TUGAS AKHIR ARTIFICIAL INTELLIGENCE

MEMPREDIKSI KETERLAMBATAN JADWAL PENERBANGAN MENGGUNAKAN ALGORITMA GENETIKA

Dosen Mata Kuliah :

Cakra Adipura Wicaksana, ST., MT

Disusun Oleh :

Dhafin Rizky Aulia (3337210045)

Fatwaraga Rafsajani (3337210031)

Satria Adjie Heriansyah (3337210035)

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS SULTAN AGENG TIRTAYASA

2023

I. Pendahuluan

1.1. Algoritma Genetika

Algoritma genetika adalah metode komputasional yang terinspirasi oleh prinsip-prinsip evolusi dan pewarisan genetik dalam proses evolusi alami. Algoritma ini digunakan untuk mencari solusi optimal atau mendekati solusi optimal dalam masalah optimasi atau pencarian ruang solusi yang besar.

Prinsip dasar algoritma genetika adalah menciptakan populasi awal dari individu-individu yang mewakili solusi potensial dalam bentuk kromosom atau genom. Setiap individu memiliki kualitas atau nilai fitness yang menggambarkan sejauh mana solusi tersebut memenuhi kriteria yang diberikan. Kemudian, individu-individu terpilih diteruskan melalui serangkaian operasi genetik seperti seleksi, rekombinasi, dan mutasi untuk menciptakan generasi baru individu-individu.

Dalam proses ini, individu-individu dengan nilai fitness yang lebih baik cenderung memiliki kesempatan yang lebih besar untuk berkontribusi pada generasi berikutnya, mirip dengan prinsip seleksi alam dalam evolusi alami. Proses ini diulang dalam beberapa generasi hingga ditemukan solusi yang memadai atau solusi yang mendekati optimal.

Algoritma genetika telah diterapkan dalam berbagai bidang, termasuk optimasi parameter, desain mesin, perencanaan jadwal, pembelajaran mesin, dan banyak lagi. Algoritma ini sangat berguna ketika pencarian solusi memerlukan ruang pencarian yang besar atau ketika tidak ada metode analitik yang efisien untuk mencari solusi optimal.

1.2. Kelebihan Dan Kekurangan

Algoritma genetika memiliki beberapa kelebihan dan kekurangan yang perlu dipertimbangkan :

Kelebihan Algoritma Genetika :

1. Kemampuan Pencarian Global

Algoritma genetika dapat melakukan pencarian dalam ruang solusi yang sangat besar dan kompleks. Mereka dapat menemukan solusi yang baik atau mendekati optimal di tengah banyak kemungkinan yang ada.

2. Fleksibilitas

Algoritma genetika dapat digunakan untuk berbagai jenis masalah optimasi, termasuk masalah diskrit dan kontinu. Mereka dapat memanipulasi representasi kromosom untuk mengatasi berbagai jenis variabel dan batasan.

3. Konvergensi Ke Solusi Optimal

Dalam beberapa kasus, algoritma genetika dapat mencapai solusi optimal atau solusi yang sangat baik dalam jumlah iterasi yang relatif sedikit. Mereka mampu mengeksplorasi dan memanfaatkan informasi dari banyak solusi alternatif.

4. Solusi Alternatif

Algoritma genetika dapat memberikan solusi alternatif yang beragam. Dalam beberapa situasi, ini dapat memberikan wawasan tambahan dan pilihan bagi pengguna.

Kekurangan Algoritma Genetika :

1. Kompleksitas Parameter

Algoritma genetika melibatkan banyak parameter yang perlu disesuaikan, seperti ukuran populasi, tingkat mutasi, dan metode seleksi. Menentukan parameter yang tepat dapat menjadi sulit dan memerlukan pemahaman yang mendalam tentang masalah yang dihadapi.

2. Kecepatan Komputasi

Algoritma genetika sering kali memerlukan waktu yang lebih lama untuk mencapai solusi optimal dibandingkan dengan metode optimasi lainnya. Ini terkait dengan kebutuhan untuk menghasilkan dan mengevaluasi banyak individu dalam setiap generasi.

3. Kemungkinan Terjebak di Optimum Lokal

Ada kemungkinan bahwa algoritma genetika dapat terjebak di dalam optimum lokal dan tidak mampu menemukan solusi yang lebih baik secara global. Ini terutama terjadi ketika ruang pencarian memiliki banyak puncak lokal yang menarik.

4. Kesulitan dalam Representasi

Pemilihan representasi kromosom yang tepat adalah faktor penting dalam kinerja algoritma genetika. Jika representasi tidak sesuai dengan masalah yang dihadapi, maka pencarian solusi optimal bisa menjadi lebih sulit.

1.3. Pengaplikasian

Pengaplikasian algoritma genetika dalam mencari akurasi berapa menit pesawat mengalami delay dapat melibatkan beberapa langkah sebagai berikut :

1. Representasi Kromosom :

Setiap individu dalam populasi awal akan direpresentasikan oleh kromosom yang mengkodekan nilai-nilai variabel yang relevan. Misalnya, kromosom dapat mengkodekan variabel-variabel seperti maskapai penerbangan, waktu keberangkatan, rute penerbangan, dan faktor-faktor lain yang dapat mempengaruhi delay pesawat.

2. Generasi Populasi Awal :

Sebuah populasi awal dari individu-individu acak akan dibuat dengan menggunakan representasi kromosom yang telah ditentukan. Jumlah dan karakteristik populasi awal akan bergantung pada kebutuhan dan batasan dari masalah yang dihadapi.

3. Evaluasi Fitness :

Setiap individu dalam populasi akan dievaluasi berdasarkan akurasi prediksi mereka tentang waktu delay pesawat. Evaluasi dilakukan dengan menggunakan data historis atau simulasi untuk memperkirakan waktu delay sesuai dengan variabel yang dikodekan dalam kromosom.

4. Seleksi :

Individu-individu dengan nilai fitness yang lebih tinggi akan lebih mungkin dipilih untuk menjadi orang tua dalam proses reproduksi. Seleksi dapat dilakukan menggunakan berbagai metode seperti seleksi turnamen atau seleksi proporsional berbasis fitness.

5. Rekombinasi :

Pasangan individu dipilih untuk melakukan operasi rekombinasi atau crossover. Bagian-bagian kromosom dari kedua individu tersebut dipertukarkan untuk menghasilkan individu keturunan baru yang menggabungkan informasi dari kedua orang tua.

6. Mutasi :

Operasi mutasi dilakukan pada individu keturunan dengan probabilitas tertentu. Pada operasi mutasi, nilai-nilai variabel dalam kromosom individu dapat diubah secara acak untuk memperkenalkan variasi baru ke dalam populasi.

7. Generasi Baru :

Dengan menggunakan langkah-langkah di atas, populasi baru yang terdiri dari individu-individu keturunan akan dihasilkan. Langkah-langkah evaluasi, seleksi, rekombinasi, dan mutasi akan diulang dalam beberapa generasi hingga konvergensi atau ditemukan solusi yang memadai.

8. Solusi Optimal :

Setelah proses evolusi berlangsung, individu dengan nilai fitness tertinggi dalam populasi akan mewakili solusi yang menghasilkan prediksi akurasi terbaik untuk waktu delay pesawat.

Dengan menggunakan algoritma genetika, pencarian parameter atau kombinasi variabel yang dapat memprediksi akurasi delay pesawat dapat dilakukan secara efisien dan menghasilkan solusi yang mendekati optimal.

1.4. Dataset

Dataset yang kami gunakan adalah data data penerbangan dari maskapai Tunisian aviation company, Tunisair. Dataset ini terdiri dari 108 ribu baris dan 10 kolom. Namun disini kami mengurangi jumlah dataset menjadi 10 ribu baris saja. Dikarenakan terlalu banyaknya data akan membuat nilai akurasi menjadi rendah.

Berikut adalah penjelasan detail mengenai isi dataset yang kami gunakan :

Nama Kolom	Deskripsi	Termasuk Fitur	Keterangan
ID	Primary Key sebuah data.	Tidak	Tidak termasuk kedalam variable x_train karena tidak berpengaruh terhadap keterlambatan jadwal penerbangan
DATOP	Tanggal penerbangan pesawat	Ya	Telah dilakukan perubahan, yaitu mengubah format tanggal menjadi bulan saja.
FLTID	ID penerbangan pesawat	Ya	-
DEPSTN	Titik keberangkatan pesawat	Ya	-
ARRSTN	Titik kedatangan pesawat	Ya	-
STD	Waktu keberangkatan pesawat yang dijadwalkan	Ya	-
STA	Waktu tiba pesawat yang dijadwalkan	Ya	-
STATUS	Waktu pasti kedatangan pesawat	Ya	-
AC	Kode pesawat	Ya	-
Target	Keterlambatan jadwal penerbangan (dalam menit)	Tidak	Telah dilakukan perubahan, yaitu mengkategorikan keterlambatan menjadi 4, yaitu ONTIME(Target=0), Common(Target<31), Medium(Target<61), Severe(Target>=61)

III. Penjelasan Source Code

In [1]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import tensorflow.keras
from tensorflow.keras.optimizers import Optimizer
import pygad.kerasga

data = pd.read_csv("Data.csv")
data.head()

y_train = data.loc[:,["target"]]
x_train = data.drop(labels =['target','ID'],axis=1)

display(data)
display(x_train)
display(y_train)
```

Kode di atas mengimport pustaka-pustaka yang diperlukan, membaca data dari file CSV, dan membagi data menjadi variabel target (y_train) dan fitur-fitur (x_train), kemudian menampilkan tampilan data.

Outputnya :

	ID	DATOP	FLTID	DEPSTN	ARRSTN	STD	STA	STATUS	AC	target
0	train_id_0	January	TU 0712	CMN	TUN	1/3/2016 10:30	2016-01-03 12.55.00	ATA	TU 32AIMN	Severe
1	train_id_1	January	TU 0757	MXP	TUN	1/13/2016 15:05	2016-01-13 16.55.00	ATA	TU 31BIMO	Common
2	train_id_2	January	TU 0214	TUN	IST	1/16/2016 4:10	2016-01-16 06.45.00	ATA	TU 32AIMN	ONTIME
3	train_id_3	January	TU 0480	DJE	NTE	1/17/2016 14:10	2016-01-17 17.00.00	ATA	TU 736IOK	ONTIME
4	train_id_4	January	TU 0338	TUN	ALG	1/17/2016 14:30	2016-01-17 15.50.00	ATA	TU 320IMU	Common
...	...	...	...	...	...	...	...	...	...	...
9995	train_id_9995	February	TU 0647	FRA	DJE	2/20/2016 19:10	2016-02-20 22.00.00	ATA	TU 736IOQ	Medium
9996	train_id_9996	February	TU 0875	GVA	TUN	2/20/2016 19:00	2016-02-20 20.55.00	ATA	TU 736IOP	Common
9997	train_id_9997	February	TU 0997	NCE	TUN	2/21/2016 10:15	2016-02-21 11.50.00	ATA	TU 320IMR	ONTIME
9998	train_id_9998	February	TU 0514	TUN	BCN	2/21/2016 14:05	2016-02-21 15.50.00	ATA	TU 32AIML	ONTIME
9999	train_id_9999	February	TU 0694	DJE	MRS	2/21/2016 14:00	2016-02-21 15.55.00	ATA	TU 32AIMN	ONTIME

10000 rows × 10 columns

	DATOP	FLTID	DEPSTN	ARRSTN	STD	STA	STATUS	AC
0	January	TU 0712	CMN	TUN	1/3/2016 10:30	2016-01-03 12.55.00	ATA	TU 32AIMN
1	January	TU 0757	MXP	TUN	1/13/2016 15:05	2016-01-13 16.55.00	ATA	TU 31BIMO
2	January	TU 0214	TUN	IST	1/16/2016 4:10	2016-01-16 06.45.00	ATA	TU 32AIMN
3	January	TU 0480	DJE	NTE	1/17/2016 14:10	2016-01-17 17.00.00	ATA	TU 736IOK
4	January	TU 0338	TUN	ALG	1/17/2016 14:30	2016-01-17 15.50.00	ATA	TU 320IMU
...	...	...	...	...	...	...	...	...
9995	February	TU 0647	FRA	DJE	2/20/2016 19:10	2016-02-20 22.00.00	ATA	TU 736IOQ
9996	February	TU 0875	GVA	TUN	2/20/2016 19:00	2016-02-20 20.55.00	ATA	TU 736IOP
9997	February	TU 0997	NCE	TUN	2/21/2016 10:15	2016-02-21 11.50.00	ATA	TU 320IMR
9998	February	TU 0514	TUN	BCN	2/21/2016 14:05	2016-02-21 15.50.00	ATA	TU 32AIML
9999	February	TU 0694	DJE	MRS	2/21/2016 14:00	2016-02-21 15.55.00	ATA	TU 32AIMN

10000 rows × 8 columns

	target
0	Severe
1	Common
2	ONTIME
3	ONTIME
4	Common
...	...
9995	Medium
9996	Common
9997	ONTIME
9998	ONTIME
9999	ONTIME

10000 rows × 1 columns

In [2]:
```
data.columns
data.info()
data.describe()
```

Kode-kode di atas digunakan untuk mendapatkan informasi tentang struktur dan karakteristik data yang dimiliki, seperti nama kolom (columns), informasi tipe data (info), dan statistik deskriptif (describe) dari dataset yang sedang digunakan.

Outputnya :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   ID      10000 non-null  object
 1   DATOP   10000 non-null  object
 2   FLTID   10000 non-null  object
 3   DEPSTN  10000 non-null  object
 4   ARRSTN  10000 non-null  object
 5   STD     10000 non-null  object
 6   STA     10000 non-null  object
 7   STATUS  10000 non-null  object
 8   AC      10000 non-null  object
 9   target  10000 non-null  object
dtypes: object(10)
memory usage: 781.4+ KB
```

Out[2]:

	ID	DATOP	FLTID	DEPSTN	ARRSTN	STD	STA	STATUS	AC	target
count	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000
unique	10000	9	718	78	80	8373	8710	5	35	4
top	train_id_0	August	TU 0217	TUN	TUN	7/4/2016 14:30	2016-07-04 09.10.00	ATA	TU 320IMU	Common
freq	1	2002	123	3702	3787	5	4	9903	503	4017

In [3]:

```
le = LabelEncoder()

cols = x_train.columns.values
for col in cols:
    x_train[col] = le.fit_transform(x_train[col])


y_train = le.fit_transform(y_train)

display(x_train)
display(y_train)
```

Kode di atas digunakan untuk melakukan proses enkoding label pada variabel target dan fitur-fitur data menggunakan objek LabelEncoder dari pustaka sklearn. Kode tersebut mengiterasi melalui setiap kolom pada x_train dan mengubah nilai-nilai fitur-fitur tersebut menjadi nilai numerik menggunakan metode fit_transform() dari LabelEncoder. Selanjutnya, variabel target y_train juga dienkoding menggunakan metode fit_transform(). Hasil enkoding ditampilkan melalui display(x_train) dan display(y_train).

Outputnya :

	DATOP	FLTID	DEPSTN	ARRSTN	STD	STA	STATUS	AC
0	3	138	19	75	486	101	0	20
1	3	162	48	75	115	358	0	4
2	3	39	73	32	199	418	0	20
3	3	83	22	57	220	474	0	24
4	3	59	73	2	222	468	0	9
...	...	...	...	...	...	...	...	...
9995	2	122	27	22	1083	1322	0	29
9996	2	178	30	75	1082	1320	0	28
9997	2	190	51	75	1101	1336	0	6
9998	2	91	73	6	1112	1344	0	18
9999	2	129	22	48	1111	1345	0	20

10000 rows × 8 columns

```
array([3, 0, 2, ..., 2, 2, 2])
```

In [4]:
```
x_train, x_test, y_train, y_test = train_test_split(x_train,y_train, test_size = 0.30, random_state = 1)

display(x_train)
display(y_train)
display(x_test)
display(y_test)
```

Kode di atas digunakan untuk membagi data menjadi data latih (training) dan data uji (testing) dengan menggunakan fungsi train_test_split dari pustaka sklearn. Data fitur (x_train) dan variabel target (y_train) akan dibagi dengan perbandingan 70:30 untuk data uji, dimana 70% data akan digunakan untuk pelatihan dan 30% data akan digunakan untuk pengujian model. Selanjutnya, hasil pemisahan tersebut ditampilkan menggunakan display() untuk melihat data latih (x_train dan y_train) serta data uji (x_test dan y_test).

Outputnya :

	DATOP	FLTID	DEPSTN	ARRSTN	STD	STA	STATUS	AC
2228	2	161	73	50	931	1165	0	6
5910	6	189	73	53	2503	1673	0	8
1950	5	154	44	22	4726	4047	0	3
2119	2	92	6	75	1291	1519	0	18
5947	6	138	19	75	2613	1836	0	10
...	...	...	...	...	...	...	...	...
2895	1	142	58	75	7393	8072	0	1
7813	1	55	73	72	7136	6912	0	24
905	2	149	73	60	1331	1571	0	17
5192	3	103	22	75	481	673	0	28
235	0	99	73	49	3302	3620	0	17

7000 rows × 8 columns

```
array([0, 0, 2, ..., 0, 2, 0])
```

	DATOP	FLTID	DEPSTN	ARRSTN	STD	STA	STATUS	AC
9953	3	115	73	12	584	155	0	28
3850	5	165	73	38	4279	3945	0	13
4962	2	148	58	75	1169	1405	0	10
3886	5	142	58	75	4687	4015	0	14
5437	4	164	14	75	5628	6226	0	13
...	...	...	...	...	...	...	...	...
5273	2	702	44	75	1022	1263	0	3
8014	1	165	73	39	6795	7496	0	11
8984	6	556	22	31	1658	1945	0	21
6498	4	10	14	1	5327	5914	0	19
6327	4	99	73	49	6426	5503	0	25

3000 rows × 8 columns

```
array([3, 0, 2, ..., 2, 0, 2])
```

In [5]:
```
for col in x_train.columns:
    print(f'Unique values in column {col}: {x_train[col].unique()}')
```

Kode di atas digunakan untuk mencetak nilai-nilai unik (unique values) dari setiap kolom pada dataframe x_train. Melalui perulangan (for loop) yang mengiterasi setiap kolom pada x_train.columns, kode tersebut mencetak pesan yang berisi nama kolom dan nilai-nilai unik yang terdapat di dalamnya.

Outputnya :

```
Unique values in column DATOP: [2 6 5 0 4 3 8 1 7]
Unique values in column FLTID: [161 189 154  92 138 164  27 160  75 153  39  62  18  81  17 111 136 150
 157 130 109 155  42 691 168 166 504 533 176  49 156 116 148 139 714 654
 100  64  87 131 147 172 120  35 167  71 184 143 312 137 162  67 578   5
 520 622 169 141 117  60 186 152 178 364 559 181 399 163 135 604 145 177
 496  30 190 121 171  99  40  50   0 188 390 128 165 498 127  59 132  46
 149  91 192 140 134  68 112 144 501  76 594  84 289 494  56 182 591 191
 645 115  72 715  43 347 101 525 142  86 118 599  55  63  61  96 564  58
 556 187 113 699  41  94  22 497  74 560 561 620 179  65 158 277 607 609
 123 680 536  66 527  70 616  77  57 174 708 545  19 236  97 129 571  44
 703  78 544 348 173 415 103 170 577 592 432 106 567  16 301   9 175 447
 270 151 122 231 159   2 555 119 375 570  28 554  45 541 412 146 717 586
 507 688 254 104  52   6  24 619 183 455 268  38 110 107 344 695 133 519
 646  26 180  82 672 608   7  15 677 124 694 383 539 652 618 634 453 426
  54 456 444 126 243 241  29 475 538 566 293 300  93  95 706 585 610 707
 611  10  37 505  79 209 335  12 185  47 427 408 535 572 449  51 334  14
 114  23 307  36  89 417 379 363 543 400 125 240 693  73  48 416 392 214
  53 551 615 676 587 353 598 573 563 509 689 710 351   8 458  83 558 673
 454 500 692 461 327 486 625 488 350 613 605  85 406 514 394 208 492 511
 712 512 284 662 499 617 355 513 638  11 108 418 105 407  80 575 705 411
 664 194 697 495 332 636 429 493 709 675 196 345 524 248 549 562 595 606
 665 659 521 701 487 397 102 282  25 667 590 281 299 658 557 361 367 279
 713  34 548 614 612 542 223  31 639 267 624 686 522 460 537 346 681 213
  88 518 469 574 452 398 526 472 582 528 484 517  13 471 534 451 434 376
  69 641 653  98 218 621 650 626 433 391 286 546 249 508 678 216 553 288
 229 523 630 324 552 303 579 272 202 644 584 370 448 234 565 275 602 326
 251 413 378 203 233 380 445 258 540 632 642 530 372 424   3 435 580 358
 597 655 316 446 588 222 199   4 489 235 381 225 576  21 387 628 640 239
 550 431 342 629 274 409 437 601 668 656 377 516 583 503 547 442 373 260
 690 682 441 313 425 205 263 414 341 393 255 291 197 529 647 278 322 438
 637 242 230 212 271 290 359 661 385 374 479 483 683 698 422 395 206 648
 276 657 269 247 491  90 349 211 315 428 478 310 687 338 700  20 257 596
 490 226 340 382   1 337 256 302 405 252 245 198 308 633  33 328 643 339
 325 716 466 663 217 627 304 696 450 320 329 420 259 635 474 220 228 368
 287 462 362 195 386 294 352 389 463 439 649 440 702 264 482 485 396 470
 317 266 237]
Unique values in column DEPSTN: [73 44  6 19 14 26 22 11 52 57 58 46 32 16 37 72 48 27 12 50 47  2 41 75
 43 55  1 33 30 51  7 38 67  0 42 61 69  4 77 56 24 39 68 23 10 17  8 54
  3 62 31 35 65 66 74 15 45 59 70 76 63 60 25  5 40 71 34 49 13 53 18 20
 36 64 29]
Unique values in column ARRSTN: [50 53 22 75 64 60 32 54 76 14 44 43 79 27 67 33 20 49 78 16 19 63 17  3
 46  8 52 48 59 24 30 39  2  6 70  0 12  7 31 41 72 11 23 73 74 10 42 38
 13 26 69 57 45 77  1 15 58 65 36 68 62 37 25 51 56 47 29 21 61 55 35  9
  4 18 71  5 28]
Unique values in column STD: [ 931 2503 4726 ... 7393  481 3302]
Unique values in column STA: [1165 1673 4047 ... 8072 6912 1571]
Unique values in column STATUS: [0 1 2 3 4]
Unique values in column AC: [ 6  8  3 18 10 11 20 17 24 14 15 19  4  5  9  2 13 29 21 31 30 23  7 25
 28 22 26 16 27 12  1 33  0 34]
```

In [6]:
```
input_layer  = tensorflow.keras.layers.Input(8)
dense_layer1 = tensorflow.keras.layers.Dense(5, activation="relu")(input_layer)
output_layer = tensorflow.keras.layers.Dense(1, activation="linear")(dense_layer1)

model = tensorflow.keras.Model(inputs=input_layer, outputs=output_layer)
```

Kode di atas digunakan untuk mendefinisikan arsitektur model neural network menggunakan library Keras dari TensorFlow. Input layer dengan dimensi 8 didefinisikan sebagai input model. Dense layer pertama dengan 5 neuron dan fungsi aktivasi ReLU diterapkan pada input layer. Kemudian, output layer dengan 1 neuron dan fungsi aktivasi linear diterapkan pada dense layer pertama. Selanjutnya, model ini dikompilasi dan siap digunakan untuk pelatihan dan prediksi.

In [7]:
```
model.summary()
```

"model.summary()" digunakan untuk mencetak ringkasan arsitektur model neural network yang telah didefinisikan sebelumnya, termasuk lapisan-lapisan dan jumlah parameter yang dimiliki oleh setiap lapisan.
Outputnya :

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 8)]               0

 dense (Dense)               (None, 5)                 45

 dense_1 (Dense)             (None, 1)                 6

=================================================================
Total params: 51
Trainable params: 51
Non-trainable params: 0
```

In [8]:
```
model.compile(optimizer='Adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Kode di atas digunakan untuk mengompilasi model neural network dengan mengatur konfigurasi optimasi, fungsi loss, dan metrik evaluasi. Dalam contoh ini, model menggunakan optimizer Adam, fungsi loss menggunakan categorical_crossentropy (untuk masalah klasifikasi multi-kelas), dan metrik evaluasi menggunakan akurasi. Dengan melakukan kompilasi model, model siap untuk dilatih dengan data latih dan dievaluasi menggunakan data uji.

In [9]:
```
keras_ga = pygad.kerasga.KerasGA(model=model, num_solutions=10)
```

Kode di atas digunakan untuk menginisialisasi objek KerasGA dari pustaka pygad. Objek ini akan digunakan untuk menerapkan algoritma genetika pada model neural network (model) menggunakan pendekatan genetik yang diberikan oleh pustaka pygad. Dalam contoh ini, objek KerasGA diinisialisasi dengan model yang ingin dioptimalkan dan jumlah solusi (num_solutions) yang akan digenerate oleh algoritma genetika.

In [10]:
```
def fitness_func(ga_instance, solution, sol_idx):
    global x_train, y_train, keras_ga, model

    model_weights_matrix = pygad.kerasga.model_weights_as_matrix(model=model,weights_vector=solution)

    model.set_weights(weights=model_weights_matrix)

    predictions = model.predict(x_train)

    mae = tensorflow.keras.losses.MeanAbsoluteError()
    error = mae(y_train, predictions).numpy()

    solution_fitness = 1.0 / (error + 0.00000001)

    return solution_fitness
```

Fungsi "fitness_func" digunakan dalam algoritma genetika untuk menghitung kecocokan solusi dengan memperbarui bobot model berdasarkan vektor solusi, melakukan prediksi menggunakan model yang diperbarui, menghitung Mean Absolute Error (MAE) antara prediksi dan nilai target, dan mengembalikan nilai fitness sebagai invers dari error tersebut.

In [11]:
```
def on_generation(ga_instance):
    print("Generation = {generation}".format(generation=ga_instance.generations_completed))
    print("Fitness    = {fitness}".format(fitness=ga_instance.best_solution(ga_instance.last_generation_fitness)[1]), end='
```

Fungsi "on_generation" adalah sebuah callback dalam algoritma genetika yang mencetak informasi tentang nomor generasi saat ini dan kecocokan terbaik yang ditemukan dalam generasi tersebut. Hal ini memungkinkan pemantauan perkembangan algoritma genetika dan menampilkan nilai kecocokan terbaik pada setiap generasi.

In [12]:
```
ga_instance = pygad.GA(num_generations=100,
                       num_parents_mating=5,
                       fitness_func=fitness_func,
                       initial_population=keras_ga.population_weights,
                       on_generation=on_generation,
                       suppress_warnings=True)
```

Kode di atas digunakan untuk membuat sebuah instance objek algoritma genetika (GA) dengan konfigurasi tertentu, termasuk jumlah generasi, jumlah induk yang akan dipilih, fungsi fitness, populasi awal, callback pada setiap generasi, dan kontrol peringatan.

In [13]:
```
ga_instance.run()
```

Kode "ga_instance.run()" digunakan untuk menjalankan algoritma genetika yang telah didefinisikan sebelumnya, yang akan melaksanakan proses evolusi populasi berdasarkan konfigurasi yang telah ditentukan seperti jumlah generasi, fungsi fitness, dan lainnya.

Outputnya :

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
```

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 1
Fitness    = 0.010276359041782408

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 2
Fitness    = 0.18663003478222775

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 3
Fitness    = 0.406338378896827

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 4
Fitness    = 0.406338378896827

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
```

```
219/219 [==============================] - 0s 1ms/step
Generation = 5
Fitness    = 0.406338378896827

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 6
Fitness    = 0.406338378896827

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 7
Fitness    = 0.45359018414762575

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 8
Fitness    = 0.5717593926237851

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
Generation = 9
Fitness    = 0.5717593926237851

219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
```

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 10
Fitness    = 0.5717593926237851

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 11
Fitness    = 0.6116939821336314

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 12
Fitness    = 0.6500507569675106

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 13
Fitness    = 0.6653820001183223

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 14
Fitness    = 0.8200714793990468
```

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 987us/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 15
Fitness    = 0.8438402944991944

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 16
Fitness    = 0.8438402944991944

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 17
Fitness    = 0.9186114206992668

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 18
Fitness    = 0.9469695862323257

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
```

```
219/219 [==============================] - 0s 1ms/step
Generation = 19
Fitness    = 0.9469695862323257

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 20
Fitness    = 0.9600796848977041

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 21
Fitness    = 1.0048533746553676

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 22
Fitness    = 1.0048533746553676

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 23
Fitness    = 1.0053929183617345

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
```

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 24
Fitness    = 1.0059408826654113

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 25
Fitness    = 1.0061558307767493

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 26
Fitness    = 1.0061558307767493

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 27
Fitness    = 1.006555384709736

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 28
Fitness    = 1.006555384709736
```

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 29
Fitness    = 1.0065923439321554

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 30
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 31
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 32
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
```

```
219/219 [==============================] - 0s 1ms/step
Generation = 33
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 34
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 35
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 2ms/step
Generation = 36
Fitness    = 1.0065927666840593

219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
Generation = 37
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 1s 4ms/step
219/219 [==============================] - 1s 2ms/step
```

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 38
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 39
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 40
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 41
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 42
Fitness    = 1.0065927666840593
```

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 43
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 44
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 45
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 46
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
```

```
219/219 [==============================] - 0s 1ms/step
Generation = 47
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 48
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 2ms/step
Generation = 49
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 3ms/step
Generation = 50
Fitness    = 1.0065927666840593

219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 51
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
```

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 52
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 53
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 54
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 55
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 56
Fitness    = 1.0065927666840593
```

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 57
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 3ms/step
Generation = 58
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 2s 7ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 59
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 1s 2ms/step
Generation = 60
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 3ms/step
```

```
219/219 [==============================] - 0s 2ms/step
Generation = 61
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 62
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 2ms/step
Generation = 63
Fitness    = 1.0065927666840593

219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
Generation = 64
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 65
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
```

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 66
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 2ms/step
Generation = 67
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
Generation = 68
Fitness    = 1.0065927666840593

219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 69
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 70
Fitness    = 1.0065927666840593
```

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 71
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 72
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 73
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
Generation = 74
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
```

```
219/219 [==============================] - 0s 1ms/step
Generation = 75
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 2ms/step
Generation = 76
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 77
Fitness    = 1.0065927666840593

219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 78
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 79
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
```

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 80
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 81
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 82
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 83
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 84
Fitness    = 1.0065927666840593
```

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 85
Fitness    = 1.0065927666840593

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 86
Fitness    = 1.0116245074613701

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 87
Fitness    = 1.0116245074613701

219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 88
Fitness    = 1.0116245074613701

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
```

```
219/219 [==============================] - 0s 1ms/step
Generation = 89
Fitness    = 1.0116245074613701

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 90
Fitness    = 1.0116245074613701

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 91
Fitness    = 1.0116245074613701

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 92
Fitness    = 1.0116245074613701

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 93
Fitness    = 1.0116245074613701

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
```

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 94
Fitness    = 1.0116245074613701

219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 1s 2ms/step
Generation = 95
Fitness    = 1.0116245074613701

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 96
Fitness    = 1.0116245074613701

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 97
Fitness    = 1.0116245074613701

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 98
Fitness    = 1.0116245074613701
```

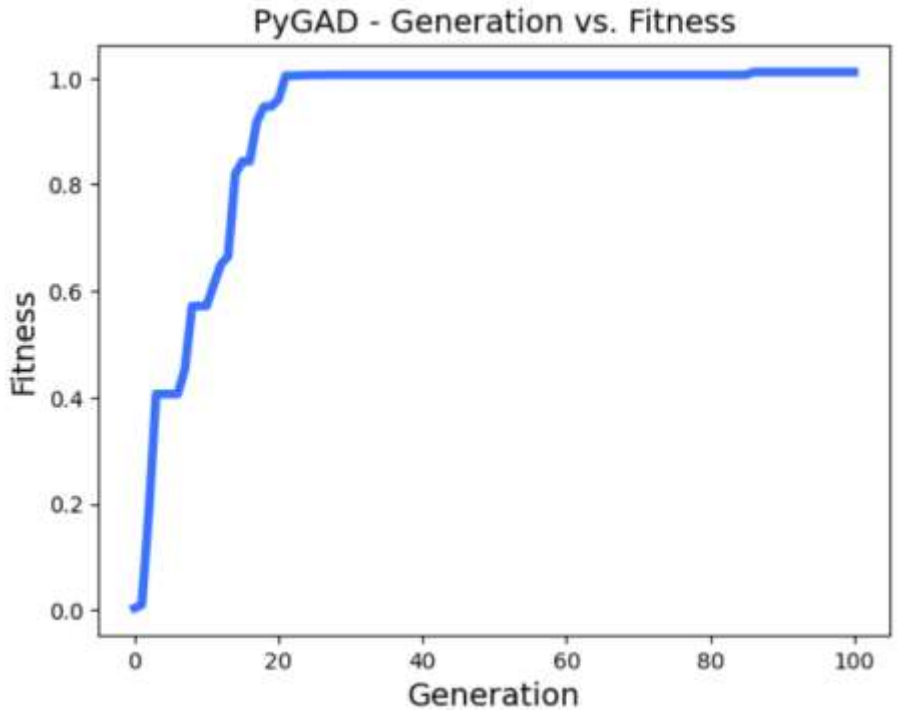
```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 2ms/step
Generation = 99
Fitness    = 1.0116245074613701

219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Generation = 100
Fitness    = 1.0116245074613701
```

In [14]:
```
fig = ga_instance.plot_result(linewidth=4)
```

Kode "ga_instance.plot_result(linewidth=4)" digunakan untuk menghasilkan sebuah grafik visualisasi hasil dari algoritma genetika, di mana grafik tersebut menampilkan perubahan nilai kecocokan terbaik dalam setiap generasi yang dievaluasi.

Outputnya :



In [15]:
```
solution, solution_fitness, _ = ga_instance.best_solution()

print("Fitness value of the best solution:\n{solution_fitness}".format(solution_fitness=solution_fitness), end='\n\n')
```

Kode di atas digunakan untuk mendapatkan solusi terbaik dari algoritma genetika yang telah dievaluasi, di mana nilai solusi terbaik disimpan dalam variabel "solution" dan nilai kecocokan solusi terbaik disimpan dalam variabel "solution_fitness". Selanjutnya, nilai kecocokan solusi terbaik tersebut dicetak.

Outputnya :

```
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 1s 3ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
219/219 [==============================] - 0s 1ms/step
Fitness value of the best solution:
1.0116245074613701
```

In [16]:
```
best_solution_weights = pygad.kerasga.model_weights_as_matrix(model=model,
                                                              weights_vector=solution)
model.set_weights(best_solution_weights)
predictions = model.predict(x_test)
print("Predictions:\n", predictions, end='\n\n')

print("Correct Outputs:\n", y_test, end='\n\n')
```

Kode di atas digunakan untuk mendapatkan bobot terbaik dari solusi terbaik yang dihasilkan oleh algoritma genetika. Bobot tersebut kemudian diterapkan pada model dengan menggunakan fungsi model_weights_as_matrix dari pustaka pygad. Selanjutnya, model yang telah diperbarui digunakan untuk melakukan prediksi terhadap data uji (x_test) dan hasil prediksi dicetak. Selain itu, nilai target yang sebenarnya (y_test) juga dicetak.

Outputnya :

```
94/94 [==============================] - 0s 1ms/step
Predictions:
 [[0.9945631]
 [0.9945631]
 [0.9945631]
 ...
 [0.9945631]
 [0.9945631]
 [0.9945631]]

Correct Outputs:
 [3 0 2 ... 2 0 2]
```

In [17]:
```
mae = tensorflow.keras.losses.MeanAbsoluteError()
abs_error = mae(y_test, predictions).numpy()
print("Absolute Error:\n", abs_error)
```

Kode di atas digunakan untuk menghitung Mean Absolute Error (MAE) antara nilai target yang sebenarnya (y_test) dengan hasil prediksi (predictions) menggunakan objek MeanAbsoluteError dari TensorFlow. Hasil MAE tersebut kemudian dicetak sebagai Absolute Error.

Outputnya :

```
Absolute Error:
 0.9877165
```

In [18]:
```
score = model.evaluate(x_train, y_train, verbose=0)
print('Test accuracy:', score[1])
```

Kode di atas digunakan untuk mengevaluasi model dengan menggunakan data latih (x_train dan y_train) dan menghitung akurasi uji model. Hasil akurasi uji tersebut kemudian dicetak.

Outputnya :

```
Test accuracy: 0.1525714248418808
```