**Implementasi Algoritma K-Nearest Neighbor**

**(Studi kasus Financial Distress pada Bank Konvensional tahun 2008 - 2010)**

*“Diajukan untuk memenuhi tugas besar mata kuliah Kecerdasan Buatan”*

Dosen : **Bambang Siswoyo, Ir., M.Si.**

****

**OLEH:**

Adhitya Kuncoro Jatti

10116472

Kelas : AI - 11

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSITAS KOMPUTER INDONESIA**

**2020**

# DAFTAR ISI

[DAFTAR ISI ii](#_Toc31710273)

[Kata Pengantar iii](#_Toc31710274)

[Abstrak iv](#_Toc31710275)

[Pendahuluan 1](#_Toc31710276)

[Metode Penelitian 1](#_Toc31710277)

[K-Nearset Neighbor (KNN) 1](#_Toc31710278)

[Hasil dan Pembahasan 3](#_Toc31710279)

[Kesimpulan 12](#_Toc31710280)

[Daftar Isi 13](#_Toc31710281)

# Daftar Gambar

[Gambar 1 Rumus Euclidean 2](file:///C:\Users\RadenZX\Desktop\10116472%20-%20Adhitya%20Kuncoro%20Jatti%20-%20Tugas%20Besar.docx#_Toc31710313)

[Gambar 2 Proses Metode K-Nearest Neighbor 2](file:///C:\Users\RadenZX\Desktop\10116472%20-%20Adhitya%20Kuncoro%20Jatti%20-%20Tugas%20Besar.docx#_Toc31710314)

[Gambar 3 Data Bank 4](file:///C:\Users\RadenZX\Desktop\10116472%20-%20Adhitya%20Kuncoro%20Jatti%20-%20Tugas%20Besar.docx#_Toc31710315)

[Gambar 4 Skema Program 4](file:///C:\Users\RadenZX\Desktop\10116472%20-%20Adhitya%20Kuncoro%20Jatti%20-%20Tugas%20Besar.docx#_Toc31710316)

[Gambar 5 Hasil tampil data 11](file:///C:\Users\RadenZX\Desktop\10116472%20-%20Adhitya%20Kuncoro%20Jatti%20-%20Tugas%20Besar.docx#_Toc31710317)

[Gambar 6 Hasil mencari nilai K 11](file:///C:\Users\RadenZX\Desktop\10116472%20-%20Adhitya%20Kuncoro%20Jatti%20-%20Tugas%20Besar.docx#_Toc31710318)

[Gambar 7 hasil train 12](file:///C:\Users\RadenZX\Desktop\10116472%20-%20Adhitya%20Kuncoro%20Jatti%20-%20Tugas%20Besar.docx#_Toc31710319)

[Gambar 8 hasil prediksi 12](file:///C:\Users\RadenZX\Desktop\10116472%20-%20Adhitya%20Kuncoro%20Jatti%20-%20Tugas%20Besar.docx#_Toc31710320)

# Kata Pengantar

Puji dan Syukur kita panjatkan ke hadirat Tuhan yang Maha Esa. Karena atas berkat dan hidayah-Nya saya bisa menyelesaikan tugas ini. Tugas ini dikerjakan untuk memenuhi salah satu tugas mata kuliah Kecerdasan Buatan.

Didalam tugas ini terdapat banyak kekurangan, untuk itu saya ingin mengucapkan permohonan maaf yang sebesar-besarnya. Kritik dan saran akan saya terima sebagai suatu masukan yang baik untuk kedepannya. Tidak lupa saya sampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu terselesaikannya tugas ini, mudah-mudahan semua bantuannya di berikan balasan yang terbaik oleh Tuhan Yang Maha Esa.

Untuk itu, sekali lagi saya ucapkan maaf yang sebesar-besarnya, mudah-mudahan tugas ini bermanfaat bagi kita semua. Terima kasih.

Bandung, Februari 2020

Adhitya Kuncoro Jatti

10116472

# Abstrak

Penelitian ini bertujuan untuk menghitung rasio kesulitan keuangan (financial distress) pada bank konvensional dengan rentan waktu antara tahun 2008 hingga 2010 mengguanakan metode K-nearest neighbor (KNN) dimana metode ini digunakan untuk menentukan class dari suatu bank dengan mentukan apakah bank tersebut adalah class bangkrut atau grey area atau non bangkrut. Hasil dari pengujian menggunakan metode K-nearest neighbor menghasilkan nilai akurasi sebesar 100%.

# Pendahuluan

Kondisi perekonomian Indonesia pasca krisis yang masih belum menentu mengakibatkan tingginya risiko suatu perusahaan mengalami kesulitan keuangan (financial distress) atau bahkan kebangkrutan. Kondisi kesulitan keuangan menurut teori-teori yang telah ada (financial distress) terjadi sebelum mendekati kebangkrutan, sehingga banyak sekali model financial distress perlu dikembangkan karena dengan mengetahui kondisi kesulitan keuangan sejak dini diharapkan dapat dilakukan kebijakan untuk mengantisipasinya. Salah satu model prediksi kesulitan keuangan dapat menggunakan analisis rasio keuangan dalam laporan keuangan dan mencerminkan kinerja keuangan suatu perusahaan[1].

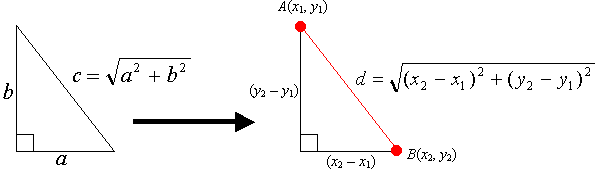
Kesulitan keuangan dan tanda-tanda awal kebangkrutan dapat diketahui melalui analisis terhadap data yang terdapat dalam laporan keuangan. Laporan keuangan yang diterbitkan oleh perusahaan merupakan salah satu sumber informasi mengenai posisi keuangan perusahaan, kinerja serta perubahan posisi keuangan perusahaan, yang sangat berguna untuk mendukung pengambilan keputusan yang tepat. Data keuangan pada laporan keuangan bermanfaat untuk melihat kondisi kesehatan keuangan perusahaan[2].

Dalam penelitian ini akan dilakukan penentuan kelas perusahaan bank dengan metode K-nearest neighbor (KNN) untuk mengetahui status dari *financial distress* perusahaan bank tersebut.

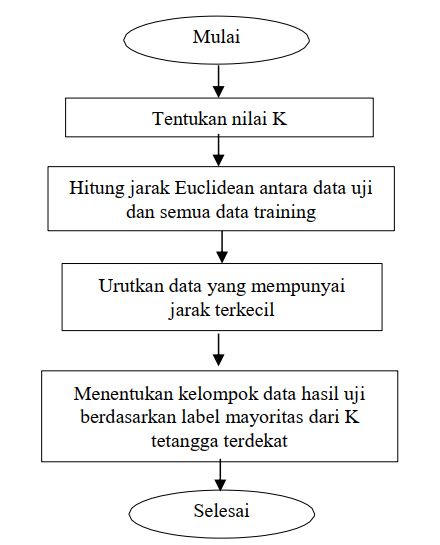
# Metode Penelitian

## K-Nearset Neighbor (KNN)

K-Nearest Neighbor (KNN) adalah metode melakukan klasifikasi terhadap objek berdasarkan data yang jaraknya paling dekat dengan objek tersebut. Metode ini bertujuan untuk mengklasifikasikan objek baru berdasarkan atribut dan training sample. Diberikan suatu titik query, selanjutnya akan ditemukan sejumlah K objek atau titik training yang paling dekat dengan titik query. Nilai prediksi dari query akan ditentukan berdasarkan klasifikasi tetanggaan[3].

Sebelum melakukan perhitungan dengan metode K-Nearest Neighbor, terlebih dahulu harus menentukan data latih dan data uji. Kemudian akan dilakukan proses perhitungan untuk mencari jarak menggunakan Euclidean. Setelah itu, akan dilakukan tahapan perhitungan dengan metode KNN.

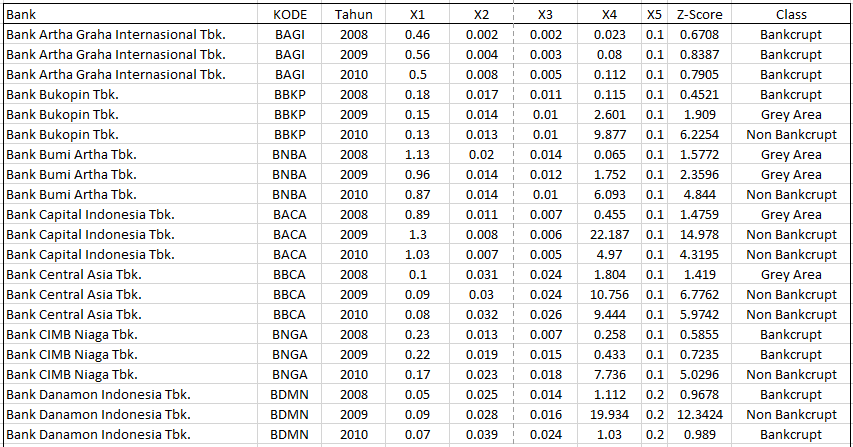
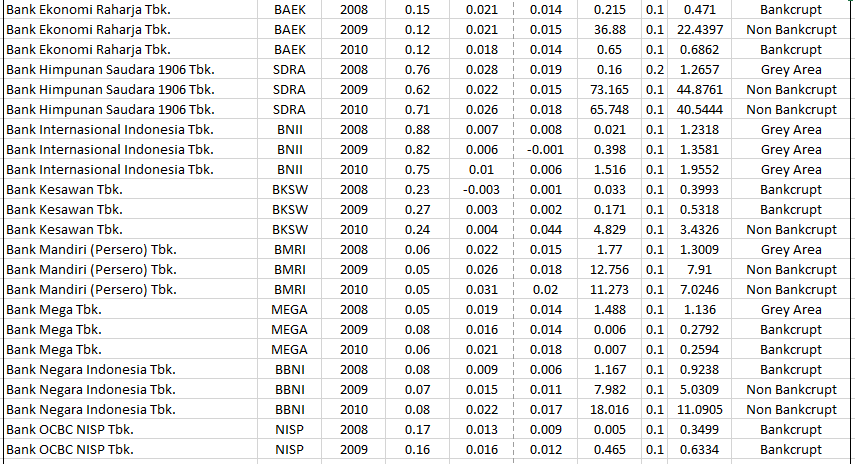
Gambar 1 Rumus Euclidean

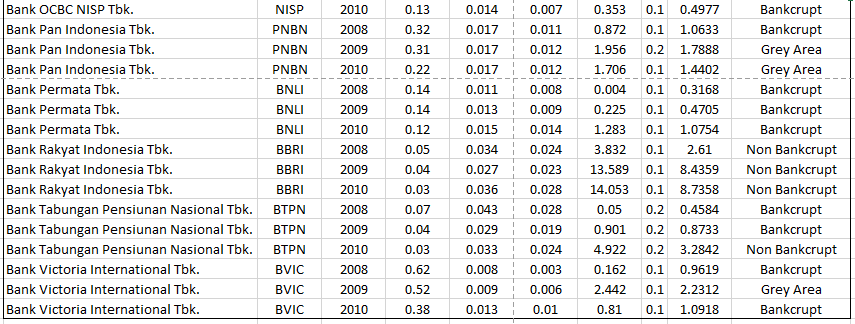
Algoritma K-Nearest Neighbor (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Teknik ini sangat sederhana dan mudah diimplementasikan. Mirip dengan teknik clustering, yaitu mengelompokkan suatu data baru berdasarkan jarak data baru itu ke beberapa data/tetangga terdekat. Pertama sebelum mencari jarak data ke tetangga adalah menentukan nilai K tetangga (neighbor). Lalu, untuk mendefinisikan jarak antara dua titik yaitu titik pada data training dan titik pada data testing, maka digunakan rumus Euclidean, sebagai berikut:

Gambar 2 Proses Metode K-Nearest Neighbor

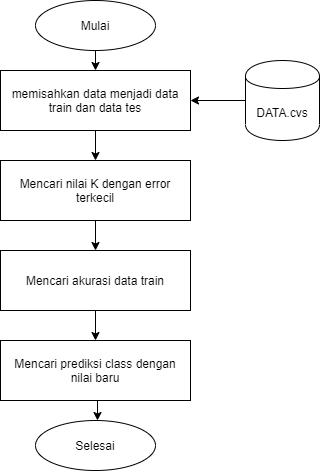
# Hasil dan Pembahasan

Dalam hasil dan pembahasan digunakan dataset bank konvensional dari tahun 2008 hingga 2010, berikut gambaran dataset:





Gambar 3 Data Bank

Program dibuat menggunakan *Environments Anaconda* dengan *python* versi 3.7.1, *library* yang digunakan diantaranya *Pandas, Numpy, Matplotlib* dan *Sklearn*. Dalam menjalankan program python, tentunya menggunakan fasilitas yang *Anaconda* berikan, salah satunya ***Jupyter Notebook*.** Berikut Skema dan *source code* program dengan metode k-nearest neighbors:

Gambar 4 Skema Program

Berikut *source code* program:

1. # Import kebutuhan Library
2. **import** pandas as pd
3. **import** numpy as np
4. **import** matplotlib.pyplot as plt
5. **from** sklearn.neighbors **import** KNeighborsClassifier
6. **from** sklearn.model\_selection **import** train\_test\_split
8. # Import data
9. data = pd.read\_csv('DATA.csv')
11. # pilih data dimana data\_x adalah nilai dari kolom X1 sampai X5, data\_y adalah data target class
12. data\_x = data.loc[:,'X1':'X5']
13. data\_y = data.Class
15. # pisahkan data menjadi 4 variabel dimana akan digunakan sebagai data train dan data test
16. x\_train, x\_test, y\_train, y\_test = train\_test\_split(data\_x, data\_y, test\_size=0.20)

19. # fungsi menampilkan data train dan data
20. **def** tampil\_data(x\_test, y\_test, x\_train, y\_train):
21. **print**("-------- Data Test --------")
22. display(pd.concat([x\_test, y\_test], axis=1))
23. **print**("-------- Data Train --------")
24. display(pd.concat([x\_train, y\_train], axis=1))

27. # fungsi mencari nilai K untuk KNN, dimana K akan dicari dengan mengetes angka 1 - 40.
28. **def** cari\_k():
29. error = []
30. **for** i **in** range(1, 40):
31. knn\_test\_error = KNeighborsClassifier(n\_neighbors=i)
32. knn\_test\_error.fit(x\_train, y\_train)
33. pred\_i = knn\_test\_error.predict(x\_test)
34. error.append(np.mean(pred\_i != y\_test))
36. # fungsi menampilkan 'hasil fungsi mencari nilai K'
37. **def** tampil\_k():
38. plt.figure(figsize=(12, 6))
39. plt.grid(True)
40. plt.plot(range(1, 40), error, color='red', linestyle='dashed', marker='o',
41. markerfacecolor='blue', markersize=5)
42. plt.title('Error Rate Nilai K')
43. plt.xlabel('Nilai K')
44. plt.ylabel('Error rata-rata')
46. # fungsi perhitungan KNN menggunakan data training dan data test, dimana nilai K diambil dari Nilai error terkecil
47. # maka K adalah 3
48. **def** cek\_k():
49. knn = KNeighborsClassifier(3).fit(x\_train,y\_train)
50. **for** y\_prediksi,y\_target **in** zip(knn.predict(x\_test),y\_test):
51. **print**('{:>5}'.format('Hasil x\_test prediksi adalah'),'{:15}'.format(y\_prediksi),'{:5}'.format('---'),
52. '{:>5}'.format('y\_test sebenarnya adalah'), '{:>5}'.format(y\_target))
54. hasil = knn.score(x\_test,y\_test)
55. **print**('Nilai Akurasi = '+ str(hasil\*100) +' %')
57. # fungsi dimana mencari class prediksi dengan nilai kustom
58. **def** prediksi\_nilai\_baru():
59. knn\_prediksi = KNeighborsClassifier(n\_neighbors=3)
60. knn\_prediksi.fit(x\_train,y\_train)
62. nilai = np.array([[0.9,0.021,0.01,0.555,0.2]])
63. hasil = knn\_prediksi.predict(nilai)
64. **print**('Prediksi KNN dengan Nilai baru')
65. **print**('X1 = '+str(nilai[0][0]))
66. **print**('X2 = '+str(nilai[0][1]))
67. **print**('X3 = '+str(nilai[0][2]))
68. **print**('X4 = '+str(nilai[0][3]))
69. **print**('X5 = '+str(nilai[0][4]))
70. **print**('')
72. **print**('Dari nilai diatas, diprediksi hasil class yaitu ' + hasil)

75. # panggil fungsi
76. tampil\_data(x\_test, y\_test, x\_train, y\_train)
78. cari\_k()
79. tampil\_k()
81. cek\_k()
83. prediksi\_nilai\_baru()

Berikut keterangan dari source code diatas:

1. # Import kebutuhan Library
2. **import** pandas as pd
3. **import** numpy as np
4. **import** matplotlib.pyplot as plt
5. **from** sklearn.neighbors **import** KNeighborsClassifier
6. **from** sklearn.model\_selection **import** train\_test\_split

Pada Line 1 – 6 dilakukan import beberapa library untuk mendukung jalanya program yang akan dibuat.

2. # Import data
3. data = pd.read\_csv('DATA.csv')
5. # pilih data dimana data\_x adalah nilai dari kolom X1 sampai X5, data\_y adalah data target class
6. data\_x = data.loc[:,'X1':'X5']
7. data\_y = data.Class
9. # pisahkan data menjadi 4 variabel dimana akan digunakan sebagai data train dan data test
10. x\_train, x\_test, y\_train, y\_test = train\_test\_split(data\_x, data\_y, test\_size=0.20)

Pada Line 9, dilakukan import dataset. Line 12 – 13 dimana melakukan pemilihan data X1 sampai X5 sebagai variabel data\_x, dan data target yaitu class bank sebagai variabel data\_y. Pada Line 16 dilakukan pemisahan DATA.csv menjadi data train dan data tes, dimana data diambil menjadi data tes sebesar 20%.

1. # fungsi menampilkan data train dan data
2. **def** tampil\_data(x\_test, y\_test, x\_train, y\_train):
3. **print**("-------- Data Test --------")
4. display(pd.concat([x\_test, y\_test], axis=1))
5. **print**("-------- Data Train --------")
6. display(pd.concat([x\_train, y\_train], axis=1))

Line 20-24 merupakan fungsi untuk menampilkan data train dan data tes.

2. # fungsi mencari nilai K untuk KNN, dimana K akan dicari dengan mengetes angka 1 - 40.
3. **def** cari\_k():
4. error = []
5. **for** i **in** range(1, 40):
6. knn\_test\_error = KNeighborsClassifier(n\_neighbors=i)
7. knn\_test\_error.fit(x\_train, y\_train)
8. pred\_i = knn\_test\_error.predict(x\_test)
9. error.append(np.mean(pred\_i != y\_test))

Line 28-34 merupakan fungsi untuk mencari nilai K yang tepat dengan mencari nilai error K, dimana K dilakukan tes dari 1 hingga 40.

2. # fungsi menampilkan 'hasil fungsi mencari nilai K'
3. **def** tampil\_k():
4. plt.figure(figsize=(12, 6))
5. plt.grid(True)
6. plt.plot(range(1, 40), error, color='red', linestyle='dashed', marker='o',
7. markerfacecolor='blue', markersize=5)
8. plt.title('Error Rate Nilai K')
9. plt.xlabel('Nilai K')
10. plt.ylabel('Error rata-rata')

Line 37-44 merupakan fungsi untuk menampilkan hasil fungsi sebelumnya yaitu fungsi mencari nilai K.

1. # fungsi perhitungan KNN menggunakan data training dan data test, dimana nilai K diambil dari Nilai error terkecil
2. # maka K adalah 3
3. **def** cek\_k():
4. knn = KNeighborsClassifier(3).fit(x\_train,y\_train)
5. **for** y\_prediksi,y\_target **in** zip(knn.predict(x\_test),y\_test):
6. **print**('{:>5}'.format('Hasil x\_test prediksi adalah'),'{:15}'.format(y\_prediksi),'{:5}'.format('---'),
7. '{:>5}'.format('y\_test sebenarnya adalah'), '{:>5}'.format(y\_target))
9. hasil = knn.score(x\_test,y\_test)
10. **print**('Nilai Akurasi = '+ str(hasil\*100) +' %')

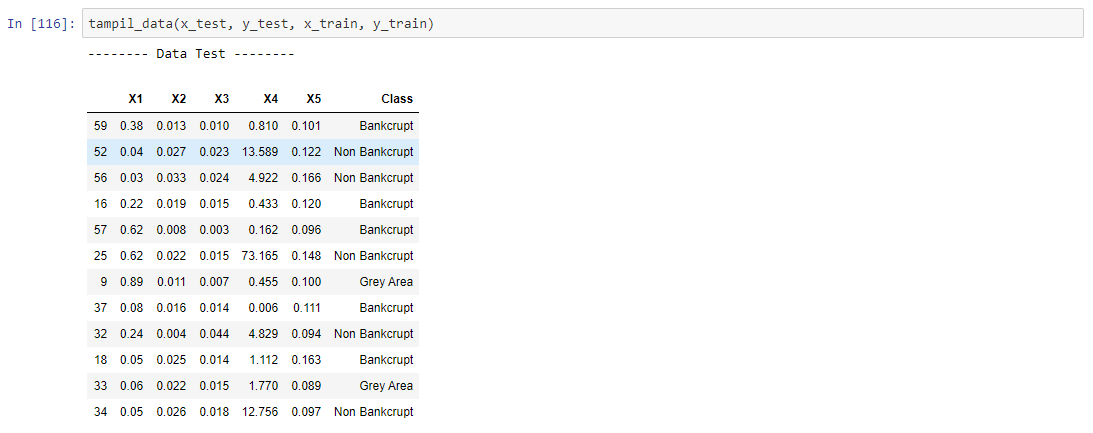
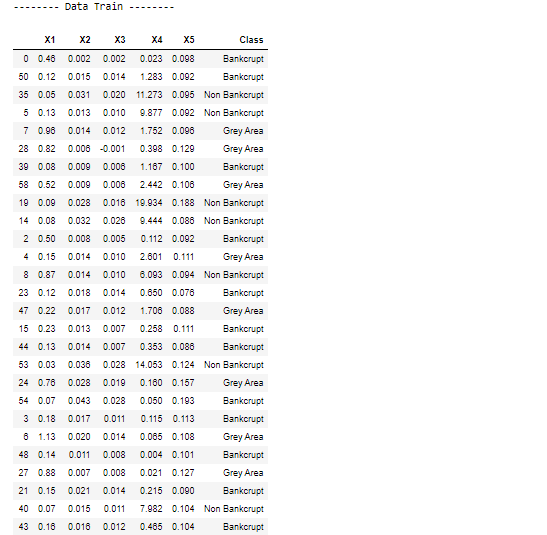
Pada Line 48-55 dilakukan pengujian data train dan data set menggunakan K = 3, dimana nilai tersebut didapatkan dari hasil mencari nilai K dengan error paling kecil.

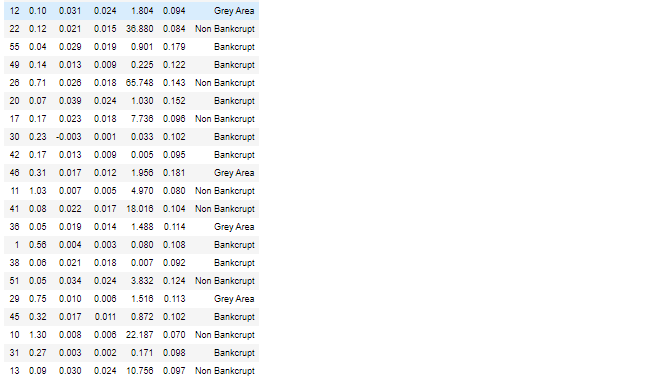
1. # fungsi dimana mencari class prediksi dengan nilai kustom
2. **def** prediksi\_nilai\_baru():
3. knn\_prediksi = KNeighborsClassifier(n\_neighbors=3)
4. knn\_prediksi.fit(x\_train,y\_train)
6. nilai = np.array([[0.9,0.021,0.01,0.555,0.2]])
7. hasil = knn\_prediksi.predict(nilai)
8. **print**('Prediksi KNN dengan Nilai baru')
9. **print**('X1 = '+str(nilai[0][0]))
10. **print**('X2 = '+str(nilai[0][1]))
11. **print**('X3 = '+str(nilai[0][2]))
12. **print**('X4 = '+str(nilai[0][3]))
13. **print**('X5 = '+str(nilai[0][4]))
14. **print**('')
16. **print**('Dari nilai diatas, diprediksi hasil class yaitu ' + hasil)

Pada Line 57-71 dilakukan prediksi dengan nilai tes baru, dimana X1=0.9; X2=0.021; X3=0.01; X4=0.555; X5=0.2.

3. # panggil fungsi
4. tampil\_data(x\_test, y\_test, x\_train, y\_train)
6. cari\_k()
7. tampil\_k()
9. cek\_k()
11. prediksi\_nilai\_baru()

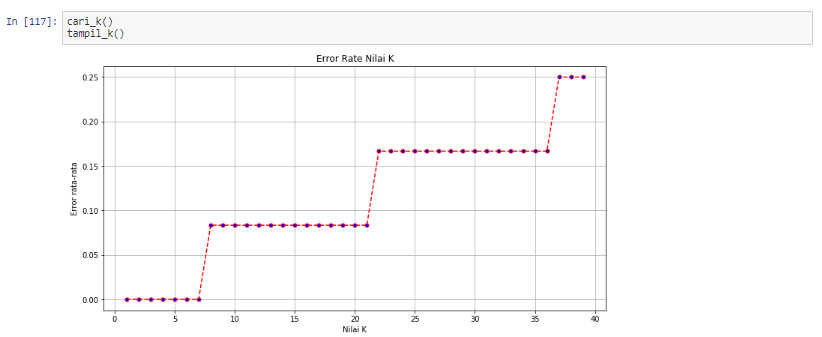
Pada Line 75-82 dilakukan pemanggilan fungsi *def* yang telah dibuat untuk dijalankan.

Berikut Hasil dari program metode k-nearest neighbors:



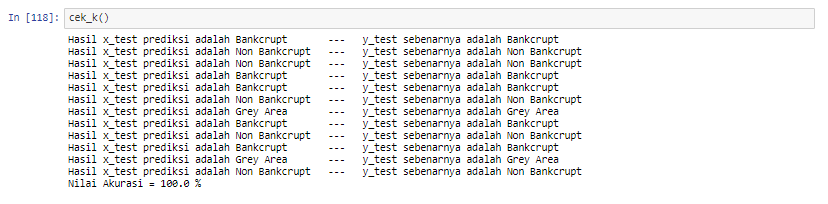
Gambar 5 Hasil tampil data

Hasil diatas merupakan hasil dari fungsi tampil\_data(), terlihat data train dan data tes.



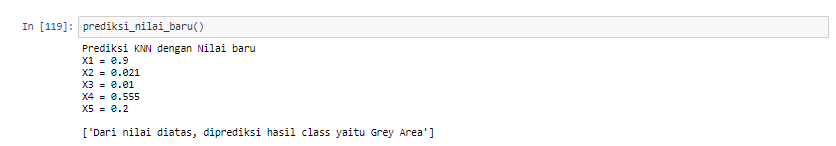
Gambar 6 Hasil mencari nilai K

Hasil diatas merupakan hasil fungsi cari\_k() dan tampil\_k(). Terlihat pada gambar bahwa K dengan nilai 1 – 7 menghasilkan error yang lebih kecil dari pada K lebih dari 7, sehingga K yang digunakan dalam penelitian ini adalah 3.



Gambar 7 hasil train

Hasil diatas merupakan hasil fungsi cek\_k(). Fungsi ini melakukan train dengan metode k-nearest neighbors menggunakan DATA.csv dengan membagi data menjadi data train dan data tes, Nilai akurasi dari proses ini sebesar 100%.



Gambar 8 hasil prediksi

Hasil diatas merupakan hasil dari fungsi prediksi\_nilai\_baru() dimana fungsi ini menghasilkan prediksi klasifikasi class dengan nilai sesuai dengan keinginan. Nilai-nilai pada gambar diatas menghasilkan prediksi klasifikasi class ‘Grey Area’.

# Kesimpulan

Dari hasil yang dipaparkan diatas, proses train menggunakan metode k-nearest neighbors menghasilkan nilai akurasi sebesar 100% dan dapat menghasilkan prediksi untuk nilai yang diberikan sesuai dengan keinginan. Untuk mendapatkan hasil akurasi yang optimal tentunya harus menggunakan nilai K yang memiliki error paling kecil.

# Daftar Isi

[1] Muslich, Mohamad. 2008. Manajemen Keuangan Modern:Analisis, Perencanaan dan Kebijaksanaan. Jakarta: Bumi Aksara.

[2] Altman, Edward I. 1968. Financial Ratios, Discriminant Analysis and The Prediction of Corporate Bankcrupty. Journal Of Financial, 23 (4) : 189-209. 2002. Revisiting Credit Scoring Models in a Basel II Environment. Prepared for “Credit Rating:Methodologies, Rationale, and Default Risk”, London Risk Books 2002.

[3] Kusrini & A. T. Luthfi, Algoritma Data Mining. Yogyakarta: C.V

ANDI OFFSET (Penerbit Andi), 2009.