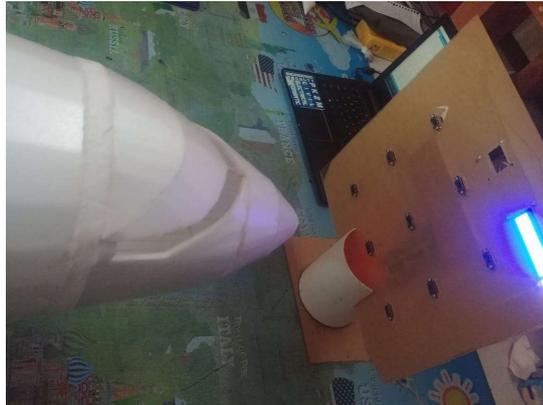


```

COM3
Sensor 1 : 123 cm
Sensor 2 : 122 cm
Sensor 3 : 123 cm
=====
Status
M1 : 1
M2 : 0
M3 : 0
=====
Sensor 1 : 125 cm
Sensor 2 : 124 cm
Sensor 3 : 126 cm
=====

```

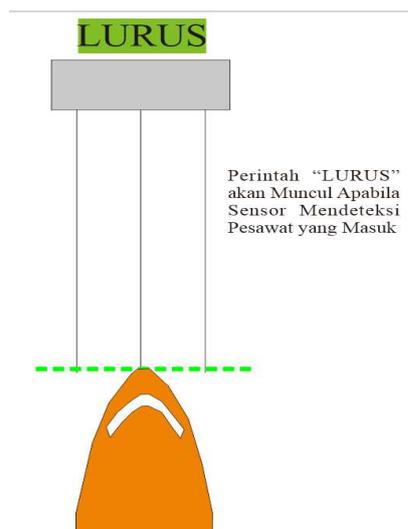
Gambar 4.29 Monitoring sensor jarak dengan program Arduino IDE.



Gambar 4.30 Proses pemarkiran modeling pesawat

4.3.2.1 Pengujian Perintah “LURUS”

Perintah “LURUS” adalah perintah untuk agar sang pilot tetap melaju di jalur yang sama hingga berhenti di jarak yang telah di tentukan, pengujian ini ditentukan oleh sensor jarak saat pertama kali sensor mendeteksi bagian kepala pesawat. Pengujian perintah “LURUS” pada sensor 2, 5 dan 8 yang berada di tengah ditentukan dari pertama kali sensor mendeteksi adanya benda seperti pada Gambar 4.31.



Gambar 4.31 Skema Pengujian Perintah “LURUS”

Tabel 4.11 Pengujian Perintah “LURUS”

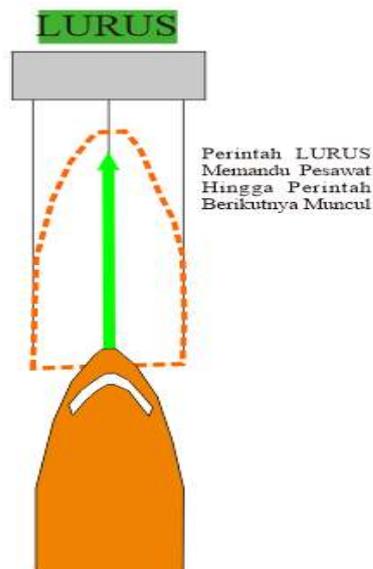
Pengujian Perintah "LURUS" (cm)			
No.	Sensor 2	Sensor 5	Sensor 8
1	71	70	64
2	80	67	63
3	78	66	63
4	76	70	66
5	81	68	64
6	79	67	64
7	77	65	62
8	79	68	50
9	78	72	59
10	78	70	58
11	80	68	65
12	77	65	61
13	78	67	59
14	77	67	56
15	79	67	54
16	78	68	61
17	76	67	57
18	80	69	56
19	80	66	61
20	78	70	60

Percobaan kali ini serupa dengan percobaan batas jarak maksimum sebelumnya, dengan menentukan jarak terjauh dari sensor bisa mendeteksi benda yang datang. Maka dari itu, error di percobaan ini tidak bisa di definisikan karena yang di cari dari hasil percobaan di atas adalah nilai jarak pertemuan antara benda dan sensor yang mendeteksinya.

Dari hasil tabel 4.11, bisa di definisikan bahwa rata-rata nilai jarak pertemuan dari masing-masing sensor tengah yang mendeteksi jarak terjauh dari jangkauan sensor adalah sebagai berikut:

1. Mode 1 menggunakan sensor 2 sebagai titik tengah dari Mode ini yaitu dengan jarak maksimum adalah pada rata-rata 78 cm.
2. Mode 2 yang menggunakan sensor 5 sebagai titik tengah dari Mode ini memiliki rata-rata yang lebih kecil dari sensor 2 yaitu memiliki rata-rata 68 cm, hasil ini tidak mempengaruhi sistem kerja dari alat VGDS ini.
3. Mode 3 menggunakan sensor 7 sebagai titik tengah yang dengan jangkauan ter pendek yaitu pada rata-rata 60 cm. Hasil ini memiliki fungsi tersendiri, dikarenakan tingkat perbandingan ukuran dari pesawat Mode 3 dengan Mode 1 sangat berbeda jauh, maka dari itu hasil diatas tidak mempengaruhi sistem kerja dari alat VGDS ini.

Percobaan berikutnya adalah percobaan pemanduan perintah “LURUS” hingga sampai pada batas pemberhentian dari Apron atau hingga perintah “STOP” muncul pada *display* (Gambar 4.32).



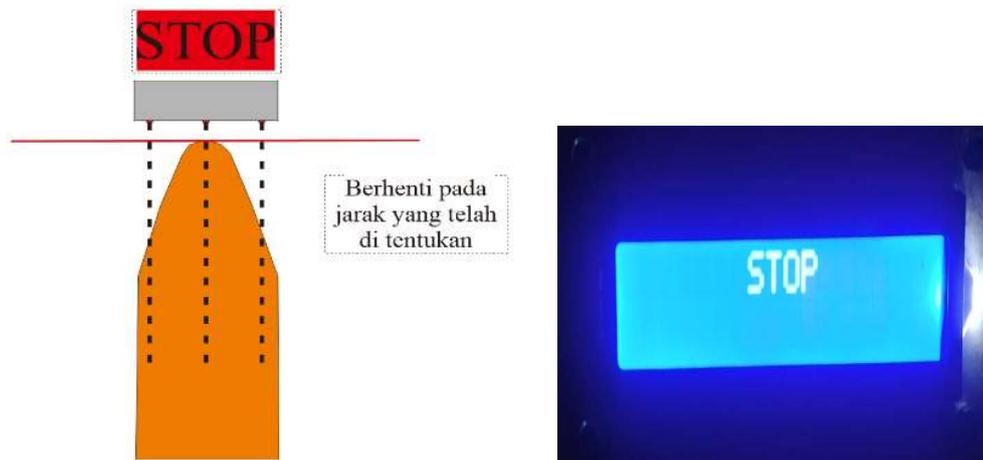
Gambar 4.32 Sketsa perintah “LURUS”

Tabel 4.12 Percobaan Pemanduan Perintah “LURUS”

Percobaan pemanduan			
Ke.	Mode 1	Mode 2	Mode 3
1	Ya	Ya	Ya
2	Ya	Ya	Ya
3	Ya	Ya	Ya
4	Ya	Ya	Ya
5	Ya	Ya	Ya
6	Ya	Ya	Ya
7	Ya	Ya	Ya
8	Ya	Ya	Ya
9	Ya	Ya	Ya
10	Ya	Ya	Ya
11	Ya	Ya	Ya
12	Ya	Ya	Ya
13	Ya	Ya	Ya
14	Ya	Ya	Ya
15	Ya	Ya	Ya
16	Ya	Ya	Ya
17	Ya	Ya	Ya
18	Ya	Ya	Ya
19	Ya	Ya	Ya
20	Ya	Ya	Ya

Dari hasil tabel 4.12, dapat disimpulkan bahwa proses pemanduan pesawat hingga perintah “STOP” muncul berhasil 100% bekerja dengan baik dan tidak ada error yang terjadi selama percobaan.

4.3.2.2 Pengujian Perintah “STOP”



Gambar 4.33 Sketsa Perintah “STOP”

Perintah “STOP” adalah perintah untuk memberhentikan laju pesawat agar pesawat berhenti di tempat yang tepat. Berdasarkan data jenis pesawat purwarupa yang akan di gunakan pada pengujian ini memiliki 3 jenis pesawat dan di bedakan menjadi 3 Mode, Masing-masing mode memiliki batas berhenti yang berbeda-beda, yaitu pada Mode 1 adalah pada 10 cm, Mode 2 adalah 8 cm, dan Mode 3 adalah pada 5 cm. jarak ini ditentukan berdasarkan ukuran masing-masing mode yang membutuhkan ukuran parkir yang sesuai dengan ukuran pesawatnya. Pengujian perintah “STOP” di tentukan dari ketika sensor mendeteksi benda pada jarak tertentu, seperti pada Gambar 4.33.

Tabel 4.13 Pengujian Perintah “STOP”

Pengujian Perintah "STOP" (cm)			
No.	Sensor 2	Sensor 5	Sensor 8
1	10,0	8,0	5,0
2	10,0	8,0	5,0
3	9,8	8,0	5,0
4	9,6	8,0	4,5
5	10,0	8,0	4,5
6	9,9	8,0	4,7
7	10,0	7,9	5,0
8	9,9	8,0	5,0
9	10,0	7,6	5,0
10	9,8	8,0	5,0
11	10,0	8,0	5,0
12	9,7	7,9	4,7
13	9,3	8,0	4,5
14	10,0	8,0	4,2
15	10,0	8,0	5,0
16	9,3	8,0	4,7
17	10,0	8,0	5,0
18	9,0	8,0	4,7
19	9,0	8,0	5,0
20	10,0	7,4	5,0

Terlihat dari hasil tabel 4.13, menentukan bahwa seberapa cepat sensor mendeteksi jarak yang di butuhkan untuk memunculkan perintah “STOP” yaitu dengan ditentukan dari nilai terdekat dari yang di tentukan oleh operator, dan jika nilai tersebut melewati batas yang di tentukan oleh operator maka data tersebut akan di katakana error.

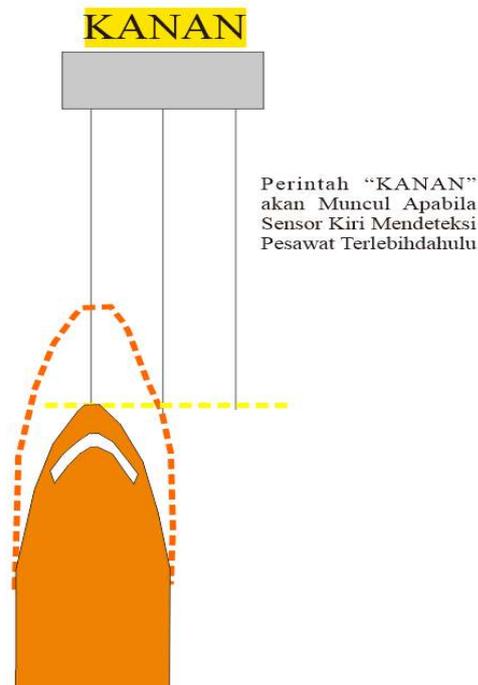
Hasil ini dikarenakan program pada Arduino Mega 2560 yang menggunakan tingkat delay selama 0,1 detik untuk kembali membaca, hal ini bisa di perbaiki dengan cara mengurangi tingkat delay jika ingin hasil yang lebih akurat. Lalu berdasarkan hasil tabel di atas yaitu:

1. Mode 1 yang diatur untuk berhenti pada jarak 10 cm memiliki tingkat akurasi keberhasilan sebesar 50%, hasil ini memiliki error yang sangat besar karna sensor membaca lebih lambat dari kecepatan pesawat yang mengakibatkan keterlambatan pesawat dapat berhenti dengan akurat.
2. Mode 2 yang di atur untuk berhenti pada jarak 8 cm memiliki tingkat akurasi keberhasilan sebesar 80%, hasil ini memiliki tingkat keberhasilan yang lebih baik dari pada Mode 1 dan 3.
3. Mode 3 yang di atur untuk berhenti pada jarak 5 cm memiliki tingkat akurasi keberhasilan sebesar 60%. Mengingat ukuran pesawat di Mode 3 ini terpaut jauh dari Mode 1 dan Mode 2.

Berdasarkan hasil diatas, bisa di tarik kesimpulan bahwa tingkat delay sensor menjadi faktor utama dari tingkat keakuratan perintah “STOP” pada setiap Mode. Jika di bandingkan dengan ukuran asli, jarak batas akhir sangat lah berpengaruh untuk sistem parkir ini mengingat bahwa jarak dari pemberhentian pesawat dengan gedung bandara dapat membahayakan keselamatan orang lain juga.

4.3.2.3 Pengujian Perintah “KANAN”

Pengujian perintah “KANAN” adalah suatu perintah untuk memberitahu pilot untuk menyesuaikan posisi pesawat agar pesawat menjadi lurus. Pengujian ini ditentukan dari pertama kali sensor mendeteksi pesawat yang perbandingan antara jumlah jarak sensor bagian tengah lebih besar daripada sensor bagian kiri, lalu akan muncul perintah untuk berbelok ke “KANAN” seperti pada Gambar 4.34.



Gambar 4.34 Sketsa perintah “KANAN”

Tabel 4.14 Pengujian Batas Jarak Perintah “KANAN”.

Pengujian Perintah "KANAN" (cm)									
No.	Mode 1		Perbandingan	Mode 2		Perbandingan	Mode 3		Perbandingan
	Sensor 2	Sensor 3		Sensor 5	Sensor 4		Sensor 8	Sensor 9	
1	84	66,1	17,9	60	37,6	22,4	53	46,5	6,5
2	Out of Range	67,5	Out of Range	Out of Range	41,7	Out of Range	53	46,6	6,4
3	88	69,9	18,1	61	38,5	22,5	54	47,4	6,6
4	84	65,8	18,2	62	39,4	22,6	Out of Range	49,5	Out of Range
5	84	66,1	17,9	58	35,7	22,3	55	48,4	6,6
6	83	65	18	62	39,6	22,4	50	43	6,5
7	81	63	18	Out of Range	42,1	Out of Range	52	45,4	6,6
8	Out of Range	68	Out of Range	Out of Range	42,1	Out of Range	54	47,4	6,6
9	87	69	18	Out of Range	42	Out of Range	54	47,5	6,5
10	Out of Range	67,9	Out of Range	61	38,6	22,4	54	47,3	6,7
11	84	65,9	18,1	Out of Range	41,8	Out of Range	53	46,6	6,4
12	Out of Range	68	Out of Range	62	39,8	22,2	Out of Range	49,6	Out of Range
13	83	64,8	18,2	60	38	22	52	45,6	6,4
14	80	61,8	18,2	Out of Range	41,3	Out of Range	Out of Range	49,4	Out of Range
15	83	64,9	18,1	61	38,9	22,1	54	47,6	6,4
16	Out of Range	68,1	Out of Range	Out of Range	42,2	Out of Range	53	46,5	6,5
17	Out of Range	68	Out of Range	Out of Range	42,1	Out of Range	52	45,5	6,5
18	83	65	18	59	37,1	21,9	53	46,6	6,4
19	83	65	18	60	37,5	22,5	50	43,4	6,6
20	85	67	18	62	39,6	22,4	55	48,5	6,5

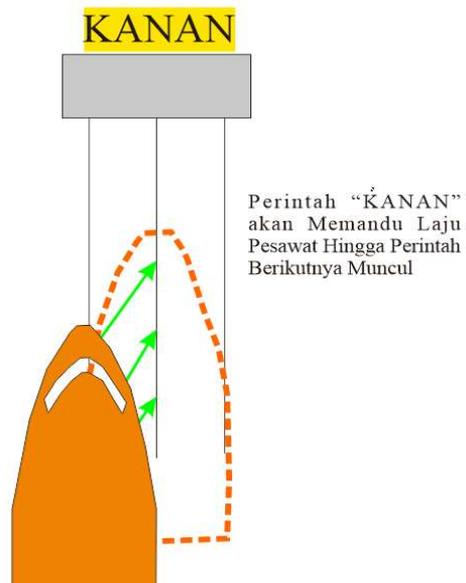
Out of Range pada tabel 4.14 menunjukkan bahwa batas akhir sensor yang tak dapat terdeteksi, maka dari itu nilai perbandingan jaraknya pun juga dikatakan *Out of Range* karena tidak memiliki nilai bandingnya. Dari hasil tabel 4.12, di simpulkan bahwa:

1. Mode 1 yang menggunakan sensor 2 sebagai sensor tengah memiliki tingkat *Out of Range* yang cukup tinggi dengan rata-rata *Out of Range* adalah 40%, pada jarak batas maksimum sensor mendeteksi adalah pada 85 cm.
2. Hal serupa juga terjadi pada Mode 2, yang menggunakan sensor 5 sebagai sensor tengah yang memiliki batas jarak maksimum 63 cm menghasilkan *Out of Range* yaitu sebesar 40% karena sensor 5 tidak terdeteksi lebih dari batas maksimumnya.
3. Mode 3 menggunakan sensor 7 sebagai sensor tengahnya memiliki tingkat *Out of Range* sebesar 15% dengan batas maksimum sensor bisa mendeteksi adalah pada 55 cm.

Dari kesimpulan dari masing-masing mode di atas, sensor kiri mendeteksi lebih dulu sebelum sensor tengahnya mendeteksi objeknya. Hasil ini jika di bandingkan dengan ukuran aslinya, tidak akan berpengaruh besar karena jika sensor kiri sudah mendeteksi maka perintah “KANAN” akan langsung muncul pada *display* dan akan langsung memandu pemarkiran pesawat pada perintah-perintah selanjutnya.

Percobaan berikutnya adalah proses pemanduan pesawat dari pertama perintah “KANAN” muncul pada *display* hingga perintah “LURUS” muncul pada *display* seperti pada Gambar 4.35. Percobaan ini di lakukan dengan perkiraan

kecepatan pesawat pada ukuran aslinya adalah kurang lebih pada ~10 km/jam atau sekitar ~5 cm per detik.



Gambar 4.35 Skema Pemanduan Perintah “KANAN”

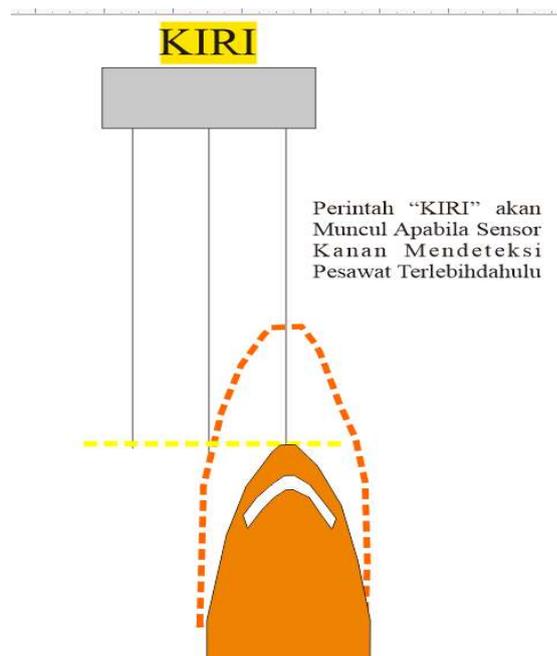
Tabel 4.15 Percobaan Pemanduan Pesawat pada Perintah “KANAN”

Percobaan Pemanduan Kanan			
ke.	Mode 1	Mode 2	Mode 3
1	Ya	Ya	Ya
2	Ya	Ya	Ya
3	Ya	Ya	Ya
4	Ya	Ya	Ya
5	Ya	Ya	Ya
6	Ya	Ya	Ya
7	Ya	Ya	Ya
8	Ya	Ya	Ya
9	Ya	Ya	Ya
10	Ya	Ya	Ya
11	Ya	Ya	Ya
12	Ya	Ya	Ya
13	Ya	Ya	Ya
14	Ya	Ya	Ya
15	Ya	Ya	Ya
16	Ya	Ya	Ya
17	Ya	Ya	Ya
18	Ya	Ya	Ya
19	Ya	Ya	Ya
20	Ya	Ya	Ya

Hasil dari tabel 4.15, menunjukkan bahwa percobaan pemanduan pesawat perintah “KANAN” berjalan dengan sukses tanpa ada hasil *Out of Range*.

4.3.2.4 Pengujian Perintah “KIRI”

Sama halnya dengan perintah “KANAN”, perintah ini merupakan perintah untuk membelokkan laju pesawat agar pesawat tidak keluar dari jalur parkir yang sudah di tentukan. Pengujian ini di tentukan oleh perbandingan apabila sensor bagian tengah lebih besar dari pada sensor bagian kanan dan akan memunculkan perintah “KIRI” pada layar LCD, seperti pada Gambar 4.36.



Gambar 4.36 Sketsa Perintah “KIRI”.

Tabel 4.16 Hasil Pengujian Perintah “KIRI”

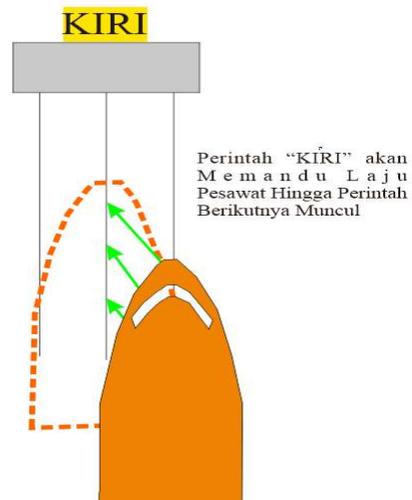
Pengujian Perintah "KIRI" (cm)									
No.	Mode 1		Perbandingan	Mode 2		Perbandingan	Mode 3		Perbandingan
	Sensor 2	Sensor 1		Sensor 5	Sensor 6		Sensor 8	Sensor 7	
1	Out of Range	67	Out of Range	59	36,7	22,3	Out of Range	47,4	Out of Range
2	79	60,5	18,5	61	38,8	22,2	Out of Range	49,7	Out of Range
3	82	64,1	17,9	Out of Range	40,8	Out of Range	47	40,6	6,4
4	Out of Range	69,7	Out of Range	61	38,9	22,1	49	42,5	6,5
5	73	54,8	18,2	60	37,5	22,5	44	37,5	6,5
6	79	61,1	17,9	59	37,1	21,9	Out of Range	47,4	Out of Range
47	77	58,4	18,6	57	35,3	21,7	51	44,6	6,4
8	80	61,5	18,5	59	37,1	21,9	50	43,4	6,6
9	82	63,5	18,5	61	38,4	22,6	51	44,8	6,2
10	89	70,8	18,2	Out of Range	40,9	Out of Range	Out of Range	47,7	Out of Range
11	Out of Range	70,1	Out of Range	62	40,1	21,9	Out of Range	48,5	Out of Range
12	75	56,2	18,8	59	37,1	21,9	Out of Range	51,5	Out of Range
13	79	60,6	18,4	60	37,7	22,3	51	44,4	6,6
14	82	63,2	18,8	57	34,9	22,1	49	42,5	6,5
15	Out of Range	64,7	Out of Range	59	36,8	22,2	Out of Range	50,5	Out of Range
16	Out of Range	67,1	Out of Range	57	35,1	21,9	51	44,3	6,7
17	80	61,1	18,9	62	39,6	22,4	53	46,4	6,6
18	Out of Range	70,2	Out of Range	61	38,9	22,1	Out of Range	48,6	Out of Range
19	Out of Range	64,9	18,1	58	36,1	21,9	51	44,5	6,5
20	82	63,8	18,2	59	37,2	21,8	49	42,3	6,7

Sama seperti pada percobaan perintah “KANAN” sebelumnya. *Out of Range* pada tabel 4.13 menunjukkan bahwa batas akhir sensor yang tak dapat terdeteksi, maka dari itu nilai perbandingan jaraknya pun juga dikatakan *Out of Range* karena tidak memiliki nilai bandingnya. Dari hasil tabel 4.16, bisa di simpulkan bahwa:

1. Mode 1 yang menggunakan sensor 2 sebagai sensor tengah memiliki tingkat *Out of Range* yang cukup tinggi dengan rata-rata *Out of Range* adalah 40%, pada jarak batas maksimum sensor mendeteksi adalah pada 82 cm.
2. Mode 2 yang menggunakan sensor 5 sebagai sensor tengah yang memiliki batas jarak maksimum 62 cm menghasilkan *Out of Range* yaitu sebesar 10% karena sensor 5 tidak terdeteksi lebih dari batas maksimumnya.
3. Mode 3 menggunakan sensor 7 sebagai sensor tengahnya memiliki tingkat *Out of Range* sebesar 40% dengan batas maksimum sensor bisa mendeteksi adalah pada 53 cm.

Dari kesimpulan diatas, sensor kiri yang mendeteksi lebih dahulu sebelum sensor tengahnya mendeteksi objeknya dan akan langsung memberikan perintah “KIRI”.

Percobaan berikutnya adalah percobaan pemanduan perintah “KIRI” hingga muncul perintah “LURUS” pada *display* seperti yang terlihat pada Gambar 4.37. Percobaan ini di lakukan dengan perkiraan kecepatan pesawat pada ukuran aslinya adalah kurang lebih pada ~10 km/jam.



Gambar 4.37 Skema Proses Pemanduan Perintah "KIRI"

Tabel 4.17 Percobaan Pemanduan Perintah "KIRI".

Percobaan Pemanduan Kanan			
ke.	Mode 1	Mode 2	Mode 3
1	Ya	Ya	Ya
2	Ya	Ya	Ya
3	Ya	Ya	Ya
4	Ya	Ya	Ya
5	Ya	Ya	Ya
6	Ya	Ya	Ya
7	Ya	Ya	Ya
8	Ya	Ya	Ya
9	Ya	Ya	Ya
10	Ya	Ya	Ya
11	Ya	Ya	Ya
12	Ya	Ya	Ya
13	Ya	Ya	Ya
14	Ya	Ya	Ya
15	Ya	Ya	Ya
16	Ya	Ya	Ya
17	Ya	Ya	Ya
18	Ya	Ya	Ya
19	Ya	Ya	Ya
20	Ya	Ya	Ya

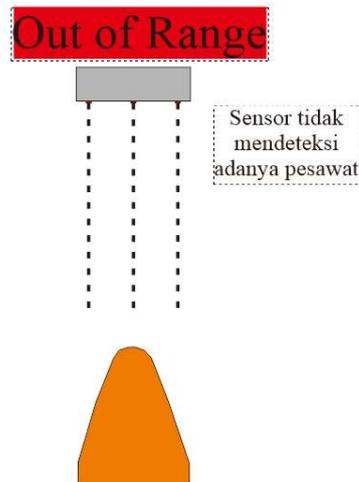
Seperti yang terlihat pada tabel 4.17, proses pemanduan perintah “KIRI” berjalan dengan baik hingga ke proses pemanduan sebelumnya dan tidak memiliki Out of Range hingga percobaan ke 20.

4.3.2.5 Percobaan Kesalahan Sistem(*Error*)

Percobaan ini berfungsi untuk menentukan eror yang terjadi jika ada masalah pada jalur masuk pesawat, kemiringan bisa terjadi kapan saja ketika pesawat masuk ke jalur sensor. Maka dari itu percobaan ini menunjukkan apabila terjadi kesalahan teknis dari pemarkiran parkir pesawat terbang.

1. Percobaan *Out of Range*

Out of Range adalah sebuah kondisi dimana sensor tidak dapat mendeteksi suatu objek di depannya atau terjadi eror pada saat proses pemanduan seperti pada Gambar 4.38. Kondisi *out of range* adalah dimana posisi pesawat terlalu jauh dari jarak jangkauan sensor dan kalimat “*OUT OF RANGE*” pada layar.



Gambar 4.38 Sketsa *Out of Range*.

Tabel 4.18 Percobaan *Out of Range*.

Pengujian <i>Out of Range</i> (cm)			
No.	Mode 1	Mode 2	Mode 3
1	73	72	63
2	76	66	60
3	75	67	60
4	76	72	61
5	80	69	61
6	80	68	63
7	80	67	60
8	79	68	57
9	77	71	58
10	78	71	59
11	81	69	62
12	76	67	63
13	80	68	58
14	80	66	59
15	79	65	58
16	79	67	60
17	77	68	59
18	78	69	58
19	79	69	60
20	80	70	61

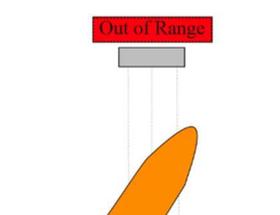
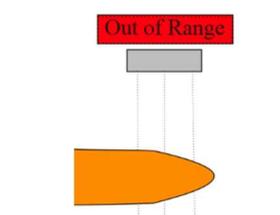
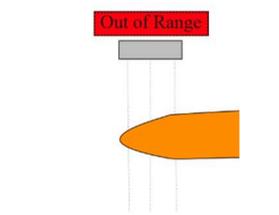
Pengujian Gambar 4.38 menggunakan pengujian yang sama dengan pengujian batas jarak maksimum dari masing-masing sensor, dengan hasil pada Tabel 4.7. Pengukuran Batas Maksimum Sensor pada Tabel 4.18, menunjukkan bahwa rata-rata pengujian masing-masing sensor memiliki batasan tersendiri, nilai terendah di miliki oleh Mode 3 yaitu dengan jarak maksimum rata-nya adalah 58.68, sedangkan nilai tertinggi dimiliki oleh Mode 1 yaitu dengan rata- jarak maksimumnya adalah 82.94, Mode 1 memiliki tingkat jarak lebih tinggi dengan nilai rata-rata 79.93, nilai yang cukup tinggi ini harus di miliki oleh sensor pada

Mode 1 dikarenakan kebutuhan jarak yang lebih jauh dibandingkan dengan mode lain karena pesawat pada mode ini lebih besar.

2. Kemungkinan Kemiringan

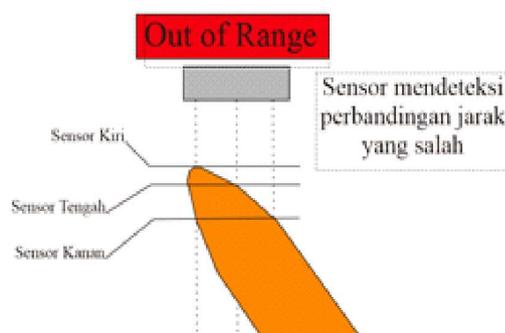
Kemungkinan terjadinya kemiringan pesawat adalah pesawat yang datang dari landasan yang menuju *Apron* datang dengan keadaan miring seperti pada Tabel 4.19, atau pesawat yang berbelok hingga menjadi miring.

Tabel 4.19 Sketsa Percobaan Kemiringan.

			
A. Percobaan Pesawat Miring ke Kiri	B. Percobaan Pesawat Miring ke Kanan	C. Pesawat Datang dari Samping Kiri	D. Pesawat Datang dari Samping Kanan

Pada Tabel 4.19 berikut adalah macam-macam percobaan kemiringan yang terjadi apabila kesalahan terjadi selama pemanduan pesawat. Fungsi dari percobaan ini adalah untuk menunjukkan bahwa alat VDGS ini mampu menentukan tingkat kesalahan yang terjadi apabila pesawat datang dari arah dan sudut yang salah.

A. Sketsa Percobaan Pesawat Miring Kekiri



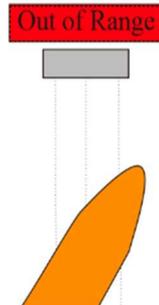
Gambar 4.39 Percobaan Pesawat Miring Kekiri.

Tabel 4.20 Data Sensor Percobaan Pesawat Miring Kekiri.

Data Sensor Pendeteksi (cm)				
No.	Sensor Kiri	Sensor Tengah	Sensor Kanan	Keterangan
1	50	60	70	Error
2	43	53	63	Error
3	67	77	-	Error
4	47	57	67	Error
5	56	66	76	Error
6	44	54	64	Error
7	39	49	59	Error
8	48	58	68	Error
9	44	54	64	Error
10	30	40	50	Error
11	69	79	-	Error
12	70	80	-	Error
13	46	56	66	Error
14	72	82	-	Error
15	35	45	55	Error
16	65	75	-	Error
17	34	44	54	Error
18	56	66	76	Error
19	35	45	55	Error
20	69	79	-	Error

Dijelaskan pada Tabel 4.20 diatas dan pada Gambar 4.39 menunjukkan bahwa sensor mendeteksi jarak perbandingan yang salah dari sudut 30° Gambar 4.39, eror yang terjadi menunjukkan bahwa tingkat keberhasilan alat dalam mendeteksi adanya tingkat perbandingan yang salah. Hasil ini menunjukkan bahwa alat VDGS ini mampu mendeteksi eror kemiringan seperti pada gambar dengan tingkat keberhasilan 100%.

B. Sketsa Percobaan Pesawat Miring Kekanan



Gambar 4.40 Percobaan Pesawat Miring Kekanan.

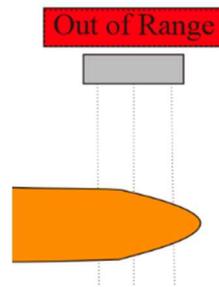
Tabel 4.21 Data Sensor Percobaan Pesawat Miring Kekanan.

Data Sensor Pendeteksi (cm)				
No.	Sensor Kiri	Sensor Tengah	Sensor Kanan	Keterangan
1	74	64	54	Error
2	76	56	46	Error
3	-	75	65	Error
4	-	77	67	Error
5	-	76	66	Error
6	74	64	54	Error
7	55	45	35	Error
8	44	34	24	Error
9	81	71	61	Error
10	73	63	53	Error
11	71	61	51	Error
12	-	76	66	Error
13	76	66	56	Error
14	72	62	52	Error
15	35	45	55	Error
16	-	82	72	Error
17	74	64	54	Error
18	56	46	36	Error
19	63	53	43	Error
20	-	79	69	Error

Pada Tabel 4.21 menunjukkan bahwa tingkat keberhasilan alat untuk mendeteksi eror yang *terjadi* selama pemanduan pada Gambar 4.40 adalah 100%,

ini dikatakan berhasil adalah karena sensor mendeteksi perbandingan jarak yang salah antara sensor tengah, sensor kiri, dan sensor kanan.

C. Sketsa Percobaan Pesawat yang Datang dari Samping Kiri



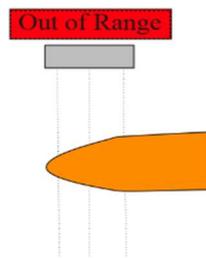
Gambar 4.41 Percobaan Pesawat Datang dari Samping Kiri.

Tabel 4.22 Data Sensor Percobaan Pesawat Datang dari Samping Kiri.

Data Sensor Percobaan Pesawat Datang dari Samping Kiri (cm)				
No.	Sensor Kiri	Sensor Tengah	Sensor Kanan	Keterangan
1	74	77	81	Error
2	43	46	50	Error
3	53	56	60	Error
4	62	65	69	Error
5	73	76	80	Error
6	25	28	32	Error
7	37	40	44	Error
8	63	66	70	Error
9	51	54	58	Error
10	73	76	80	Error
11	64	67	71	Error
12	44	47	51	Error
13	48	51	55	Error
14	38	41	45	Error
15	29	32	36	Error
16	39	42	46	Error
17	58	61	65	Error
18	67	70	74	Error
19	78	81	-	Error
20	77	80	-	Error

Percobaan ini adalah contoh kesalahan pesawat yang datang dari samping kiri Apron dengan sudut 90° , Hasil dari percobaan Gambar 4.41 adalah pada Tabel 4.22 dengan keberhasilan alat VDGS menentukan kesalahan yang terjadi pada perbandingan jarak yang salah adalah 100%.

D. Sketsa Percobaan Pesawat Datang dari Samping Kanan



Gambar 4.42 Percobaan Pesawat Datang dari Samping Kanan.

Tabel 4. 22 Data Sensor Percobaan Pesawat Datang dari Samping Kanan.

Data Sensor Percobaan Pesawat Datang dari Samping Kanan (cm)				
No.	Sensor Kiri	Sensor Tengah	Sensor Kanan	Keterangan
1	60	55	52	Error
2	76	71	68	Error
3	53	48	45	Error
4	80	75	72	Error
5	79	74	71	Error
6	72	57	54	Error
7	45	40	37	Error
8	35	30	27	Error
9	69	64	61	Error
10	73	68	65	Error
11	55	50	47	Error
12	70	65	62	Error
13	-	81	78	Error
14	62	57	54	Error
15	66	61	58	Error
16	71	66	63	Error
17	-	-	80	Error
18	46	41	38	Error
19	72	67	64	Error
20	-	78	75	Error

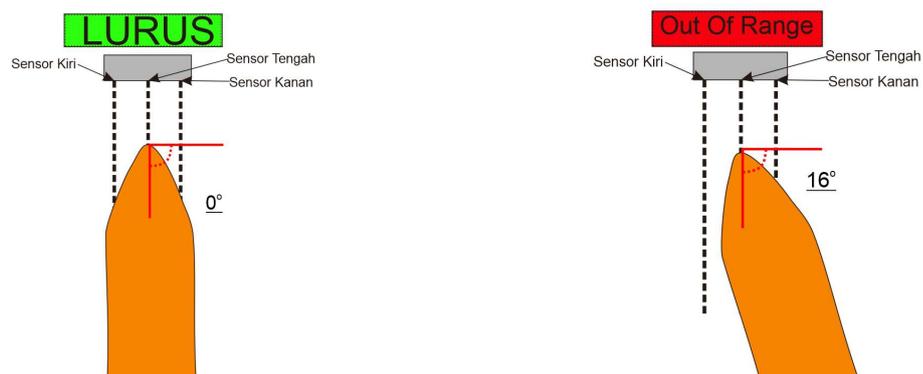
Hasil percobaan pada Gambar 4.42 menunjukkan hasil dengan tingkat keberhasilan mencapai 100% selama masa percobaan, Tabel 4.22 ini menunjukkan bahwa nilai jarak yang salah dalam proses pemanduan pemarkiran.

4. E. Hasil Percobaan Sketsa Kemiringan

Kesimpulan dari hasil percobaan diatas adalah 100% keberhasilan alat VDGS ini menentukan tingkat eror dari kemiringan dari berbagai sudut yang telah di coba, hasil ini menunjukkan bahwa alat VDGS mampu di gunakan dalam kondisi bandara yang memiliki *Apron* yang kurang memadai.

3. Pengujiann Kemiringan Berdasarkan Sudut

Pengujian kemiringan sudut adalah percobaan yang di tujukan untuk mendeteksi batas kemiringan pesawat yang bisa di deteksi oleh alat VDGS ini. Pengujian ini dilakukan dengan memiringkan pesawat berdasarkan sudutnya (derajat) hingga terjadi eror pada sistem.



Gambar 4.43 Sketsa Kemiringan Pesawat Berdasarkan Sudut.

Tabel 4.23 Pengujian Kemiringan Sudut.

Percobaan Tingkat Kemiringan Berdasarkan Sudut		
No	Sudut (Derajat)	Keterangan
1	0	Berhasil
2	1	Berhasil
3	2	Berhasil
4	3	Berhasil
5	4	Berhasil
6	5	Berhasil
7	6	Berhasil
8	7	Berhasil
9	8	Berhasil
10	9	Berhasil
11	10	Berhasil
12	11	Berhasil
13	12	Berhasil
14	13	Berhasil
15	14	Berhasil
16	15	Berhasil
17	16	Tidak Berhasil
18	17	Tidak Berhasil
19	18	Tidak Berhasil
20	19	Tidak Berhasil

Dari hasil tersebut bisa kita lihat pada Tabel 4.23 dan Gambar 4.43 menunjukkan keberhasilan sensor mendeteksi pada sudut tertentu hingga menghasilkan eror, eror tersebut adalah hasil dari kesalahan sensor yang gagal mendeteksi pada sudut $< 16^\circ$ ataupun mendeteksi perbandingan yang salah. Maka dari itu alat VDGS hanya dapat mendeteksi kemiringan hanya pada sudut 0° sampai dengan 15° , apabila sudut pesawat melebihi batas sudutnya, maka layar perintah akan menunjukkan perintah “OUT OF RANGE”.

4.4 Analisis

Hasil dari pengujian sistem keseluruhan adalah semua proses berhasil dan berfungsi sesuai dengan kondisi yang diharapkan, mulai dari pemilihan mode jenis pesawat hingga poses pemberhentian pesawat di *Apron* dengan tingkat keberhasilan mencapai 90%. Komponen-komponen sensor maupun kontrol penyusun VDGS dapat di simpulkan bekerja sesuai dengan fungsinya masing-masing dengan hasil yang ada pada data di atas, karena hasil data menunjukkan sensor dan Push Button berfungsi sesuai dengan kondisi yang di harapkan pada perencanaan. Dengan data di atas juga dapat di simpulkan *wirning* yang di kerjakan telah seuai dengan gambar rancangan yang dibuat.

4.5 Kelebihan dan Kekurangan

Kelebihan dan kekurangan dari sistem *Visual Docking Guidance System* ini adalah sebagai berikut.

4.5.1 Kelebihan

1. Tidak Memerlukan sumber daya manusia yang banyak, hanya membutuhkan 1 sumber daya manusia sebagai operator sistem.
2. Hanya menggunakan 1 *Microcontroller* yaitu Arduino Mega 2560.
3. Memiliki fungsi untuk menentukan jenis pesawat yang akan masuk. Dengan menggunakan Push Button, alat ini bisa memilih tipe pesawat yang diinginkan untuk masuk ke landasan *Apron*.
4. Sensor VL53L0X akurat sehingga lebih mudah mengatur jarak pesawat yang akan datang. Sensor ini menggunakan laser sensor yang memberikan data dengan perkiraan waktu 0,1 detik, sehingga kita dapat menentukan jarak sensor yang mendeteksi jarak dengan cepat dan akurat.
5. Dengan rumus perbandingan jarak antara sensor kiri, tengah, dan kanan. Alat ini dapat mendeteksi adanya kemiringan yang terjadi sema proses pemanduan pesawat pada sudut kemiringan $0^{\circ} \sim 15^{\circ}$, dan menentukan eror yang terjadi selama pemanduan apabila pesawat datang dari sudut yang salah.

4.5.2 Kekurangan

1. Tidak dilengkapi kamera pengawas lapangan untuk operator, dikarenakan menambahkan kamera pengawas membutuhkan perangkat tambahan untuk alat bisa bekerja dan itu akan menjadi sebuah hambatan yaitu membutuhkan tambahan power supply.
2. Alat ini tidak memiliki fungsi untuk menanggulangi kesalahan (Error), dikarenakan alat ini hanya bertujuan untuk membantu pemanduan pesawat.
3. Tidak memiliki fitur untuk membedakan jenis-jenis pesawat secara otomatis kepada pesawat yang akan masuk pada Apron/Landasan bandara udara, karena alat ini hanya sebatas memandu pesawat yang datang ke Apron dan berhenti pada jarak yang sudah di tentukan.
4. Alat ini berupa purwarupa yang tidak tahan dalam segala macam kondisi mengingat alat ini diharuskan bekerja di luar ruangan yang akan berhadapan dengan segala macam cuaca. Jika alat ini ingin di buat dengan sekali aslinya, maka harus menggunakan perangkat-perangkat yang memenuhi kebutuhan seperti tahan lama dan panel box yang kuat dan tahan basah.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil perancangan sistem *Visual Docking Guidance System* untuk sistem pemarkiran bandara pada *Apron*, maka dapat ditarik kesimpulan sebagai berikut:

1. Penelitian ini dibuat perancangan *Visual Docking Guidance System* adalah dengan menggunakan Arduino Mega 2560 untuk kontrol otomatis pada alat VDGS. Menggunakan 12 input yang terdiri dari 9 sensor VL53L0X sebagai sensor utama dari kerja sistem, dan 3 *push button* sebagai kontrol manual. Terdapat 1 *output* yang di gunakan adalah LCD I2C sebagai monitor yang akan memandu pilot pesawat mulai dari datangnya pesawat yang akan memunculkan perintah “LURUS” yang memerintahkan pilot untuk memajukan pesawat secara perlahan, Perintah “KANAN” / “KIRI” untuk mengatur kemiringan pesawat, dan perintah “STOP” untuk memerintahkan pilot untuk berhenti. Dengan tercapainya alat *Visual Docking Guidance System* dengan tingkat keberhasilan 100% ini dapat memudahkan dan mengefisiensikan proses pemanduan pesawat terbang di bandara.
2. Perancangan VDGS membutuhkan perhitungan berdasarkan standarisasi ukuran *apron* dan ukuran pesawat yang akan masuk ke *apron*, tergantung dari jenis pesawat maka standar dari ukuran apron bisa berubah-ubah. Berdasarkan jenis pesawat pada penelitian ini standar pemberhentian atau pada perintah “STOP” di bagi menjadi 3, yaitu jenis *Wide Body* yang

membutuhkan 10 cm, jenis *Narrow Body* pada jarak 8 cm, dan jenis Perintis pada jarak 5 cm. Ukuran jarak pada penelitian ini di simpulkan berdasarkan perbandingan dengan ukuran yang aslinya yaitu dengan perbandingan 1:20. Alat VDGS ini menggunakan 3 *push button* untuk menentukan sensor yang digunakan untuk memandu pesawat yang dibagi menjadi 3 Mode yaitu Mode 1 untuk pesawat jenis *Wide Body*, Mode 2 untuk jenis pesawat *Narrow Body*, dan Mode 3 untuk pesawat jenis Perintis.

3. Dalam percobaan pemanduan *Visual Docking Guidance System*, terdapat kesalahan yang mungkin akan terjadi selama proses pemanduan pesawat seperti terjadinya kemiringan pesawat atau sensor tidak mendeteksi keberadaan pesawat di landasan utama. Dari hasil analisis kesalahan yang terjadi pada proses pemanduan, alat VDGS ini menyatakan tingkat kemiringan pada sudut 0° sampai dengan 15° , dan rata-rata batas jarak yang bisa sensor deteksi pada Mode 1 adalah pada 78 cm, Mode 2 adalah 68 cm, dan Mode 3 adalah pada 60 cm.

5.2 Saran

1. Untuk menyempurnakan VDGS pada penelitian ini perlu ditambahkan fungsi *Monitoring* untuk memantau keadaan saat sistem mulai hingga selesai beroperasi supaya tidak ada halangan yang mengganggu fungsi sensor.
2. Sensor jarak bisa menggunakan sensor berkualitas tinggi untuk jangkauan jarak yang lebih jauh dan efektif, dan alat lainnya yang di butuhkan sesuai dengan standarisasi industri penerbangan.

DAFTAR PUSTAKA

- Amaral, G. et al. 2013. “No 主観的健康感を中心とした在宅高齢者における健康関連指標に関する共分散構造分析Title.” *Journal of Petrology* 369(1): 1689–99.
<http://dx.doi.org/10.1016/j.jsames.2011.03.003><https://doi.org/10.1016/j.gr.2017.08.001><http://dx.doi.org/10.1016/j.precamres.2014.12.018><http://dx.doi.org/10.1016/j.precamres.2011.08.005><http://dx.doi.org/10.1080/00206814.2014.902757><http://dx.doi.org/10.1016/j.precamres.2011.08.005>
- Andriani, Diah. 2015. “Analisa Kelayakan Dimensi Runway , Taxiway , Dan Apron (Studi Kasus Bandar Udara Soekarno-Hatta Dengan Pesawat Airbus 380).” (March 2019).
- Dell, Manuale, and Assistente Al. 2003. “Marshaller Syllabus.”
- Denny Dermawan; M. Jalu Purnomo. 2016. “PERANCANGAN TAMPILAN VISUAL DOCKING GUIDANCE SYSTEM (VDGS) PADA SISTEM PARKIR PESAWAT TERBANG.” : 181–91.
- Fatonah, Feti. 2013. “Pentingnya Optimalisasi Sistem Parkir Pesawat Udara Pada Remote Apron Terminal B Dan C Di Bandara Soetta Cengkareng.”
- Fitriandi, Afrizal. 2016. “Rancang Bangun Alat Monitoring Arus Dan Tegangan Berbasis Mikrokontroler Dengan SMS Gateway.” *Rancang Bangun Alat Monitoring Arus dan Tegangan Berbasis Mikrokontroler dengan SMS*

Gateway 10.

Jin, Tony. 2015. "Arduino Industrial 101." <https://store.arduino.cc/usa/arduino-industrial-101> (September 27, 2019).

Kesuma, Ari, Jeffry F.R. Silaban, and M. Ilyas Firmansyah. 20AD. "SEJARAH, JENIS, DAN SPESIFIKASI PESAWAT PENERBANGAN SIPIL." *SEJARAH, JENIS, DAN SPESIFIKASI PESAWAT PENERBANGAN SIPIL* (0215011037).

Lee, Hai-Wu. 2018. "The Study of Using Arduino to Control Aircraft Combined with Obstacle Avoidance Warning Function." *2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)* 1: 1–2.

Maria, Putut Son, and Elva Susianti. 2019. "Uji Kinerja Surface Scanner 3D Menggunakan Sensor VL53L0X Dan Mikrokontroler ATMEGA8535." *Jurnal Teknik Elektro* 11(1): 1–8.

Microelectronics, ST. 2018. "VL53L0X World 's Smallest Time-of-Flight Ranging and Gesture Detection." (April): 40.

Minohara, Tatsuo. 2005. "A Visual Guidance in the Final Approach to Space Station." : 1–11.

PT Angkasa Pura II (Persero). 2013. "Bandara Changi Terbaik, Soetta Berkembang Pesat - PT Angkasa Pura II." www.angkasapura2.co.id.
<http://www.angkasapura2.co.id/en/informasi/berita-detail/bandara-changi->

terbaik-soetta-berkembang-pesat-1240 (September 30, 2019).

Quarta Aviation Training Center. 2016. "AIRCRAFT MARSHALLER."

<http://quartaaviationtrainingcenter.blogspot.com/2016/11/apa-itu-aircraft-marshaller-marshaller.html> (September 27, 2019).

Saghoa, Yohanes C., Sherwin R. U. A. Somppie, and Novi M. Tulung. 2018.

"Kotak Penyimpanan Uang Berbasis Mikrokontroler Arduino Uno." *Kotak Penyimpanan Uang Berbasis Mikrokontroler Arduino Uno* 7(2): 167–74.

Santoso, Hari. 2015. *Panduan Praktis Arduino Untuk Pemula*.

Saputro, Jimmy Harto, and Tejo Sukmadi. 2013. "ANALISA PENGGUNAAN

LAMPU LED PADA PENERANGAN DALAM RUMAH." *ANALISA PENGGUNAAN LAMPU LED PADA PENERANGAN DALAM RUMAH*.

SKEP 77-VI-2005 Dirjen Perhubungan Udara. 2005. "Persyaratan Teknis

Pengoperasian Fasilitas Teknik Bandar Udara. Direktur Jenderal Perhubungan Udara." *Skep/77/Vi/2005*.

www.nasa.gov. 2010. "Parts of an Airplane." www.nasa.gov.

Grantham, Russell (February 29, 2008). "Delta's new Boeing 777 Can Fly Farther,

Carry More". The Atlanta Journal-Constitution. Delta will put the new "triple seven" — as airline folks call the jet — into service March 8.

Norris, Guy and Mark Wagner (2001). "Airbus A340 and A330". St. Paul,

Minnesota: MIB Publishing, 2001.

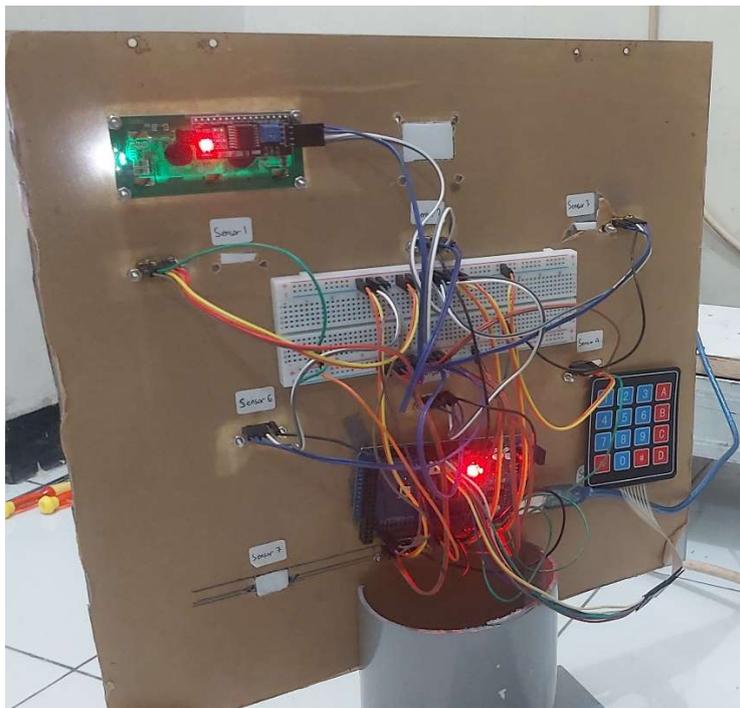
Goyer, Robert (April 26, 2008). "Cessna Caravan Perfected?". Flying Magazine.

Retrieved 23 May 2017.

LAMPIRAN



Gambar 1 Percobaan Alat VDGS



Gambar 2 *Wiring* Alat VDGS.



Gambar 3 Percobaan LCD I2C pada Alat VDGS



Gambar 4 Pemasangan Sensor VL53L0X.



Gambar 5 Tampilan Alat VDGS.