Analisis Kurva ROC Sebagai Penentu Jumlah Komponen PCA Terbaik Pada Klasifikasi Cut Berlian Menggunakan Algoritma Random Forest

Zulfikar Akbar[1], Atikah Aulia Putri[2], Agus Setiawan[3], Muhammad Insan Khamil[4], Muhammad Eka Suryana, M. Kom.[5]

[1]Email: zulfikar.78.akbar@gmail.com
[2]Email: atikahap.yes@gmail.com
[3]Email: agus2429e@gmail.com
[4]Email: Khamil_insan0@yahoo.com
[5]Email: eka-suryana@unj.ac.id

Program Studi Ilmu Komputer, FMIPA, Universitas Negeri Jakarta

***Abstrak*— Algoritma PCA merupakan salah satu algoritma untuk melakukan reduksi dimensi suatu data yang besar. Data awal berisi atribut-atribut yang saling berkorelasi. Namun setelah direduksi data tersebut menjadi tidak berkorelasi satu sama lain. Untuk mengetahui jumlah hasil reduksi yang akan menjadi n-komponen PCA bisa dilihat melalui varians per n-komponen PCA. Selain itu bisa juga menggunakan analisis kurva ROC. Data yang akan direduksi merupakan data berlian yang diunduh dari laman Kaggle. Data akan diklasifikasi juga menggunakan algoritma Random Forest. Alat bantu kami yaitu paket *machine learning* pada bahasa pemrograman Python. Hasil dari penelitian ini yaitu didapat 5-komponen PCA sebagai jumlah komponen PCA terbaik.**

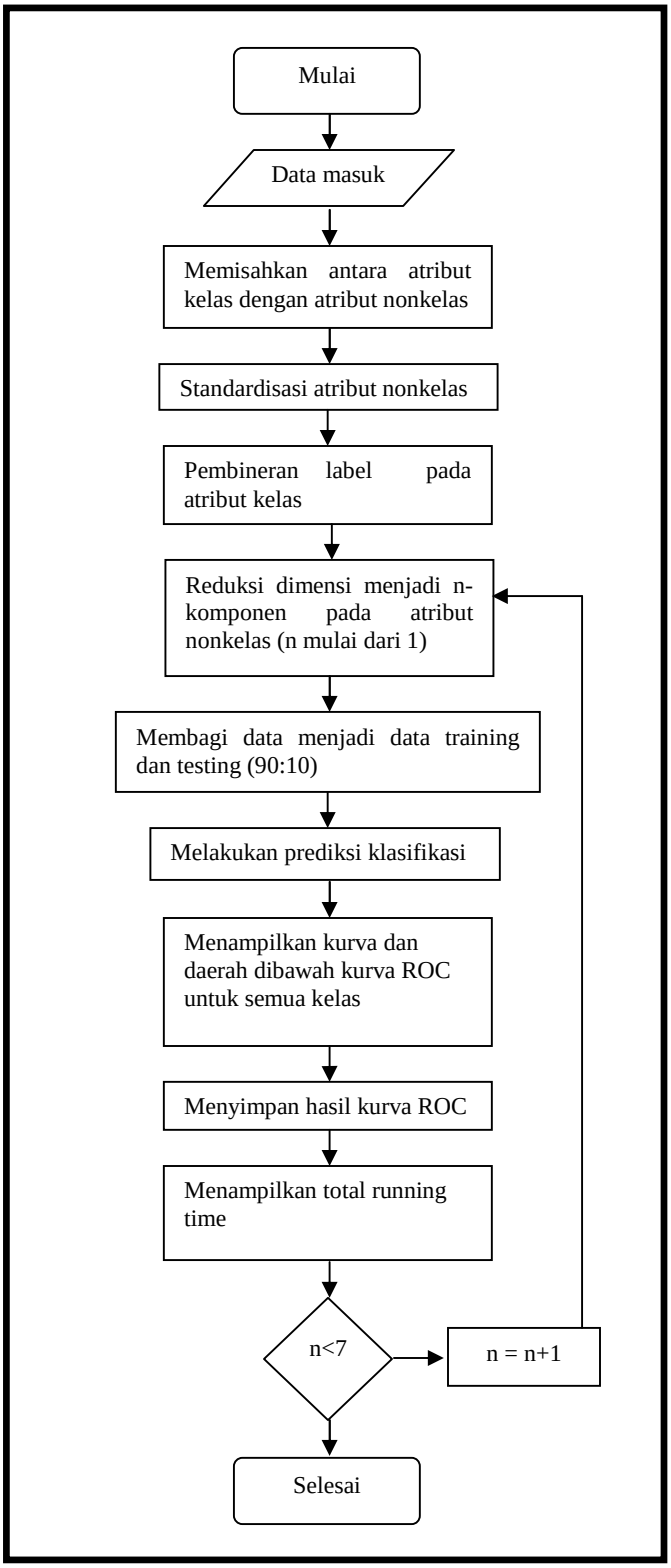
***Kata kunci*— Algoritma, Analisis, Berlian, Data, Dimensi, Kaggle, Kurva, N-komponen, PCA, Random Forest, Reduksi, ROC, Terbaik, Varians**

I. PENDAHULUAN

Berlian merupakan jenis permata. Di dunia ini banyak sekali jenis berlian mulai dari yang terbaik sampai yang terburuk. Aga terhindar dari kecurangan oknum penjual, maka sudah menjadi keharusan untuk mengetahui kriteria penentu kualitas berlian. Kualitas berlian biasanya diukur berdasarkan 4'Cs (*Cut, Clarity, Color, Carat*). Untuk melakukan pemeriksaan resmi yang diakui, maka lembaga seperti GIA (*Gemmology Institute America*) dan AGS (*America Gem Society*) akan melakukan pemeriksaan secara lengkap dan terpercaya karena terdapat sertifikat berisi segala informasi berlian yang diperiksa. Agar nantiya berlian tersebut berharga tinggi saat dijual lagi. Ada pun penelitian terkait dengan reduksi dimensi data terdapat dalam paper [13] yang mereduksi data jantung koroner dari 13 variabel menjadi 4 variabel. Pada penelitian terdahulu tersebut tidak dilakukan dilakukan analisis lanjut untuk mendapat alasan penggunaan n-komponen PCA. Pada penelitian ini kami akan melakukan tersebut yaitu malakukan analisis kurva ROC untuk mendapat reduksi data menjadi n-komponen PCA terbaik.

II. METODE PENELITIAN

Pada penelitian ini kami sudah membuat diagram alir kegiatan yang bisa dilihat di gambar 2. 1



Gambar 2.1 Diagram alir penelitian

A. Data

Data yang dipakai merupakan data berlian yang dapat diunduh pada laman Kaggle[1]. Informasi mengenai data dapat dilihat pada tabel 2. 1

Tabel 2. 1 Informasi data

no	atribut	informasi
1	banyak *instance*	53940
2	banyak atribut	10
3	atribut 1 sampai 9	*instance*
4	kelas	*1. fair*
		2. good
		3. very good
		4. premium
		5. ideal
5	distribusi kelas	*1. fair: 1610*
		2. good: 4906
		3. very good: 12082
		4. premium: 13791
		5. ideal: 21551

Tabel 2. 2 Informasi atribut

no	atribut	rentang	informasi
1	*carat*	0.2 - 5.01	berat berlian
2	*color*	0.0: D (terbaik)	warna berlian
		0.5: E	rentang 0.0 sampai 3.0 bedasarkan sistem AGS
		1.0: F	rentang D sampai J berdasarkan sistem GIA
		1.5: G	
		2.0: H	
		2.5: I	
		3.0: J	
3	*clarity*	0: IF (terbaik)	kejernihan berlian
		1: VVS1	rentang 0 sampai 7 bedasarkan sistem AGS
		2: VVS2	rentang IF sampai I1 berdasarkan sistem GIA
		3: VS1	
		4: VS2	
		5: SI1	
		6: SI2	
		7: I1	
4	*price*	326 - 18823	harga berlian dalam US$ (tidak dipakai)
5	*x*	0 - 10.74	dimensi panjang berlian dalam milimeter
6	*y*	0 - 58.9	dimensi lebar berlian dalam milimeter
7	*z*	0 - 31.8	dimensi tinggi berlian dalam milimeter
8	*depth*	43 - 79	kedalaman berlian
9	*table*	43 - 95	sisi mendatar dan paling luas di atas berlian

10	*cut (*kelas*)*	*1. fair* *2. good* *3. very good* *4. premium* *5. ideal* (terbaik)	kelas berdasarkan potongan (*cut*) berlian

B. Standardisasi data

Dalam *machine learning*, banyak sekali data yang bisa diolah, dan data ini memiliki beberapa dimensi. Standardisasi fitur membuat nilai setiap fitur dalam data memiliki mean nol (bila mengurangi mean pada pembilang) dan varians unit. Metode ini banyak digunakan untuk normalisasi di banyak algoritma *machine learning* (misalnya *support vector machines*, *logistic regression*, and *neural networks*) [2]. Metode perhitungan umum adalah menentukan mean distribusi dan standar deviasi untuk masing-masing fitur. Selanjutnya kita kurangi mean dari masing-masing fitur. Kemudian kita membagi nilai (mean sudah dikurangkan) dari setiap fitur dengan standar deviasi.

Gambar 2. 2 Perhitungan standardisasi

$$x' = \frac{x - \bar{x}}{\sigma}$$

Yang mana x adalah vektor fitur asli, $\bar{x}$ adalah mean dari vektor fitur itu, dan σ adalah standar deviasinya.

C. Random Forest

Random Forest adalah algoritma *bagging* yang berhasil mengarah pada titik regularisasi dimana kualitas model setinggi mungkin dan varians dan masalah bias dikompromikan [3]. Untuk mengatasi masalah *overfitting* yang dihadapi *decision tree*, *Random Forest* membangun ratusan atau ribuan diantaranya. Untuk membuat pohon berbeda satu sama lain, *Random Forest* menggunakan sampel acak dengan penggantian [4]. Rata-rata, 37% dari baris akan ditinggalkan dari setiap sampel [5]. Setiap pohon mengklasifikasikan pengamatannya, dan pada akhirnya memilih mayoritas [6], keputusan dipilih. *Random Forest* juga dapat digunakan dalam *unspervised mode* untuk menilai proksimiti di antara titik data [7].

D. PCA (*Principal Component Analysis*)

Principal component analysis (analisa komponen utama) adalah salah satu fitur ekstraksi (reduksi) variabel yang banyak digunakan. Bisa dikatakan *principal component analysis* merupakan analisa tertua dan paling terkenal dari teknik statistika *multivariate* [8]. PCA pertama kali perkenalkan oleh Karl Pearson pada tahun 1901. Harold Hotelling melakukan analisa untuk variabel stokastik. Hotelling menggunakan pendekatan PCA yang sebelumnya telah dikemukan oleh Pearson dan memperkenalkan istilah "*component*" sebagai variabel yang dihasilkan dengan menggunakan metodologi PCA. Perkembangan selanjutnya dikenal dengan istilah "*principal component*" yang menjelaskan komponen utama atau variabel baru yang dihasilkan atau direduksi. Inilah cikal bakal dari analisa PCA. Analisa PCA dikenal juga dengan dengan Transformasi Karhunen-Loeve dan Transformasi Hotelling.

Metode PCA sangat berguna digunakan jika data yang ada memiliki jumlah variabel yang besar dan memiliki korelasi antar variabelnya. Perhitungan dari *principal component analysis* didasarkan pada perhitungan nilai eigen dan vektor eigen yang menyatakan penyebaran data dari suatu dataset. Tujuan dari analisa PCA adalah untuk mereduksi variabel yang ada menjadi lebih sedikit tanpa harus kehilangan informasi yang termuat dalam data awal. Dengan menggunakan PCA, variabel sebanyak n variabel akan direduksi menjadi k variabel baru (*principal component*) dengan jumlah k lebih sedikit dari n dan dengan hanya menggunakan k *principal component* akan menghasilkan nilai yang sama dengan menggunakan n variabel [9]. Variabel hasil dari reduksi tersebut dinamakan *principal component* (komponen utama) atau bisa juga disebut faktor. Sifat dari variabel baru yang terbentuk dengan analisa PCA nantinya selain memiliki jumlah variabel yang berjumlah lebih sedikit tetapi juga menghilangkan korelasi antar variabel yang terbentuk.

E. Kurva dan daerah di bawah kurva ROC (*Receive Operation Characteristic*)

Pada umumnya kurva dan daerah di bawah kurva ROC berguna untuk menilai suatu algoritma klasifikasi pada klasifikasi biner. Namun juga bisa digunakan pada klasifikasi multi-kelas. Kurva ROC dbentuk dari *confusion matrix. Confusion matrix* merupakan tabel berisi empat kemungkinan kemunculan pada suatu klasifikasi seperti pada tabel 2. 3

Tabel 2. 3 *Confusion matrix* (*contingency table*)

KEADAAN SEBENARNYA	PREDIKSI	
	True	*False*
True	**TP**	**FN**
False	**FP**	**TN**

Pada pengukuran kinerja menggunakan *confusion matrix*, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Nilai *True Negative* (TN) merupakan jumlah data negatif yang terdeteksi dengan benar, sedangkan *False Positive* (FP) merupakan data negatif namun terdeteksi sebagai data positif. Sementara itu, *True Positive* (TP) merupakan data positif yang terdeteksi benar. *False Negative* (FN) merupakan kebalikan dari True Positive, sehingga data positif, namun terdeteksi sebagai data negatif.

Berdasarkan nilai *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), dan *True Positive* (TP) dapat diperoleh nilai akurasi, presisi (*Positif Predictive Value*), *recall* (*True Positive rate* atau *Sensitivity*), *False Positive rate*, *True Negative rate* (Specificity), dan *False Negative rate*. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Nilai akurasi dapat diperoleh dengan persamaan (1). Nilai presisi menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif. Presisi dapat diperoleh dengan persamaan (2). Nilai *recall* menunjukkan berapa persen data kategori positif yang terklasifikasikan dengan benar oleh sistem. Nilai *recall* diperoleh dengan persamaan (3). Nilai *False Positive rate* menunjukkan banyak data dalam kategori negatif yang salah diklasifikasikan menjadi positif, bisa dilihat pada persamaan (4). Nilai *True Negative rate* menunjukkan banyaknya data yang dikategorikan negatif dan diklasifikasi dengan benar, dapat dilihat pada persamaan (5). *Nilai False Negative rate* menunjukkan banyak data dalam kategori positif yang salah diklasifikasikan menjadi negatif, ditunjukkan oleh persamaan (6).

$$\text{Akurasi} = (TP+TN)/(TP+TN+FP+FN)\ 100\% \quad \text{.....(1)}$$

$$\text{Presisi} = TP/(TP+FP)\ 100\% \quad \text{.....(2)}$$

$$\text{TP rate} = TP/(TP+FN)\ 100\% \quad \text{.....(3)}$$

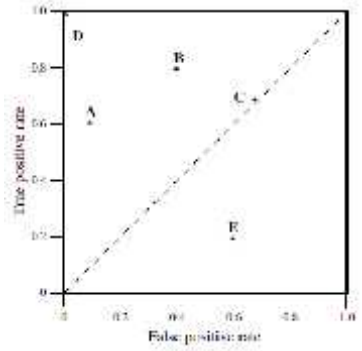
$$\text{FP rate} = FP/(FP+TN)\ 100\% \quad \text{.....(4)}$$

$$\text{TN rate} = TN/(FP+TN)\ 100\% \quad \text{.....(5)}$$

$$\text{FN rate} = FN/(FN+TP)\ 100\% \quad \text{.....(6)}$$

Grafik ROC merupakan grafik 2-dimensi, yang mana sumbu-x menunjukkan FP rate dan sumbu-y menunjukkan TP rate. Grafik ROC menggambarkan *tradeoffs* relatif antara *false positive* (FP) dan *true positive* (TP). Titik-titik pada ROC sangat perlu untuk diperhatikan [12].

Gambar 2. 3 Grafik ROC

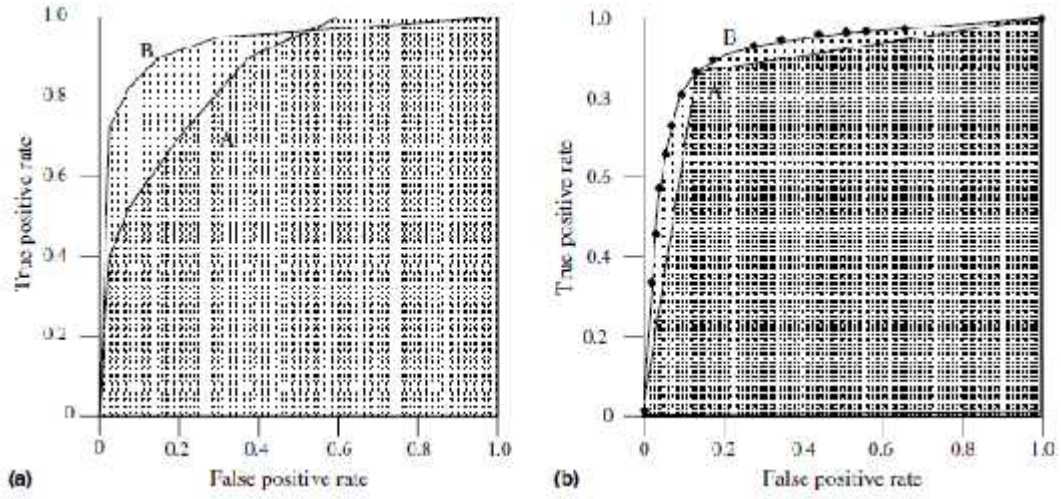


Titik terbawah di sebelah kiri (0,0) merupakan strategi untuk tidak pernah mengeluarkan klasifikasi positif; seperti *classifier* tidak mengeluarkan kesalahan *false positive* tapi juga tidak mendapatkan *true positive*. Strategi yang berlawanan, tanpa syarat mengeluarkan klasifikasi positif, diwakili oleh titik kanan atas (1, 1). Intinya (0, 1) mewakili klasifikasi sempurna. Kinerja D sempurna seperti yang ditunjukkan. Secara informal satu poin di ruang ROC lebih baik dari yang lain jika mengarah ke barat laut (tingkat TP lebih tinggi, FP rate lebih rendah, atau keduanya) yang pertama [12].

Klasifikasi muncul di sisi kiri grafik ROC, di dekat sumbu X, mungkin saja dianggap " konservatif ": mereka membuat klasifikasi positif hanya dengan bukti kuat sehingga mereka membuat sedikit kesalahan *false positive*, tapi mereka sering memiliki tingkat *true positive* yang rendah. Klasifikasi di sisi kanan atas ROC Grafik dapat dianggap sebagai " liberal ": mereka membuat klasifikasi positif dengan bukti lemah sehingga mereka mengklasifikasikan hampir semua positif dengan benar, tapi mereka sering memiliki *false positive rates* yang tinggi. Pada Gambar 2. 3 lebih konservatif daripada B. Banyak Domain dunia nyata didominasi oleh sejumlah besar contoh negatif, jadi performa di sisi paling kiri dari grafik ROC menjadi lebih menarik [12].

Untuk membandingkan *classifier* yang mungkin kita inginkan mengurangi kinerja ROC menjadi nilai skalar tunggal yang mewakili kinerja yang diharapkan, metode yang umum adalah menghitung daerah di bawah kurva ROC, disingkat AUC [10] [11] AUC adalah sebagian dari luas unit persegi, nilainya akan selalu antara 0 dan 1. Namun, karena acak menebak menghasilkan garis diagonal antara (0, 0) dan (1, 1), yang memiliki luas 0,5, tidak ada klasifikasi yang realistis harus memiliki AUC kurang dari 0,5. AUC memiliki properti statistik penting: AUC dari *classifier* setara dengan probabilitas itu *classifier* akan menentukan contoh positif yang dipilih secara acak lebih tinggi dari contoh negatif yang dipilih secara acak [12].

Gambar 2. 4 Dua Grafik ROC

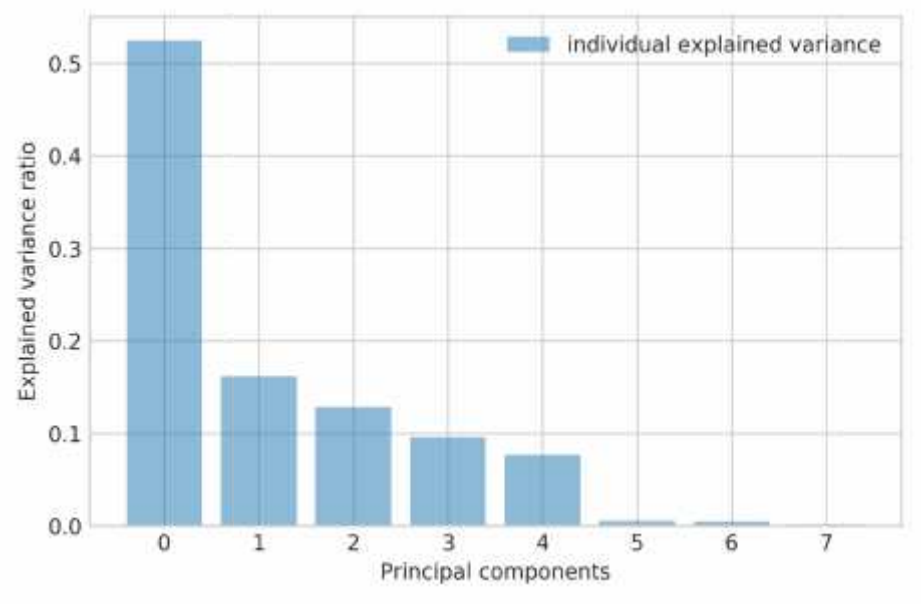


Pada gambar 2. 4a *classifier* B memliki daerah yang lebih besar dan karena itu memiliki rata-rata performansi yang lebih baik. Pada gambar 2. 4b, A merupakan AUC untuk *binary classifier* dan B merupakan AUC untuk *scoring classifier*. *Classifier* A mewakili kinerja B ketika B digunakan dengan ambang batas tunggal tetap. Padahal penampilan keduanya sama dengan titik tetap (ambang batas A), kinerja A menjadi lebih rendah dari B lebih jauh dari titik ini [12].

III. HASIL PENELITIAN

Setelah dilakukan serangkaian alur kerja seperti pada gambar 2. 1 maka didapatkan gambar variansi per n-komponen PCA dan kurva ROC untuk setiap komponen PCA, yang mana n mulai dari 1 sampai dengan 7. Adapun hasilnya berupa analisis untuk mengambil keputusan mengenai nilai n komponen PCA (hasil reduksi data atribut nonkelas awal) yang terbaik.

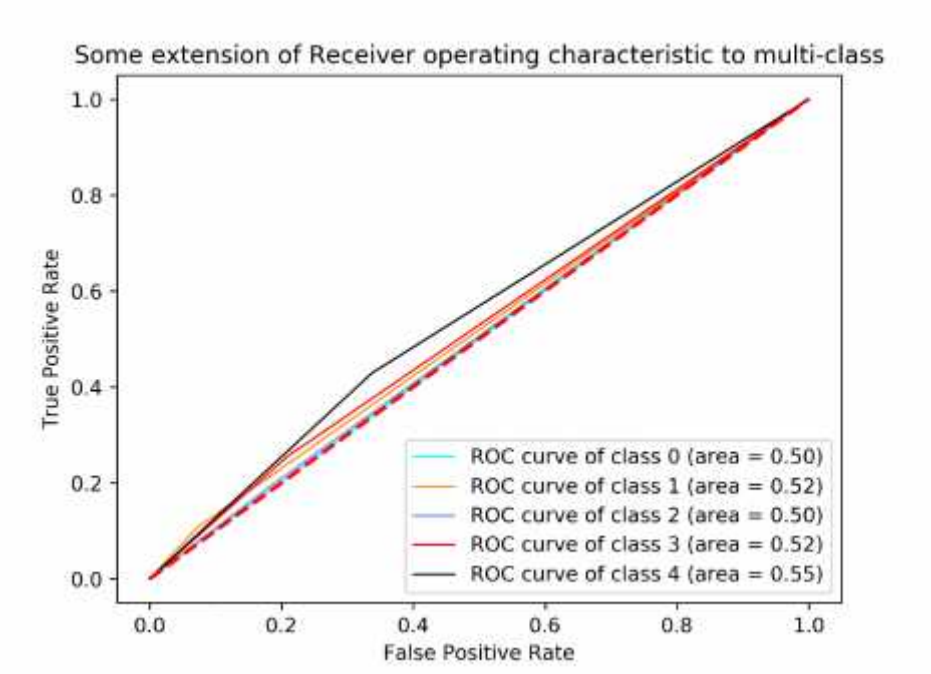
Gambar 3. 1 Varians per n-komponen PCA



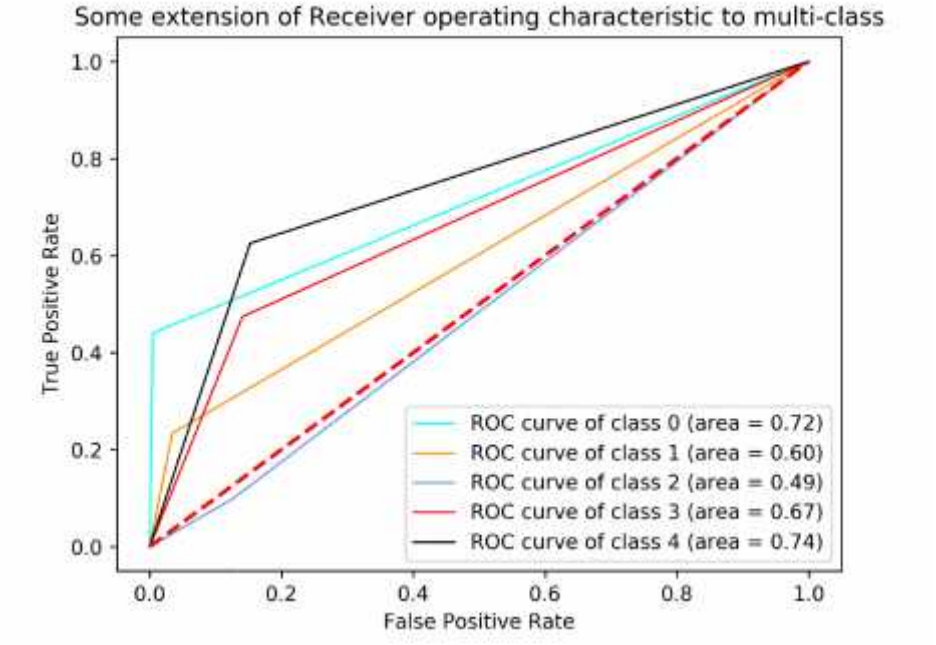
Tabel 3. 1 Tabel varians per n-komponen PCA

n-komponen PCA	rasio varians per komponen	rasio kumulatif varians (%)
1	0.52433069	52.433069
2	0.16188763	68.621832
3	0.12840308	81.462140
4	0.09611053	91.073193
5	0.07680177	98.753370
6	0.00586057	99.339427
7	0.00497189	99.836616
8	0.00163383	99.838250

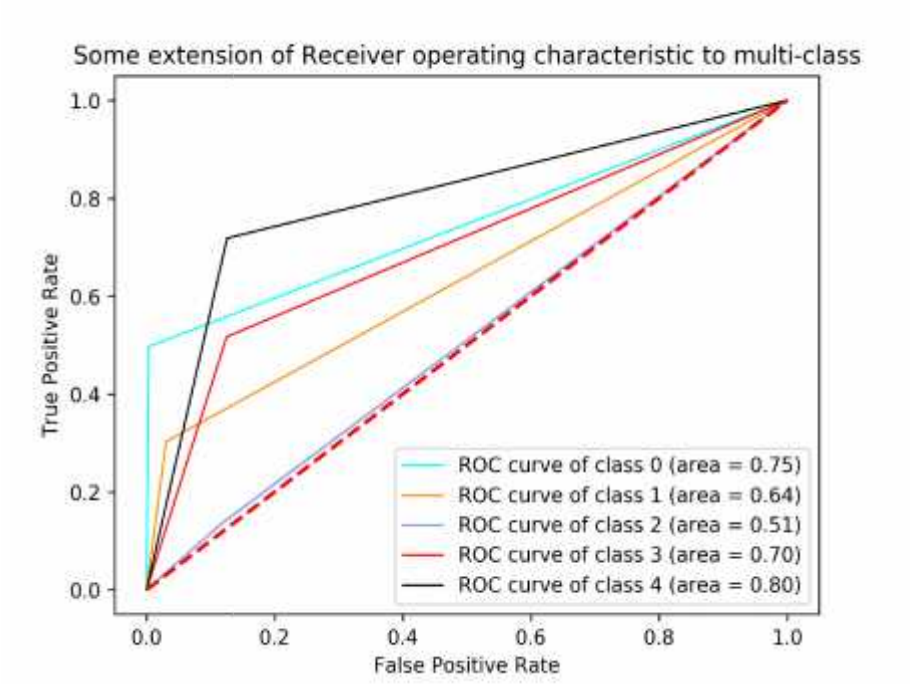
Gambar 3. 2 1-komponen PCA



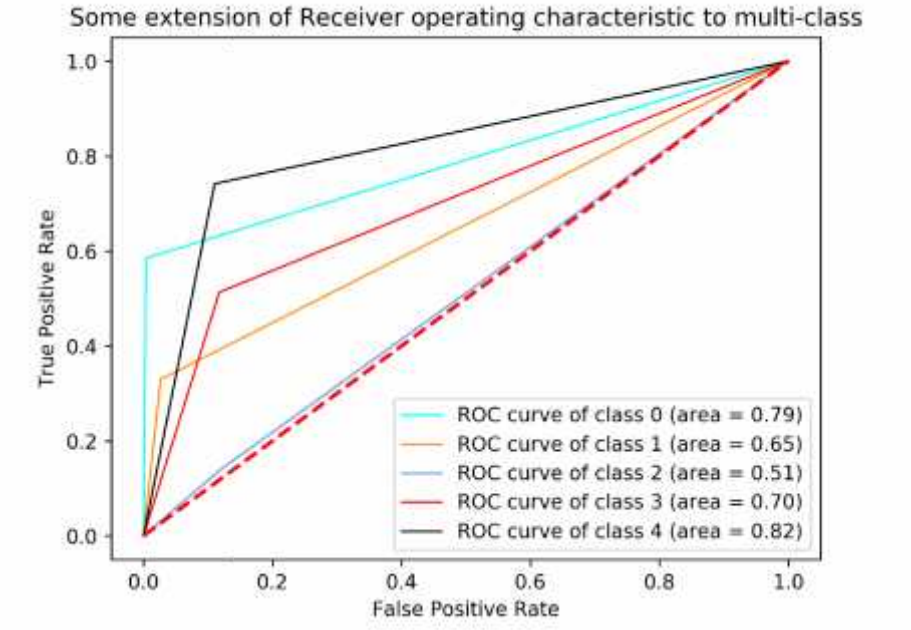
Gambar 3. 3 2-komponen PCA



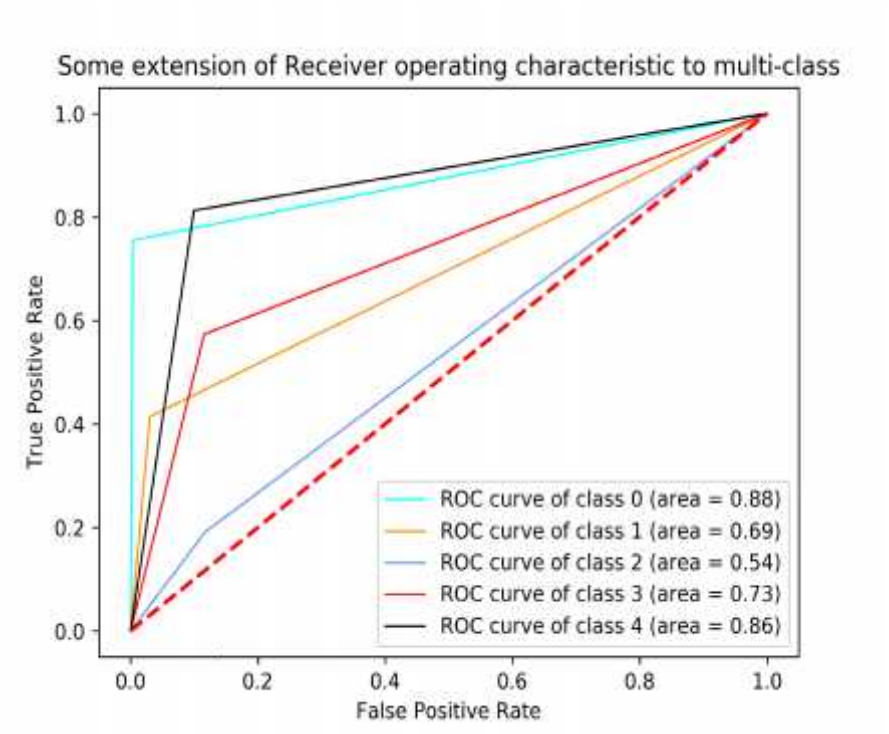
Gambar 3. 4 3-komponen PCA



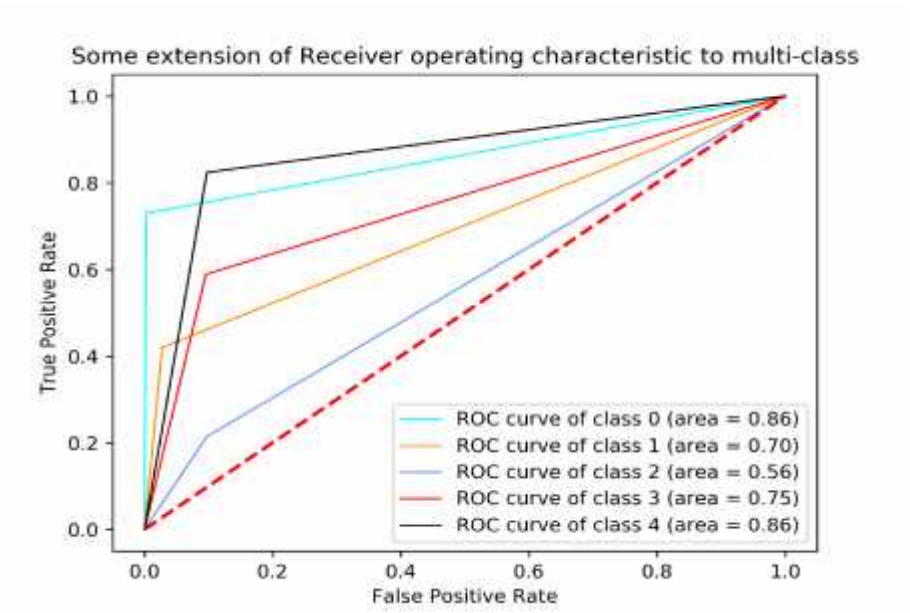
Gambar 3. 5 4-komponen PCA



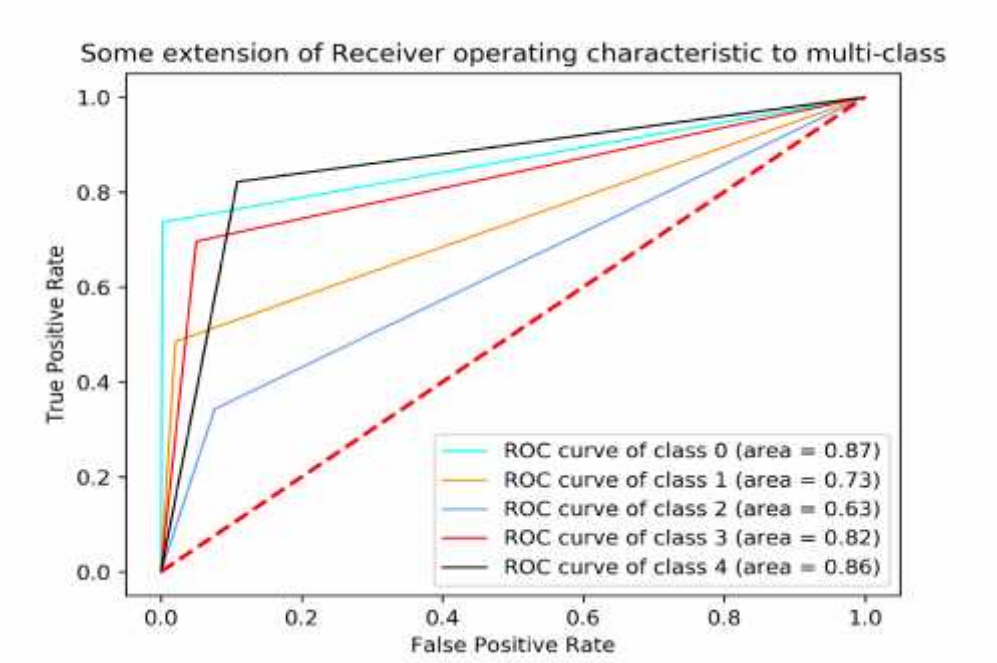
Gambar 3. 6 5-komponen PCA



Gambar 3. 7 6-komponen PCA



Gambar 3. 8 7-komponen PCA



Tabel 3. 2 kumpulan data dari gambar kurva ROC

n-komponen PCA	kurva ROC untuk kelas ke-n					*Running time* *wall-clock second ellapsed*
	0	1	2	3	4	
1	0.500830765349	0.517515051998	0.504938046593	0.522938875535	0.545740602378	5.70426415550
2	0.717738011569	0.599719485495	0.486236032129	0.666670265467	0.736488716437	6.57501214453
3	0.746803986232	0.636152162014	0.509091485406	0.696036842444	0.796382618969	8.43265316362
4	0.790638121497	0.651512041598	0.510029428983	0.697898764160	0.815698981597	9.15232205698
5	0.875639292858	0.691810344828	0.536949172353	0.728843994457	0.856585927088	10.77495330340
6	0.863633742409	0.696120689655	0.558474328636	0.746506357799	0.863232703941	12.09861130430
7	0.867064929448	0.732177066229	0.633293381500	0.823015328694	0.856514162181	13.78338605470

*keterangan:
Kelas ke – 0: fair
1: good
2: very good
3: premium
4: ideal

Berdasarkan penjelasan sebelumnya yaitu secara informal satu poin di ruang ROC lebih baik dari yang lain jika mengarah ke barat laut (tingkat TP lebih tinggi, FP rate lebih rendah, atau keduanya) yang pertama [12]. Maka dapat dilihat bahwa untuk setiap gambar kurva ROC kelas ke-0 yang paling mengarah ke arah barat laut ujungnya. Masih di kelas ke-0, dari n-komponen PCA pertama sampai kelima mengalami kenaikan (0.500830765349 sampai dengan 0.875639292858) , namun turun di n-

komponen PCA keenam dan ketujuh (0.863633742409 dan 0.867064929448). Hal ini menunjukkan bahwa jumlah n-komponen PCA terbaik yaitu 5-komponen PCA.

IV. KESIMPULAN

1. Jumlah n-komponen PCA terbaik yaitu 5-komponen PCA dari 8 variabel awal
2. Kelima variabel baru tersebut mampu menjelaskan kumulatif varians sebesar 98.753370. Lihat tabel 3. 1

REFERENSI

[1] Website Kaggle. https://www.kaggle.com/shivam2503/diamonds
[2] Grus, Joel (2015). Data Science from Scratch. Sebastopol, CA: O'Reilly. pp. 99, 100. ISBN 978-1-491-90142-7.
[3] Ziegel, E. R. (2012). The Elements of Statistical Learning. Technometrics.
[4] Kotsiantis, S. B. (2013). Decision trees: a recent overview. Artificial Intelligence Review, 39(4), 261-283.
[5] Montano-Gutierrez, L. F., Ohta, S., Kustatscher, G., Earnshaw, W. C., & Rappsilber, J. (2016). Nano Random Forests to mine protein complexes and their relationships in quantitative proteomics data, 050302.
[6] Pudlo, P., Marin, J. M., Estoup, A., Cornuet, J. M., Gautier, M., & Robert, C. P. (2016). Reliable ABC model choice via random forests. Bioinformatics, 32(6), 859-866.
[7] Afanador, N. L., Smolinska, A., Tran, T. N., & Blanchet, L. (2016). Unsupervised random forest: a tutorial with case studies. Journal of Chemometrics, 30(5), 232-241.
[8] Jolliffe, I.T. Principal Component Analysis. Edisi kedua. Springer-Verlag. New York. 2002.
[9] Johnson dan Wichern. Applied Multivariate Statistical Analysis. Edisi keenam. Pearson Prentice Hall. 2007.
[10] Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn. 30 (7), 1145–1159.
[11] Hanley, J.A., McNeil, B.J., 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143, 29–36.
[12] Fawcet, Tom., 2005. An Introduction to ROC Analysis. Pattern Recognition Letters 27 (2006) 861–874
[13] Setiawan, N. A., Adji, T. B., M.,Galih Hendro. 2012. Penggunaan Metodologi Analisa Komponen Utama (PCA) untuk Mereduksi Faktor-Faktor yang Mempengaruhi Penyakit Jantung Koroner. Seminar Nasional ”Science, Engineering and Technology”.