

**HUBUNGAN ANTARA POWER PANCARAN DENGAN  
LEBAR PANCARAN MENGGUNAKAN MATLAB DAN  
APLIKASI KONVERSI PARAMETER UNTUK KALIBRASI  
ILS (INSTRUMENT LANDING SYSTEM)**

**TUGAS AKHIR**



Disusun Oleh:  
**I GEDE NARA BAYU KUSUMA**  
NIT. 30218011

**PROGRAM STUDI DIPLOMA 3 TEKNIK NAVIGASI UDARA  
POLITEKNIK PENERBANGAN SURABAYA  
2021**

**HUBUNGAN ANTARA POWER PANCARAN DENGAN  
LEBAR PANCARAN MENGGUNAKAN MATLAB DAN  
APLIKASI KONVERSI PARAMETER UNTUK KALIBRASI  
ILS (INSTRUMENT LANDING SYSTEM)**

**TUGAS AKHIR**

Diajukan sebagai Syarat Mendapatkan Gelar Ahli Madya (A.md.) Pada Program  
Studi Diploma 3 Teknik Navigasi Udara



Disusun Oleh:  
**I GEDE NARA BAYU KUSUMA**  
NIT. 30218011

**PROGRAM STUDI DIPLOMA 3 TEKNIK NAVIGASI UDARA  
POLITEKNIK PENERBANGAN SURABAYA  
2021**

## LEMBAR PERSETUJUAN

### HUBUNGAN ANTARA POWER PANCARAN DENGAN LEBAR PANCARAN MENGGUNAKAN MATLAB DAN APLIKASI KONVERSI PARAMETER UNTUK KALIBRASI ILS (INSTRUMENT LANDING SYSTEM)

Oleh:  
IGEDE NARA BAYU KUSUMA  
NIT: 30218011

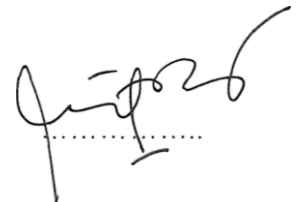
Disetujui untuk diujikan pada  
Surabaya, 12 Agustus 2021

Pembimbing I : Dr. MOCHAMMAD RIFA'I, ST. M.Pd  
NIP. 19770216 199903 1 003



.....

Pembimbing II : MEITA MAHARANI SUKMA, M.Pd  
NIP. 19800502 200912 2 002



.....

## LEMBAR PENGESAHAN

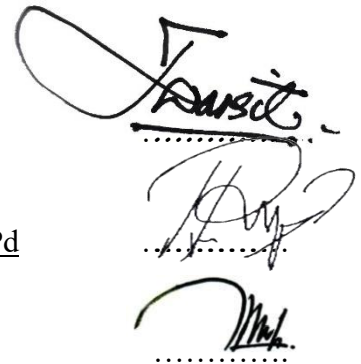
### HUBUNGAN ANTARA POWER PANCARAN DENGAN LEBAR PANCARAN MENGGUNAKAN MATLAB DAN APLIKASI KONVERSI PARAMETER UNTUK KALIBRASI ILS (INSTRUMENT LANDING SYSTEM)

Oleh:  
I GEDE NARA BAYU KUSUMA  
NIT. 30218011


Telah dipertahankan dan dinyatakan lulus pada Ujian Tugas Akhir  
Program Pendidikan Diploma 3 Teknik Navigasi Udara  
Politeknik Penerbangan Surabaya  
Pada tanggal: 12 Agustus 2021

Panitia Penguji:

1. Ketua : TOTOK WARSILO, S.SiT, MM  
NIP. 19570316 197703 1 001
2. Sekertaris : Dr. MOCHAMMAD RIFA'I, ST. M.Pd  
NIP. 19770216 199903 1 003
3. Anggota : TEGUH IMAM SUHARTO, MT  
NIP. 19910913 201503 1 003



Ketua Program Studi  
D 3 Teknik Navigasi Udara



NYARIS PAMBUDIYATNO, S.SiT, M.MTr  
NIP. 19820525 200502 1 001

## PERNYATAAN KEASLIAN DAN HAK CIPTA

Saya yang bertanda tangan di bawah ini :

Nama : I Gede Nara Bayu Kusuma

NIT : 30218011

Program Studi : D3 Teknik Navigasi Udara

Judul Tugas Akhir : Hubungan Antara Power Pancaran Dengan Lebar Pancaran Menggunakan Matlab Dan Aplikasi Konversi Parameter Untuk Kalibrasi ILS (Instrument Landing System)

dengan ini menyatakan bahwa :

1. Tugas Akhir ini merupakan karya asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Politeknik Penerbangan Surabaya maupun di Perguruan Tinggi lain, serta dipublikasikan, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
2. Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan Hak Bebas Royalti Non Eksklusif (Non-Exclusive Royalty-Free Right) kepada Politeknik Penerbangan Surabaya beserta perangkat yang ada (jika diperlukan). Dengan hak ini, Politeknik Penerbangan Surabaya berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir saya dengan tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya. Apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Politeknik Penerbangan Surabaya.

Surabaya,  
Yang membuat pernyataan



I Gede Nara Bayu Kusuma  
NIT. 30218011

## KATA PENGANTAR

Puji syukur kami panjatkan kepada tuhan yang maha esa karena berkat limpahan rahmat dan hidayahnya, Tugas Akhir yang berjudul HUBUNGAN ANTARA POWER PANCARAN DENGAN LEBAR PANCARAN MENGGUNAKAN MATLAB DAN APLIKASI KONVERSI PARAMETER UNTUK KALIBRASI ILS (INSTRUMENT LANDING SYSTEM) ini dapat diselesaikan dengan baik. Penyusunan Tugas Akhir ini dimaksudkan sebagai Syarat Mendapatkan Gelar Ahli Madya (A.md) Program Studi Diploma 3 Teknik Navigasi Udara di Politeknik Penerbangan Surabaya. Ucapan terima kasih kami sampaikan kepada segenap pihak yang telah membantu selama proses penyusunan Tugas Akhir ini, terutama kepada :

1. Bapak M. Andra Aditiyawarman, ST, MT, selaku Direktur Politeknik Penerbangan Surabaya.
2. Bapak Nyaris Pambudiyatno, S.SiT, M.MTr, selaku Ketua Program Studi Teknik Navigasi Udara di Politeknik Penerbangan Surabaya.
3. Bapak Dr. M. Rifa'i, ST, M.Pd dan Ibu Meita Maharani Sukma, M.Pd selaku Dosen Pembimbing.
4. Seluruh dosen dan civitas akademika Pogram Studi D3 Teknik Navigasi Udara Poltekbang Surabaya.
5. Kepada orang tua saya, serta saudara-saudara saya yang telah memberikan doa, semangat dan dukungan yang diberikan.
6. Rekan-rekan sekelas, atas kebersamaan dan kerjasamanya.
7. Dan semua pihak yang tidak dapat penulis tuliskan satu persatu yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari sepenuhnya bahwa Tugas Akhir ini jauh dari kata sempurna, oleh karena itu penulis mengharapkan saran dan kritik yang bersifat membangun serta semoga Tugas Akhir ini berguna bagi penulis pada khususnya dan bagi pembaca umumnya.

Surabaya, 1 Agustus 2021

Penulis

## ABSTRAK

### HUBUNGAN ANTARA POWER PANCARAN DENGAN LEBAR PANCARAN MENGGUNAKAN MATLAB DAN RANCANGAN APLIKASI KONVERSI PARAMETER UNTUK KALIBRASI ILS (INSTRUMENT LANDING SYSTEM)

Oleh :

I Gede Nara Bayu Kusuma  
NIT : 30218011

Instrument Landing Sistem (ILS) sebagai peralatan navigasi udara, berfungsi sebagai alat bantu dalam melakukan pendaratan agar tepat di landasan. ICAO mensyaratkan adanya pengujian atau kalibrasi terhadap peralatan ILS. Tujuan adanya kalibrasi adalah untuk memastikan peralatan bekerja dengan baik sesuai dengan parameter yang telah ditetapkan oleh ICAO. Untuk kalibrasi indikator *width* dan DDM terdapat aspek-aspek yang perlu diperhatikan, yaitu adanya hubungan antara *power* dengan *width* dan DDM dengan *angle*.

Metode penelitian ini yaitu dengan memilih parameter yang ingin dikalibrasi. Kemudian memasukan nilai skala awal dan nilai skala kalibrasi yang didapatkan dari petugas kalibrasi setelah dilakukan penyetulan pada parameter ILS yang dikalibrasi dan dilanjutkan dengan melakukan penghitungan di aplikasi untuk mendapatkan nilai skala yang baru sampai mendapatkan hasil yang diinginkan. Selanjutnya membuat grafik perbandingan antara DDM localizer dengan  $\mu A$ , lebar pancaran localizer dengan daya pancar localizer, DDM glide path dengan sudut kemiringan glide path, dan lebar pancaran glide path dengan daya pancar glide path menggunakan MATLAB.

Grafik perbandingan antara DDM localizer dengan  $\mu A$ , lebar pancaran localizer dengan daya pancar localizer, DDM glide path dengan sudut kemiringan glide path, dan lebar pancaran glide path dengan daya pancar glide path memberikan hasil bahwa nilai grafik antara DDM localizer dengan  $\mu A$  dan DDM glide path dengan sudut kemiringan glide path berbanding lurus sedangkan nilai grafik antara lebar pancaran dan daya pancar localizer dan glide path berbanding terbalik.

**Kata kunci** : Kalibrasi, ILS, Localizer, Glide Path, daya pancar, lebar pancaran, DDM, sudut kemiringan, MATLAB

## **ABSTRACT**

### ***THE RELATIONSHIP BETWEEN THE POWER BEAM WITH THE WIDTH OF THE BEAM USING MATLAB AND THE DRAFT APPLICATION PARAMETER CONVERSION FOR THE CALIBRATION OF THE ILS (INSTRUMENT LANDING SYSTEM)***

By :

I Gede Nara Bayu Kusuma  
NIT : 30218011

*Instrument Landing System (ILS) as air navigation equipment, serves as a tool in making landings to be precise on the runway. ICAO requires testing or calibration of ILS equipment. The purpose of calibration is to ensure the equipment is working properly in accordance with the parameters set by the ICAO. For the calibration of the indicator width and DDM are aspects that need to be considered, namely the existence of a relationship between the power with the width and DDM with the angle.*

*The method of this research is to choose the parameters that would like to calibrated. Then put the value of the scale of the initial and the value of the scale calibration recovered from the officer calibration after having performed fine-tuning! on the parameters of the ILS are calibrated and extended by performing the enumeration in the application to get the value of the scale of the new until I get my desired results. Next make a graphics comparison between the DDM localizer with  $\mu A$ , the width of the beam localizer with the power of the beam localizer, DDM glide path with the angle of inclination of the glide path, and the width of the beam of a glide path with the power of the glide path using MATLAB.*

*A comparison chart between the DDM localizer with  $\mu A$ , the width of the beam localizer with the power of the beam localizer, DDM glide path with a slope angle of the glide path, and width of the beam glide path with the power of the beam glide path gives the result that the value of the chart between the DDM localizer with  $\mu A$  and DDM glide path with a slope angle of glide path is directly proportional while the value of the chart between the width of the beam and transmit power localizer and glide path is inversely proportional.*

**Keywords** : Calibration, ILS, Localizer, Glide Path, power, width, DDM, angle, MATLAB



## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>LEMBAR PERSETUJUAN</b> .....	<b>ii</b>
<b>LEMBAR PENGESAHAN</b> .....	<b>iii</b>
<b>PERNYATAAN KEASLIAN DAN HAK CIPTA</b> .....	<b>iv</b>
<b>KATA PENGANTAR</b> .....	<b>v</b>
<b>ABSTRAK</b> .....	<b>vi</b>
<b>ABSTRACT</b> .....	<b>vii</b>
<b>DAFTAR ISI</b> .....	<b>viii</b>
<b>DAFTAR GAMBAR</b> .....	<b>x</b>
<b>DAFTAR TABEL</b> .....	<b>xi</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>xii</b>
<b>BAB 1 PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian .....	3
1.5 Manfaat penelitian.....	3
1.6 Sistematika Penulisan .....	3
<b>BAB 2 LANDASAN TEORI</b> .....	<b>5</b>
2.1 Pengertian ILS .....	5
2.1.1 Localizer .....	5
2.1.2 Glide Path.....	6
2.1.3 Marker Beacon .....	7
2.2 Pengertian Kalibrasi Penerbangan .....	7
2.2.1 Parameter yang Dinilai Dalam Kalibrasi ILS .....	9
2.3 Pengertian Konversi Nilai Satuan.....	10
2.4 Pengertian MATLAB.....	11
2.4.1 Fungsi Matlab.....	12
<b>BAB 3 METODE PENELITIAN</b> .....	<b>15</b>
3.1 Desain Penelitian .....	15
3.2 Perancangan Instrumen / Alat.....	16
3.2.1 Desain Instrumen/ Alat.....	16
3.2.2 Cara Kerja Instrumen/ Alat.....	16
3.2.3 Komponen Instrumen/ Alat.....	17
a. Microsoft Visual Studio.....	18
b. Matlab.....	19
3.3 Teknik Pengujian .....	19
3.4 Tempat Dan Alur Penelitian .....	20
<b>BAB 4 PEMBAHASAN</b> .....	<b>21</b>

4.1 Hasil Penelitian.....	21
4.1.1 Perangkat Lunak dan aplikasi.....	21
4.1.1.1 Program Software Aplikasi Konverter Kalibrasi ILS.....	21
4.1.1.2 Tampilan aplikasi Konverter Kalibrasi ILS.....	22
4.1.1.3 Pembuatan Grafik Menggunakan MATLAB.....	23
4.2 Pembahasan Hasil Penelitian.....	28
4.2.1 Pengujian Aplikasi.....	28
4.2.2 Pembahasan Grafik Pada MATLAB.....	32
<b>BAB 5 PENUTUP.....</b>	<b>40</b>
5.1 Kesimpulan.....	40
5.2 Saran.....	40
<b>DAFTAR PUSTAKA.....</b>	<b>42</b>
<b>LAMPIRAN</b>	

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Pola Pancaran Localizer.....	6
Gambar 2.2 Pola Pancaran Glide Path.....	7
Gambar 2.3 Contoh Data Hasil Kalibrasi ILS.....	8
Gambar 2.4 Jendela Software MATLAB.....	12
Gambar 2.5 Contoh Grafik Pada MATLAB.....	13
Gambar 3.1 Urutan Desain Penelitian.....	15
Gambar 3.2 Urutan Desain Instrumen/alat.....	16
Gambar 3.3 Urutan cara Kerja Instrumen/alat.....	17
Gambar 3.4 Tampilan pada Microsoft Visual Studio.....	18
Gambar 3.5 Tampilan pada Matlab.....	19
Gambar 4.1 Coding Aplikasi Konverter Kalibrasi ILS.....	21
Gambar 4.2 Tampilan Awal Pada Aplikasi Konverter Kalibrasi ILS.....	22
Gambar 4.3 Tampilan Bagian Indikator Pada Aplikasi Konverter Kalibrasi ILS.....	22
Gambar 4.4 Aplikasi Konverter Kalibrasi ILS.....	28
Gambar 4.5 Tampilan Indikator Localizer Width.....	28
Gambar 4.6 Input Data Pada Indikator Localizer Width.....	29
Gambar 4.7 Tampilan Indikator Glide Path Width.....	29
Gambar 4.8 Input Data Pada Indikator Glide Path Width.....	30
Gambar 4.9 Tampilan Indikator Glide Path Angle.....	30
Gambar 4.10 Input Data Pada Indikator Glide Path Angle.....	31
Gambar 4.11 Tampilan Indikator Localizer DDM.....	31
Gambar 4.12 Input Data Pada Indikator Localizer DDM.....	32
Gambar 4.13 Grafik DDM Localizer Dengan $\mu A$ .....	32
Gambar 4.14 Grafik Width Localizer Dengan Daya Pancar(Course Width= $4,38^\circ$ dan Panjang Runway 2500 Meter).....	33
Gambar 4.15 Grafik Width Localizer Dengan Daya Pancar(Course Width= $3,71^\circ$ dan Panjang Runway 3000 Meter).....	34
Gambar 4.16 Grafik Width Localizer Dengan Daya Pancar(Course Width= $3,23^\circ$ dan Panjang Runway 3500 Meter).....	35
Gambar 4.17 Grafik Width Localizer Dengan Daya Pancar(Course Width= $2,85^\circ$ dan Panjang Runway 4000 Meter).....	36
Gambar 4.18 Grafik Panjang Runway Dengan Width Localizer.....	37
Gambar 4.19 Grafik Angle Glide Path Dengan DDM.....	38
Gambar 4.20 Grafik Width Glide Path Dengan Daya Pancar.....	39

## DAFTAR TABEL

	Halaman
Tabel 2.1 Contoh Konversi Modulation.....	10
Tabel 2.2 Contoh Konversi DDM/Angle.....	10
Tabel 2.3 Contoh Konversi Width. ....	11
Tabel 3.1 Alur Pembuatan Proposal Tugas Akhir .....	20
Tabel 4.1 Tabel DDM Localizer Dengan $\mu A$ .....	33
Tabel 4.2 Tabel Width Localizer Dengan Daya Pancar(Course Width= $4,38^\circ$ dan Panjang Runway 2500 Meter).....	34
Tabel 4.3 Tabel Width Localizer Dengan Daya Pancar(Course Width= $3,71^\circ$ dan Panjang Runway 3000 Meter).....	34
Tabel 4.4 Tabel Width Localizer Dengan Daya Pancar(Course Width= $3,23^\circ$ dan Panjang Runway 3500 Meter).....	35
Tabel 4.5 Tabel Width Localizer Dengan Daya Pancar(Course Width= $2,85^\circ$ dan Panjang Runway 4000 Meter).....	36
Tabel 4.6 Tabel Panjang Runway Dengan Width Localizer.....	37
Tabel 4.7 Tabel Angle Glide Path Dengan DDM.....	38
Tabel 4.8 Tabel Width Glide Path Dengan Daya Pancar.....	39

## DAFTAR LAMPIRAN

	Halaman
Lampiran A Proses Coding Dari Aplikasi Konversi Besaran Kalibrasi ILS ....A1-A12	

## DAFTAR PUSTAKA

Tommy. Penjelasan Konversi Satuan Hitung Dalam Matematika. Kotak pintar. Diakses pada 12 Maret 2021 melalui <https://kotakpintar.com/penjelasan-konversi-satuan-hitung-dalam-matematika/>

Hemera. (2019, Oktober 21). Memahami Microsoft Visual Studio. Hemera Academy. Diakses pada 12 Maret 2021 melalui <https://itlearningcenter.id/memahami-microsoft-visual-studio/>

Advernesia. (2020, September 21). Pengertian MATLAB dan Kegunaannya. Advernesia. Diakses pada 23 Maret 2021 melalui <https://www.advernesia.com/blog/matlab/apa-itu-matlab/>

Swanstatistic. (2018, Desember 18). Pengertian MATLAB. SWAN. Diakses pada 15 Juni 2021 melalui [swanstatistic.com/pengertian-matlab/](http://swanstatistic.com/pengertian-matlab/)

AC 171-5 KALIBRASI 05-2010

Manual Book Instrument Landing System

## LAMPIRAN A

### Proses Coding Dari Aplikasi Konversi Besaran Kalibrasi ILS

#### A. Localizer DDM

```
B. def LocalizerDdm():
C.     def clear():
D.         in_SE.set('')
E.         in_DK.set('')
F.         out.set('')
G.
H.     def calculate(SE,DK,Stn):
I.         if Stn == "DDM":
J.             return SE-DK
K.         else:
L.             return (SE-DK)*1000
M.
N.     def callback():
O.         try:
P.             SE = float(in_field.get())
Q.             DK = float(in_field2.get())
R.             Stn = out_unit.get()
S.             out.set(calculate(SE,DK,Stn))
T.         except ValueError:
U.             out.set('Invalid input')
V.             return None
W.
X.
Y.     root = Toplevel()
Z.     root.title("Localizer DDM")
AA.    root.resizable(0,0)
BB.
CC.    #Initiate frame
DD.    mainframe = ttk.Frame(root,padding="3 3 12 12")
EE.    mainframe.pack(fill=BOTH, expand=1)
FF.    titleLabel = Label(mainframe, text="Localizer DDM", font = ("
    Arial",12,"bold"),justify = CENTER).grid(column=2,row=1)
GG.    in_SE = StringVar()
HH.    in_DK = StringVar()
II.
JJ.    out_unit = StringVar()
KK.    out_unit.set("DDM")
```

```

LL.     out = StringVar()
MM.
NN.     # Add input field
OO.     in_field = ttk.Entry(mainframe, width=20, textvariable=in_SE)
PP.     in_field.grid(row=2, column=2, sticky=(W, E))
QQ.     in_field2 = ttk.Entry(mainframe, width=20, textvariable=in_DK
    )
RR.     in_field2.grid(row=3, column=2, sticky=(W, E))
SS.     titleLabel = Label(mainframe, text="Skala Exiting", font = ("
    Arial",10,"bold"),justify = CENTER).grid(column=1,row=2)
TT.     titleLabel = Label(mainframe, text="DDM Kalibrasi", font = ("
    Arial",10,"bold"),justify = CENTER).grid(column=1,row=3)
UU.
VV.
WW.     # Add output field and drop-down
XX.     ttk.Entry(mainframe, textvariable=out, state="readonly").grid
    (column=2, row=5, sticky=(W, E))
YY.     in_select = OptionMenu(mainframe, out_unit, "DDM","µA").grid(
    column=3, row=5, sticky=W)
ZZ.
AAA.     calc_button = ttk.Button(mainframe, text="Calculate",co
    mmand=callback).grid(column=2, row=4, sticky=E)
BBB.     clear_button = ttk.Button(mainframe, text="Clear All",c
    ommand=clear).grid(column=2, row=6, sticky=E)
CCC.
DDD.     for child in mainframe.winfo_children(): child.grid_con
    figure(padx=5, pady=5)
EEE.
FFF.     in_field.focus()

```

## B. Glide Path Angle

```

C. def GlidePathAngle():
D.
E.     def graph():
F.         plt.style.use('ggplot')
G.         plt.style.use(['dark_background'])
H.
I.         fig = plt.figure()
J.         fig.suptitle("GlidePath Angle")
K.         ax1 = fig.add_subplot(1,1,1)

```



```

L.
M.
N.     def animate(i):
O.         graph_data = open('GlidePathAngle.txt','r').read()
P.         lines = graph_data.split('\n')
Q.         xs = []
R.         ys = []
S.         for line in lines:
T.             if len(line) > 1:
U.                 x, y = line.split(',')
V.                 xs.append(float(x))
W.                 ys.append(float(y))
X.         ax1.clear()
Y.         ax1.plot(xs, ys,'o-')
Z.         ax1.set_xlabel("Angle Kalibrasi")
AA.        ax1.set_ylabel("DDM (kalibrasi)")
BB.
CC.        ani = FuncAnimation(fig, animate, interval=1000)
DD.        plt.show()
EE.

FF.     def clear():
GG.         in_SE.set('')
HH.         in_AK.set('')
II.         out.set('')
JJ.
KK.     def calculate(SE,AK):
LL.         return(((AK-3.01)/0.01)*0.005)+SE
MM.
NN.     def callback():
OO.         try:
PP.             SE = float(in_field.get())
QQ.             AK = float(in_field2.get())
RR.             out.set(calculate(SE,AK))
SS.             outfile = open('GlidePathAngle.txt','a')
TT.             teks = "{},{}".format( AK,SE)
UU.             outfile.write(teks+"\n")
VV.             outfile.close()
WW.         except ValueError:
XX.             out.set('Invalid input')
YY.             return None

```

```

ZZ.
AAA.     def history():
BBB.         outfile = open('GlidePathAngle.txt','a')
CCC.         outfile.truncate(0)
DDD.         outfile.close()
EEE.

FFF.     root = Toplevel()
GGG.     root.title("GlidePath Angle")
HHH.     root.resizable(0,0)
III.
JJJ.     #Initiate frame
KKK.     mainframe = ttk.Frame(root,padding="3 3 12 12")
LLL.     mainframe.pack(fill=BOTH, expand=1)
MMM.     titleLabel = Label(mainframe, text="GlidePath Angle", f
        ont = ("Arial",12,"bold"),justify = CENTER).grid(column=2,row=1)
NNN.     in_SE = StringVar()
OOO.     in_AK = StringVar()
PPP.     out = StringVar()
QQQ.
RRR.     # Add input field
SSS.     in_field = ttk.Entry(mainframe, width=20, textvariable=
        in_SE)
TTT.     in_field.grid(row=2, column=2, sticky=(W, E))
UUU.     titleLabel = Label(mainframe, text="Skala Existing", fo
        nt = ("Arial",10,"bold"),justify = CENTER).grid(column=1,row=2)
VVV.     label = Label(mainframe,text="DDM").grid(column=3, row=
        2, sticky=W)
WWW.
XXX.     in_field2 = ttk.Entry(mainframe, width=20, textvariable
        =in_AK)
YYY.     in_field2.grid(row=3, column=2, sticky=(W, E))
ZZZ.     titleLabel = Label(mainframe, text="Angle Kalibrasi", f
        ont = ("Arial",10,"bold"),justify = CENTER).grid(column=1,row=3)
AAAA.     label = Label(mainframe,text="o").grid(column=3, row=3,
        sticky=W)
BBBB.
CCCC.     # Add output field and drop-down
DDDD.     ttk.Entry(mainframe, textvariable=out, state="readonly"
        ).grid(column=2, row=5, sticky=(W, E))

```

```

EEEE.         label = Label(mainframe,text="DDM").grid(column=3, row=
              5, sticky=W)
FFFF.
GGGG.         calc_button = ttk.Button(mainframe, text="Calculate",co
              mmand=callback).grid(column=2, row=4, sticky=E)
HHHH.         clear_button = ttk.Button(mainframe, text="Clear All",c
              ommand=clear).grid(column=2, row=6, sticky=E)
IIII.         graph_button = ttk.Button(mainframe, text="See Graph",c
              ommand=graph).grid(column=1, row=6, sticky=E)
JJJJ.         history_button = ttk.Button(mainframe, text="Clear Hist
              ory",command=history).grid(column=3, row=6, sticky=E)
KKKK.
LLLL.         for child in mainframe.winfo_children(): child.grid_con
              figure(padx=5, pady=5)
MMMM.
NNNN.         in_field.focus()

```

### C. Glide Path Width

```

D. def GlidePathWidth():
E.
F.     def graph():
G.         plt.style.use('ggplot')
H.         plt.style.use(['dark_background'])
I.
J.         fig = plt.figure()
K.         fig.suptitle("GlidePath Width")
L.         ax1 = fig.add_subplot(1,1,1)
M.         tick_spacing = 0.05
N.
O.     def animate(i):
P.         graph_data = open('GlidePathWidth.txt','r').read()
Q.         lines = graph_data.split('\n')
R.         xs = []
S.         ys = []
T.         tick_spacing = 0.05
U.         for line in lines:
V.             if len(line) > 1:
W.                 x, y = line.split(',')
X.                 xs.append(float(x))
Y.                 ys.append(float(y))

```

```

Z.         ax1.clear()
AA.         ax1.plot(xs, ys,'o-')
BB.         xs.sort()
CC.         ys.sort()
DD.         ax1.set_xlabel("Width Kalibrasi")
EE.         ax1.set_ylabel("Daya (Watt)")
FF.         ani = FuncAnimation(fig, animate, interval=1000)
GG.         plt.show()
HH.
II.         def clear():
JJ.             in_PE.set('')
KK.             in_WK.set('')
LL.             out.set('')
MM.
NN.         def calculate(PE,WK):
OO.             return ((WK**2)/0.49)*PE
PP.
QQ.         def callback():
RR.             try:
SS.                 PE = float(in_field.get())
TT.                 WK = float(in_field2.get())
UU.                 out.set(calculate(PE,WK))
VV.                 outfile = open('GlidePathWidth.txt','a')
WW.                 teks = "{},{}".format( WK,PE)
XX.                 outfile.write(teks+"\n")
YY.                 outfile.close()
ZZ.             except ValueError:
AAA.                 out.set('Invalid input')
BBB.                 return None
CCC.
DDD.         def history():
EEE.             outfile = open('GlidePathWidth.txt','a')
FFF.             outfile.truncate(0)
GGG.             outfile.close()
HHH.
III.         root = Toplevel()
JJJ.         root.title("GlidePath Width")
KKK.         root.resizable(0,0)
LLL.
MMM.         #Initiate frame
NNN.         mainframe = ttk.Frame(root,padding="3 3 12 12")

```

```

000.         mainframe.pack(fill=BOTH, expand=1)
PPP.         titleLabel = Label(mainframe, text="GlidePath Width", font = ("Arial",12,"bold"),justify = CENTER).grid(column=2,row=1)
QQQ.         in_PE = StringVar()
RRR.         in_WK = StringVar()
SSS.         out = StringVar()
TTT.
UUU.         # Add input field
VVV.         in_field = ttk.Entry(mainframe, width=20, textvariable=in_PE)
WWW.         in_field.grid(row=2, column=2, sticky=(W, E))
XXX.         titleLabel = Label(mainframe, text="Power Existing", font = ("Arial",10,"bold"),justify = CENTER).grid(column=1,row=2)
YYY.         label = Label(mainframe,text="Watt").grid(column=3, row=2, sticky=W)
ZZZ.
AAAA.        in_field2 = ttk.Entry(mainframe, width=20, textvariable=in_WK)
BBBB.        in_field2.grid(row=3, column=2, sticky=(W, E))
CCCC.        titleLabel = Label(mainframe, text="Width Kalibrasi", font = ("Arial",10,"bold"),justify = CENTER).grid(column=1,row=3)
DDDD.        label = Label(mainframe,text="o").grid(column=3, row=3, sticky=W)
EEEE.
FFFF.        # Add output field
GGGG.        ttk.Entry(mainframe, textvariable=out, state="readonly").grid(column=2, row=5, sticky=(W, E))
HHHH.        label = Label(mainframe,text="Watt").grid(column=3, row=5, sticky=W)
IIII.
JJJJ.        calc_button = ttk.Button(mainframe, text="Calculate",command=callback).grid(column=2, row=4, sticky=E)
KKKK.        clear_button = ttk.Button(mainframe, text="Clear All",command=clear).grid(column=2, row=6, sticky=E)
LLLL.        graph_button = ttk.Button(mainframe, text="See Graph",command=graph).grid(column=1, row=6, sticky=E)
MMMM.        history_button = ttk.Button(mainframe, text="Clear History",command=history).grid(column=3, row=6, sticky=E)
NNNN.
0000.        for child in mainframe.winfo_children(): child.grid_configure(padx=5, pady=5)

```

```
PPPP.  
QQQQ.         in_field.focus()
```

#### D. Localizer Width

```
E. def LocalizerWidth():  
F.  
G.     def graph():  
H.         plt.style.use('ggplot')  
I.         plt.style.use(['dark_background'])  
J.  
K.         fig = plt.figure()  
L.         fig.suptitle("Localizer Width")  
M.         ax1 = fig.add_subplot(1,1,1)  
N.  
O.  
P.     def animate(i):  
Q.         graph_data = open('LocalizerWidth.txt','r').read()  
R.         lines = graph_data.split('\n')  
S.         xs = []  
T.         ys = []  
U.         for line in lines:  
V.             if len(line) > 1:  
W.                 x, y = line.split(',')  
X.                 xs.append(float(x))  
Y.                 ys.append(float(y))  
Z.  
AA.         ax1.clear()  
BB.         ax1.plot(xs, ys,'o-')  
CC.         xs.sort()  
DD.         ys.sort()  
EE.         ax1.set_xlabel("Width Kalibrasi")  
FF.         ax1.set_ylabel("Daya (Watt)")  
GG.  
HH.         ani = FuncAnimation(fig, animate, interval=1000)  
II.         plt.show()  
JJ.  
KK.     def widthstandard(PLandasan,JThreshold):  
LL.         return math.degrees(math.atan(107/(PLandasan+JThreshold))  
MM.     ) * 2
```

```

NN.     def finalans(WStandard,WKalibrasi,PExisting):
OO.         return ((WKalibrasi**2)/(WStandard**2))*PExisting
PP.
QQ.     def clear():
RR.         in_JT.set('')
SS.         in_PL.set('')
TT.         in_WK.set('')
UU.         in_PE.set('')
VV.         out2.set('')
WW.         out1.set('')
XX.
YY.     def callback2():
ZZ.         try:
AAA.             WKalibrasi = float(in_field3.get())
BBB.             WStandard = float(in_field4.get())
CCC.             PExisting = float(in_field5.get())
DDD.             out2.set(finalans(WStandard,WKalibrasi,PExisting))
EEE.             outfile = open('LocalizerWidth.txt','a')
FFF.             teks = "{},{}".format( WKalibrasi,PExisting)
GGG.             outfile.write(teks+"\n")
HHH.             outfile.close()
III.
JJJ.         except ValueError:
KKK.             out2.set('Invalid input')
LLL.             return None
MMM.
NNN.     def callback():
OOO.         try:
PPP.             PLandasan = float(in_field1.get())
QQQ.             JThreshold = float(in_field2.get())
RRR.             out1.set(widthstandard(PLandasan,JThreshold))
SSS.
TTT.         except ValueError:
UUU.             out.set('Invalid input')
VVV.             return None
WWW.
XXX.     def history():
YYY.         outfile = open('LocalizerWidth.txt','a')
ZZZ.         outfile.truncate(0)
AAAA.         outfile.close()

```

```

BBBB.
CCCC.     root = Toplevel()
DDDD.     root.title("Localizer Width")
EEEE.     root.resizable(0,0)
FFFF.
GGGG.     #Initiate frame
HHHH.     mainframe = ttk.Frame(root,padding="3 3 12 12")
IIII.     mainframe.pack(fill=BOTH, expand=1)
JJJJ.     titleLabel = Label(mainframe, text="Localizer Width", f
          ont = ("Arial",12,"bold"),justify = CENTER).grid(column=2,row=1)
KKKK.
LLLL.     in_PL = StringVar()
MMMM.     in_JT= StringVar()
NNNN.     in_WK = StringVar()
OOOO.     in_PE = StringVar()
PPPP.
QQQQ.     out1 = StringVar()
RRRR.     out2 = StringVar()
SSSS.
TTTT.     # Add input field
UUUU.     in_field1 = ttk.Entry(mainframe, width=20, textvariable
          =in_PL)
VVVV.     in_field2 = ttk.Entry(mainframe, width=20, textvariable
          =in_JT)
WWWW.     in_field3 = ttk.Entry(mainframe, width=20, textvariable
          =in_WK)
XXXX.     in_field4 = ttk.Entry(mainframe, width=20, textvariable
          =out1)
YYYY.     in_field5 = ttk.Entry(mainframe, width=20, textvariable
          =in_PE)
ZZZZ.
AAAAA.     # Input grid
BBBBB.     in_field1.grid(row=2, column=2, sticky=(W, E))
CCCCC.     in_field2.grid(row=3, column=2, sticky=(W, E))
DDDDD.     in_field3.grid(row=9, column=2, sticky=(W, E))
EEEEEE.    in_field4.grid(row=8, column=2, sticky=(W, E))
FFFFF.     in_field5.grid(row=7, column=2, sticky=(W, E))
GGGGG.
HHHHH.     # Input Label
IIIIII.    titleLabel = Label(mainframe, text="Panjang landasan",
          font = ("Arial",10,"bold"),justify = CENTER).grid(column=1,row=2)

```



```

JJJJJ.      titleLabel = Label(mainframe, text="Jarak Threshold", f
           ont = ("Arial",10,"bold"),justify = CENTER).grid(column=1,row=3)
KKKKK.      titleLabel = Label(mainframe, text="Width Kalibrasi", f
           ont = ("Arial",10,"bold"),justify = CENTER).grid(column=1,row=9)
LLLLL.      titleLabel = Label(mainframe, text="Width Standar", fon
           t = ("Arial",10,"bold"),justify = CENTER).grid(column=1,row=8)
MMMMM.      titleLabel = Label(mainframe, text="Power Existing", fo
           nt = ("Arial",10,"bold"),justify = CENTER).grid(column=1,row=7)
NNNNN.
OOOOO.      # unit
PPPPP.      label = Label(mainframe,text="m").grid(column=3, row=2,
           sticky=W)
QQQQQ.      label = Label(mainframe,text="m").grid(column=3, row=3,
           sticky=W)
RRRRR.      label = Label(mainframe,text="o").grid(column=3, row=9,
           sticky=W)
SSSSS.      label = Label(mainframe,text="o").grid(column=3, row=8,
           sticky=W)
TTTTT.      label = Label(mainframe,text="Watt").grid(column=3, row
           =7, sticky=W)
UUUUU.
VVVVV.      # Add output field
WWWWW.      ttk.Entry(mainframe, textvariable=out1,state="readonly"
           ).grid(column=2, row=6, sticky=(W, E))
XXXXX.      label = Label(mainframe,text="o").grid(column=3, row=6,
           sticky=W)
YYYYY.      ttk.Entry(mainframe, textvariable=out2, state="readonly
           ").grid(column=2, row=12, sticky=(W, E))
ZZZZZ.      label = Label(mainframe,text="Watt").grid(column=3, row
           =12, sticky=W)
AAAAAA.
BBBBBB.      calc_button1 = ttk.Button(mainframe, text="Calculate",c
           ommand=callback).grid(column=2, row=4, sticky=E)
CCCCCC.      calc_button2 = ttk.Button(mainframe, text="Calculate",c
           ommand=callback2).grid(column=2, row=11, sticky=E)
DDDDDD.      clr_button = ttk.Button(mainframe, text="Clear All",com
           mand=clear).grid(column=2, row=13, sticky=E)
EEEEEE.      graph_button = ttk.Button(mainframe, text="See Graph",c
           ommand=graph).grid(column=1, row=13, sticky=E)
FFFFFF.      history_button = ttk.Button(mainframe, text="Clear Hist
           ory",command=history).grid(column=3, row=13, sticky=E)

```

```

GGGGGG.
HHHHHH.     for child in mainframe.wininfo_children(): child.grid_con
              figure(padx=5, pady=5)
IIIIII.
JJJJJJ.     in_field1.focus()
KKKKKK.     in_field2.focus()
LLLLLL.
MMMMMM.     #####
              #####
NNNNNN.     #####
              #####
000000.     button = Button(root,text="Quit",width=10,justify=CENTER,bd
              =3,command=quit)
PPPPPP.     button.place(x=365,y=315)
QQQQQQ.
RRRRRR.     button = Button(root,text="Localizer DDM",width=20,height=2
              ,activebackground="red",relief = RAISED, bd=3, justify = CENTER,c
              ommand=LocalizerDdm)
SSSSSS.     button.place(x=130,y=60)
TTTTTT.
UUUUUU.     button = Button(root,text="Localizer Width",width=20,height
              =2,activebackground="red",relief = RAISED, bd=3, justify = CENTER
              ,command=LocalizerWidth)
VVVVVV.     button.place(x=130,y=120)
WWWWWW.
XXXXXX.     button = Button(root,text="Glide Path Width",width=20,height
              t=2,activebackground="red",relief = RAISED, bd=3, justify = CENTE
              R,command=GlidePathWidth)
YYYYYY.     button.place(x=130,y=180)
ZZZZZZ.
AAAAAAA.     button = Button(root,text="Glide Path Angle",width=20,height
              t=2,activebackground="red",relief = RAISED, bd=3, justify = CENTE
              R,command=GlidePathAngle)
BBBBBB.     button.place(x=130,y=240)
CCCCCC.
DDDDDD.     root.mainloop()

```

## DAFTAR RIWAYAT HIDUP



I Gede Nara Bayu Kusuma, lahir di Tabanan, Bali pada tanggal 1 Februari 2000. Adalah anak pertama dari pasangan I Made Sudirka dan Ayu Putu Dewi Utari. Bertempat tinggal di Perum Citramas Jalan Noja Gang XXIX no 8, Banjar Abian Nangka Kaja, Kelurahan Kesiman Petilan, Kecamatan Denpasar Timur, Kota Denpasar, Bali. Mengenyam Sekolah Dasar pada tahun 2006 – 2012 di SDN 6 Kota Batam.

Melanjutkan hingga tamat Sekolah Menengah Pertama pada Tahun 2012 – 2015 di SMP Negeri 12 Batam. Melanjutkan Sekolah Menengah Atas pada Tahun 2015 dan tamat pada Tahun 2018 di SMA Negeri 3 Denpasar. Pada bulan September 2018 diterima menjadi Taruna Politeknik Penerbangan Surabaya pada Program Studi Diploma 3 Teknik Navigasi Udara Angkatan XI. Pengalaman On the Job Training (OJT) pada Semester 4 dan Semester 5 yang dimulai dari tanggal 12 Juli 2020 hingga tanggal 15 Februari 2021 di Perum Lembaga Penyelenggara Pelayanan Navigasi Penerbangan Indonesia atau di singkat LPPNPI (Airnav Indonesia) Cabang Denpasar.