

praktikum-1-4

October 26, 2024

Nama : Elis Nurhidayati

NIM : 2241720035

Kelas : TI - 3C

Link Google Colab : [https://colab.research.google.com/drive/1Ie3LNI2ZFgGB0UEQU5xf2zSq1iPnGGkL?usp=](https://colab.research.google.com/drive/1Ie3LNI2ZFgGB0UEQU5xf2zSq1iPnGGkL?usp=sharing)

##Praktikum 1 (Pra Pengolahan Data - Data Terstruktur)##

##Data Imputation## Pada praktikum ini kita akan mencoba melakukan data imputation terhadap data yang hilang (missing value). Dataset yang digunakan adalah dataset Titanic.

#Langkah 1 - Load Data

```
[ ]: #Upload dataset yang akan digunakan
from google.colab import files
uploaded = files.upload()
```

<IPython.core.display.HTML object>

Saving Titanic-Dataset.csv to Titanic-Dataset.csv

```
[ ]: #Import library yang dibutuhkan kemudian, load data kedalam DataFrame Pandas
import pandas as pd
import io

df = pd.read_csv(io.BytesIO(uploaded['Titanic-Dataset.csv'])) # load dataset
```

```
[ ]: #Menampilkan data terbatas
df.head()
```

```
[ ]: PassengerId  Survived  Pclass  \
0             1         0         3
1             2         1         1
2             3         1         3
3             4         1         1
4             5         0         3
```

```
                                Name      Sex  Age  SibSp  \
0                Braund, Mr. Owen Harris   male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th... female  38.0      1
```

2		Heikkinen, Miss. Laina	female	26.0	0
3		Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1
4		Allen, Mr. William Henry	male	35.0	0

	Parch	Ticket	Fare	Cabin	Embarked
0	0	A/5 21171	7.2500	NaN	S
1	0	PC 17599	71.2833	C85	C
2	0	STON/O2. 3101282	7.9250	NaN	S
3	0	113803	53.1000	C123	S
4	0	373450	8.0500	NaN	S

#Langkah 2 - Pengecekan Data

```
[ ]: #Pengecekan informasi data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
#   Column          Non-Null Count  Dtype
---  -
0   PassengerId     891 non-null    int64
1   Survived        891 non-null    int64
2   Pclass          891 non-null    int64
3   Name            891 non-null    object
4   Sex             891 non-null    object
5   Age             714 non-null    float64
6   SibSp           891 non-null    int64
7   Parch           891 non-null    int64
8   Ticket          891 non-null    object
9   Fare            891 non-null    float64
10  Cabin           204 non-null    object
11  Embarked        889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

Dari df.info() kita dapat mengetahui jumlah data untuk setiap kolom. Selain itu, kita mengetahui bahwa,

- “Age” berjenis float64
- “Cabin” berjenis object
- “Embarked” berjenis object

Pada Pandas, object sering kali mewaliki tipe data string. Kita harus cermat dalam hal ini. Hal ini berkaitan dengan teknik imputasi yang akan kita gunakan nantinya.

```
[ ]: #Jumlahh data yang hilang pada tiap kolom
df.isnull().sum()
```

```
[ ]: PassengerId      0
      Survived        0
      Pclass          0
      Name            0
      Sex             0
      Age            177
      SibSp           0
      Parch           0
      Ticket          0
      Fare            0
      Cabin           687
      Embarked        2
      dtype: int64
```

#Langkah 3 - Imputasi

Pada langkah ini kita akan melakukan imputasi terhadap data yang hilang pada “Age”, “Cabin”, dan “Embarked”.

Strategi yang akan kita gunakan adalah,

- “Age” → Dikarenakan “Age” adalah data nominal, maka kita akan menggunakan strategi mean.
- “Cabin” → “Cabin” merupakan informasi terkait dengan nomor kabin penumpang. Disini kita akan berasumsi bahwa, seluruh penumpang yang tidak memiliki nomor kabin, merupakan penumpang yang tinggal di dek-dek kapal. Sehingga kita akan mensubstitusi data yang hilang dengan informasi “DECK”.
- “Embarked” → “Embarked” merupakan informasi lokasi embarkasi penumpang. Hanya ada 2 data yang hilang disini. Dikarenakan data merupakan data nominal, maka kita dapat menggunakan modus (mode) untuk mensubstitusi data yang hilang.

```
[ ]: # Age - mean
      df['Age'].fillna(value=df['Age'].mean(), inplace=True)

      # Cabin - "DECK"
      df['Cabin'].fillna(value="DECK", inplace=True)

      # Embarked - modus
      df['Embarked'].fillna(value=df['Embarked'].mode, inplace=True)
```

#Langkah 4 - Validasi Hasil

Lakukan pengecekan kembali dengan df.info(). Selanjutnya, kita juga dapat memastikan kembali apakah nilai yang disajikan sudah benar dengan melakukan pengecekan data secara langsung. Hasil dari proses imputasi adalah sebagai berikut

```
[ ]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
```

Data columns (total 12 columns):

#	Column	Non-Null Count	Dtype
0	PassengerId	891 non-null	int64
1	Survived	891 non-null	int64
2	Pclass	891 non-null	int64
3	Name	891 non-null	object
4	Sex	891 non-null	object
5	Age	891 non-null	float64
6	SibSp	891 non-null	int64
7	Parch	891 non-null	int64
8	Ticket	891 non-null	object
9	Fare	891 non-null	float64
10	Cabin	891 non-null	object
11	Embarked	891 non-null	object

dtypes: float64(2), int64(5), object(5)

memory usage: 83.7+ KB

```
[ ]: #Hasil Proses Validasi
df.head(10)
```

```
[ ]: PassengerId  Survived  Pclass  \
0             1         0         3
1             2         1         1
2             3         1         3
3             4         1         1
4             5         0         3
5             6         0         3
6             7         0         1
7             8         0         3
8             9         1         3
9            10         1         2
```

	Name	Sex	Age
0	Braund, Mr. Owen Harris	male	22.000000
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000
2	Heikkinen, Miss. Laina	female	26.000000
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000
4	Allen, Mr. William Henry	male	35.000000
5	Moran, Mr. James	male	29.699118
6	McCarthy, Mr. Timothy J	male	54.000000
7	Palsson, Master. Gosta Leonard	male	2.000000
8	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.000000
9	Nasser, Mrs. Nicholas (Adele Achem)	female	14.000000

	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	A/5 21171	7.2500	DECK	S

1	1	0	PC 17599	71.2833	C85	C
2	0	0	STON/O2.	3101282	7.9250	DECK
3	1	0	113803	53.1000	C123	S
4	0	0	373450	8.0500	DECK	S
5	0	0	330877	8.4583	DECK	Q
6	0	0	17463	51.8625	E46	S
7	3	1	349909	21.0750	DECK	S
8	0	2	347742	11.1333	DECK	S
9	1	0	237736	30.0708	DECK	C

1 Praktikum 2 (Seleksi Fitur, Encoding, dan Standarisasi)

##Seleksi Fitur## Pada tahap ini, kita memilih variabel yang akan digunakan sebagai fitur dalam model pembelajaran mesin. Tidak semua variabel harus digunakan, sehingga kita memilih yang penting untuk tujuan model. Dalam kasus penumpang Titanic yang selamat, fitur yang digunakan adalah “Survived,” “Pclass,” “Sex,” “Age,” dan “Cabin.” Variabel-variabel ini dipisahkan dari variabel lainnya, dan data yang digunakan adalah data Titanic yang telah melalui proses imputasi.

2 Langkah 0 - Import Library

```
[ ]: import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler
```

3 Langkah 1 - Load Data

```
[ ]: dpath = 'Titanic-Dataset-fixed.csv'
df = pd.read_csv(dpath)
df.head()
```

```
[ ]: PassengerId  Survived  Pclass  \
0              1         0         3
1              2         1         1
2              3         1         3
3              4         1         1
4              5         0         3
```

```

                                Name      Sex  Age  SibSp  \
0                Braund, Mr. Owen Harris   male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                Heikkinen, Miss. Laina   female  26.0      0
3  Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                Allen, Mr. William Henry   male  35.0      0
```

```
Parch      Ticket      Fare Cabin Embarked
```

0	0	A/5	21171	7.2500	DECK	S
1	0	PC	17599	71.2833	C85	C
2	0	STON/O2.	3101282	7.9250	DECK	S
3	0		113803	53.1000	C123	S
4	0		373450	8.0500	DECK	S

4 Langkah 2 - Slice Data

```
[ ]: df = df[['Survived', 'Pclass', 'Age', 'Sex', 'Cabin']]
df.head()
```

```
[ ]:   Survived  Pclass   Age    Sex Cabin
0         0        3  22.0   male  DECK
1         1        1  38.0 female  C85
2         1        3  26.0 female  DECK
3         1        1  35.0 female  C123
4         0        3  35.0   male  DECK
```

5 Langkah 3 - Encoding

Encoding pada “Sex” dan “Cabin”

```
[ ]: le = LabelEncoder() # membuat objek dari LabelEncoder
df['Sex'] = le.fit_transform(df['Sex']) # proses encoding
df['Cabin'] = le.fit_transform(df['Cabin']) # proses encoding
```

6 Langkah 4 - Verifikasi Hasil

```
[ ]: #Hasil Encoding
df.head()
```

```
[ ]:   Survived  Pclass   Age  Sex  Cabin
0         0        3  22.0    1   115
1         1        1  38.0    0    81
2         1        3  26.0    0   115
3         1        1  35.0    0    55
4         0        3  35.0    1   115
```

7 Langkah 5 - Standarisasi

```
[ ]: std = StandardScaler()
df['Age'] = std.fit_transform(df[['Age']])
```

8 Langkah 6 - Verifikasi Hasil Standarisasi

```
[ ]: #Hasil Standarisasi
df.head()
```

```
[ ]:   Survived  Pclass      Age  Sex  Cabin
0         0        3 -0.592481    1   115
1         1        1  0.638789    0    81
2         1        3 -0.284663    0   115
3         1        1  0.407926    0    55
4         0        3  0.407926    1   115
```

9 Praktikum 3 (Spliting Data)

Pada pratikum ini, kita akan mencoba 3 macam strategi pembuatan data latih, validasi, dan uji. Data yang digunakan adalah data kapal Titanic yang sudah kita ekstraksi fiturnya pada pratikum sebelumnya

10 Random Split

10.1 Langkah 1 - Load Data

```
[ ]: import pandas as pd

df = pd.read_csv('Titanic-Dataset-selected.csv')
df.head()
```

```
[ ]:   Survived  Pclass      Age  Sex  Cabin
0         0        3 -0.592481    1   115
1         1        1  0.638789    0    81
2         1        3 -0.284663    0   115
3         1        1  0.407926    0    55
4         0        3  0.407926    1   115
```

10.2 Langkah 2 - Split Data

```
[ ]: # Split data
from sklearn.model_selection import train_test_split

# Split data training dan dan lainnya
# data lainnya, akan kita split lagi menjadi validasi dan testing.
# Rasio yang akan kita gunakan adalah 8:1:1
df_train, df_unseen = train_test_split(df, test_size=0.2, random_state=0)

# Split lagi antara validasi dan testing
df_val, df_test = train_test_split(df_unseen, test_size=0.5, random_state=0)
```

```

# Cek masing-masing ukuran data

print(f'Jumlah data asli: {df.shape[0]}')
print(f'Jumlah data train: {df_train.shape[0]}')
print(f'Jumlah data val: {df_val.shape[0]}')
print(f'Jumlah data test: {df_test.shape[0]}')

# Cek rasio tiap label
print('=====')
print(f'Jumlah label data asli:\n{df.Survived.value_counts()}')
print(f'Jumlah label data train:\n{df_train.Survived.value_counts()}')
print(f'Jumlah label data val:\n{df_val.Survived.value_counts()}')
print(f'Jumlah label data test:\n{df_test.Survived.value_counts()}')

```

```

Jumlah data asli: 891
Jumlah data train: 712
Jumlah data val: 89
Jumlah data test: 90
=====
Jumlah label data asli:
Survived
0    549
1    342
Name: count, dtype: int64
Jumlah label data train:
Survived
0    439
1    273
Name: count, dtype: int64
Jumlah label data val:
Survived
0     53
1     36
Name: count, dtype: int64
Jumlah label data test:
Survived
0     57
1     33
Name: count, dtype: int64

```


11 Stratified Split

11.1 Langkah 1 - Load Data

```
[ ]: import pandas as pd

df2 = pd.read_csv('Titanic-Dataset-selected.csv')
df2.head()
```

```
[ ]:      Survived  Pclass      Age  Sex  Cabin
0          0         3 -0.592481    1   115
1          1         1  0.638789    0    81
2          1         3 -0.284663    0   115
3          1         1  0.407926    0    55
4          0         3  0.407926    1   115
```

11.2 Langkah 2 - Split Data

```
[ ]: # Split data
from sklearn.model_selection import train_test_split

# Split data training dan dan lainnya
# data lainnya, akan kita split lagi menjadi validasi dan testing.
# Rasio yang akan kita gunakan adalah 8:1:1
df2_train, df2_unseen = train_test_split(df2, test_size=0.2, random_state=0,
    ↪stratify=df['Survived'])

# Split lagi antara validasi dan testing
df2_val, df2_test = train_test_split(df2_unseen, test_size=0.5, random_state=0,
    ↪stratify=df_unseen['Survived'])

# Cek masing-masing ukuran data

print(f'Jumlah label data asli:\n{df2.Survived.value_counts()}')
print(f'Jumlah label data train:\n{df2_train.Survived.value_counts()}')
print(f'Jumlah label data val:\n{df2_val.Survived.value_counts()}')
print(f'Jumlah label data test:\n{df2_test.Survived.value_counts()}')
```

```
Jumlah label data asli:
Survived
0      549
1      342
Name: count, dtype: int64
Jumlah label data train:
Survived
0      439
1      273
```

```
Name: count, dtype: int64
Jumlah label data val:
Survived
0      51
1      38
Name: count, dtype: int64
Jumlah label data test:
Survived
0      59
1      31
Name: count, dtype: int64
```

12 Cross Validation 1

12.1 Langkah 1 - Load Data

```
[ ]: import pandas as pd

df3 = pd.read_csv('Titanic-Dataset-selected.csv')
df3.head()
```

```
[ ]:      Survived  Pclass      Age  Sex  Cabin
0          0         3 -0.592481    1    115
1          1         1  0.638789    0     81
2          1         3 -0.284663    0    115
3          1         1  0.407926    0     55
4          0         3  0.407926    1    115
```

12.2 Langkah 2 - Split Data

Jika berhasil, maka seharusnya setiap fold memiliki nomor index yang berbeda untuk setiap data latih dan data uji.

Berikut hasilnya untuk 2 fold pertama.

```
[ ]: # Implementasi k-fold cross validation (random) dengan training dan testing saja
from sklearn.model_selection import KFold

# inisiasi obyek kfold
kf = KFold(n_splits=4)
print(f'Jumlah fold: {kf.get_n_splits()}')
print(f'Obyek KFold: {kf}')

# Lakukan splitting dengan KFold
kf_split = kf.split(df3)
print(f'Jumlah data df: {df.shape[0]}')

# cek index data tiap fold
```

```
for train_index, test_index in kf_split:
    print(f'Index train: {train_index} | Index test: {test_index}')
```

Jumlah fold: 4

Obyek KFold: KFold(n_splits=4, random_state=None, shuffle=False)

Jumlah data df: 891

Index train: [223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238

239 240

241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258
259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276
277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294
295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312
313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330
331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348
349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366
367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384
385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402
403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420
421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438
439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456
457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474
475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492
493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510
511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528
529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546
547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564
565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582
583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600
601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618
619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636
637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654
655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672
673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690
691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708
709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726
727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744
745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762
763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780
781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798
799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816
817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834
835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852
853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870
871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888
889 890] | Index test: [0 1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35

```

36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107
108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179
180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197
198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
216 217 218 219 220 221 222]
Index train: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
16 17
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107
108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179
180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197
198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
216 217 218 219 220 221 222 446 447 448 449 450 451 452 453 454 455 456
457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474
475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492
493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510
511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528
529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546
547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564
565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582
583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600
601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618
619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636
637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654
655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672
673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690
691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708
709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726
727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744
745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762
763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780
781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798
799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816
817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834
835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852
853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870

```

871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888
 889 890] | Index test: [223 224 225 226 227 228 229 230 231 232 233 234 235 236
 237 238 239 240
 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258
 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276
 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294
 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312
 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330
 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348
 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366
 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384
 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402
 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420
 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438
 439 440 441 442 443 444 445]
 Index train: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 16 17
 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107
 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179
 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197
 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233
 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251
 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269
 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287
 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305
 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323
 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341
 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359
 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377
 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395
 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413
 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431
 432 433 434 435 436 437 438 439 440 441 442 443 444 445 669 670 671 672
 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690
 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708
 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726
 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744
 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762
 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780
 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798

799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816
 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834
 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852
 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870
 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888
 889 890] | Index test: [446 447 448 449 450 451 452 453 454 455 456 457 458 459
 460 461 462 463
 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481
 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499
 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517
 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535
 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553
 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571
 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589
 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607
 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625
 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643
 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661
 662 663 664 665 666 667 668]
 Index train: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 16 17
 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107
 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179
 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197
 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233
 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251
 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269
 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287
 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305
 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323
 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341
 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359
 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377
 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395
 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413
 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431
 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449
 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467
 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485
 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503

```

504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521
522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539
540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557
558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575
576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593
594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611
612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629
630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647
648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665
666 667 668] | Index test: [669 670 671 672 673 674 675 676 677 678 679 680 681
682 683 684 685 686
687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704
705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722
723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740
741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758
759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776
777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794
795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812
813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830
831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848
849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866
867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884
885 886 887 888 889 890]

```

13 Cross Validation 2

13.1 Langkah 1 - Load Data

```

[ ]: import pandas as pd

df4 = pd.read_csv('Titanic-Dataset-selected.csv')
df4.head()

```

```

[ ]:
Survived  Pclass      Age  Sex  Cabin
0         0       3 -0.592481   1   115
1         1       1  0.638789   0    81
2         1       3 -0.284663   0   115
3         1       1  0.407926   0    55
4         0       3  0.407926   1   115

```

13.2 Langkah 2 - Split Data

Hasilnya, pada jumlah data latih seharusnya berkurang karena telah kita split pada tahap awal. Berikut merupakan hasil untuk 2 fold pertama

```

[ ]:

```

```

# Implementasi k-fold cross validation (random) dengan training, validation,
↳ dan testing data
from sklearn.model_selection import train_test_split, KFold

# Split dulu antara data training dan testing dengan train_test_split
# Rasio 8:2 untuk training dan testing
df4_train, df4_test = train_test_split(df4, test_size=0.2, random_state=0)

# inisiasi obyek kfold
kf2 = KFold(n_splits=4)
print(f'Jumlah fold: {kf2.get_n_splits()}')
print(f'Obyek KFold: {kf2}')

# Lakukan splitting dengan KFold untuk data df_training
# Dengan acara ini, kita masih memiliki data testing untuk keperluan pengujian
↳ model
# namun tetap dapat melakukan evaluasi dengan menggunakan data validasi
kf2_split = kf2.split(df_train)
print(f'Jumlah data df_train: {df4_train.shape[0]}')

# cek index data tiap fold
for train_index, test_index in kf2_split:
    print(f'Index train: {train_index} | Index test: {test_index}')

```

```

Jumlah fold: 4
Obyek KFold: KFold(n_splits=4, random_state=None, shuffle=False)
Jumlah data df_train: 712
Index train: [178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193
194 195
196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213
214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231
232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249
250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267
268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285
286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303
304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321
322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339
340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357
358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375
376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393
394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411
412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429
430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447
448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465
466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483
484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501
502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519

```



```

520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537
538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555
556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573
574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591
592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609
610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627
628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645
646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663
664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681
682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699
700 701 702 703 704 705 706 707 708 709 710 711] | Index test: [ 0 1 2 3
4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107
108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177]
Index train: [ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107
108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 356 357
358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375
376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393
394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411
412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429
430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447
448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465
466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483
484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501
502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519
520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537
538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555
556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573
574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591
592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609
610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627
628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645

```

646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663
 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681
 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699
 700 701 702 703 704 705 706 707 708 709 710 711] | Index test: [178 179 180 181
 182 183 184 185 186 187 188 189 190 191 192 193 194 195
 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213
 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231
 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249
 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267
 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285
 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303
 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321
 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339
 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355]
 Index train: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 16 17
 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107
 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179
 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197
 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233
 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251
 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269
 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287
 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305
 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323
 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341
 342 343 344 345 346 347 348 349 350 351 352 353 354 355 534 535 536 537
 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555
 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573
 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591
 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609
 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627
 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645
 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663
 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681
 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699
 700 701 702 703 704 705 706 707 708 709 710 711] | Index test: [356 357 358 359
 360 361 362 363 364 365 366 367 368 369 370 371 372 373
 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391
 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409

```

410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427
428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445
446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463
464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481
482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499
500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517
518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533]
Index train: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
16 17
 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107
108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179
180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197
198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233
234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251
252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269
270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287
288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305
306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323
324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341
342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359
360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377
378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395
396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413
414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431
432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449
450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467
468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485
486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503
504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521
522 523 524 525 526 527 528 529 530 531 532 533] | Index test: [534 535 536 537
538 539 540 541 542 543 544 545 546 547 548 549 550 551
552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569
570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587
588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605
606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623
624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641
642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659
660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677
678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695
696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711]

```

14 Praktikum 4 (Ekstraksi Fitur Data Tidak Terstruktur)

Pada pratikum ini, kita akan mencoba untuk mengekstraksi fitur dari sebuah citra. Proses yang akan kita lakukan hanya proses sederhana. Pembahasan lengkap terkait dengan pra pengolahan data dan ekstraksi fitur pada data citra akan dibahas pada perkuliahan pengolahan citra digital.

15 Langkah 0 - Instal Pillow

```
[ ]: !pip install Pillow
```

Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (9.4.0)

16 Langkah 1 - Load Image

```
[ ]: from PIL import Image  
  
img = Image.open('Lenna.png')  
img.show() # tampilkan gambar  
display(img) # metode alternatif tampilkan gambar
```



17 Langkah 2 - Ekstrak Fitur

```
[ ]: # Ekstrak setiap channel red, green, blue  
r, g, b = img.split()  
  
# Cek panjang ukuran channel red  
print(len(r.histogram()))  
  
# fitur histogram pada channel red  
print(r.histogram())
```

[illegible]