

BAB V

PENGUJIAN DAN ANALISIS

5.1 Skenario Pengujian

Pengujian yang dilakukan untuk mengetahui pengaruh nilai k dan jumlah data latih terhadap nilai f -measure dengan menggunakan persamaan 2.11.

5.1.1 Skenario Pengujian Pengaruh Nilai K dan Data Latih

Pengujian yang dilakukan yaitu menguji nilai k yang baik untuk mengetahui tingkat resiko penyakit gagal ginjal. Pengujian ini dilakukan pada tiga jumlah data latih yang berbeda yaitu data latih 60, 90 dan 120. Sedangkan data yang diujikan bernilai sama yaitu berjumlah 30 data. Kemudian pada setiap data latih dilakukan pengujian sebanyak 10 kali, disesuaikan dengan jumlah nilai k yang ada. Setiap nilai k yang dimasukan, digunakan sebagai parameter pengujian. Setiap data latih, dilakukan pengujian yang sama dengan nilai k , sehingga dari pengujian tersebut dapat diketahui pengaruh nilai k terhadap tingkat Nilai f -measure sistem.

5.2 Pengujian Pengaruh Nilai K dan Pengaruh Jumlah Data Latih

Pada sub bab ini dibahas mengenai implementasi dari uji coba untuk sistem yang sudah dikembangkan sesuai dengan skenario pengujian yang ada.

Pada pengujian pengaruh nilai k akan menggunakan nilai k dalam rentang 1–10. Data latih yang digunakan, dimulai dari data 60, 90 dan 120, sedangkan data uji yang digunakan bernilai sama yaitu berjumlah 30 data. Tabel 5.1 menunjukkan nilai pengujian pengaruh nilai k dengan 60 data latih laki-laki dan tabel 5.2 pada data latih perempuan.

Tabel 5.1 Pengujian Pengaruh nilai K terhadap 60 Data Latih Laki-laki

Jumlah Data Latih	Nilai K	Nilai <i>f-measure</i>
60	1	0.7
	2	0.7
	3	0.8
	4	0.8
	5	0.8
	6	0.8
	7	0.8
	8	0.8
	9	0.8
	10	0.8

Pada tabel 5.1 dapat terlihat bahwa nilai *f-measure* terendah didapatkan pada nilai $k=1$ dan $k=2$ dengan nilai *f-measure* = 0.7, sedangkan untuk nilai *f-measure* tertinggi berada pada nilai $k=3$ sampai dengan $k=10$, dengan nilai *f-measure* = 0.8.

Tabel 5.2 Pengujian Pengaruh nilai K terhadap 60 Data Latih Perempuan

Jumlah Data Latih	Nilai K	Nilai <i>f-measure</i>
60	1	0.8
	2	0.8
	3	0.8
	4	0.8
	5	0.8
	6	0.8
	7	0.8
	8	0.8
	9	0.8
	10	0.8

Pada tabel 5.2 dapat terlihat bahwa nilai *f-measure* stabil yaitu 0.8 yang didapatkan pada nilai $k=1$ sampai dengan $k=10$.

Hasil dari pengujian data latih 90 ditunjukkan pada tabel 5.3 untuk data latih laki-laki dan tabel 5.4 untuk data latih perempuan.

Tabel 5.3 Pengujian terhadap 90 Data Latih Laki-laki

Jumlah Data Latih	Nilai K	Nilai f -measure
90	1	0.7
	2	0.7
	3	0.8
	4	0.8
	5	0.8
	6	0.8
	7	0.8
	8	0.8
	9	0.8
	10	0.8

Dari tabel 5.3 dapat diketahui nilai f -measure terendah berada pada nilai $k=1$ dan $k=2$ dengan nilai f -measure = 0.7, sedangkan untuk nilai f -measure tertinggi berada pada nilai $k=3$ sampai dengan $k=10$ dengan nilai f -measure = 0.8.

Tabel 5.4 Pengujian terhadap 90 Data Latih Perempuan

Jumlah Data Latih	Nilai K	Nilai f -measure
90	1	0.8
	2	0.8
	3	0.8
	4	0.8
	5	0.8
	6	0.8
	7	0.8
	8	0.8
	9	0.8
	10	0.8

Dari tabel 5.4 dapat terlihat bahwa nilai f -measure stabil yaitu 0.8 yang didapatkan pada nilai $k=1$ sampai dengan $k=10$. Selanjutnya, hasil dari pengujian data latih 120 ditunjukkan pada tabel 5.5 pada data latih laki-laki dan tabel 5.6 pada data latih perempuan.

Tabel 5.5 Pengujian terhadap 120 Data Latih Laki-laki

Jumlah Data Latih	Nilai K	Nilai f -measure
120	1	0.8
	2	0.8
	3	0.8
	4	0.8
	5	0.8
	6	0.8
	7	0.8
	8	0.8
	9	0.8
	10	0.8

Dari tabel 5.5 dapat terlihat bahwa nilai f -measure stabil yaitu 0.8 yang didapatkan pada nilai $k=1$ sampai dengan $k=10$.

Tabel 5.6 Pengujian terhadap 120 Data Latih Perempuan

Jumlah Data Latih	Nilai K	Nilai f -measure
120	1	0.8
	2	0.8
	3	0.8
	4	0.8
	5	0.8
	6	0.8
	7	0.8
	8	0.8
	9	0.8
	10	0.8

Dari tabel 5.6 dapat terlihat bahwa nilai f -measure stabil yaitu 0.8 yang didapatkan pada nilai $k=1$ sampai dengan $k=10$. Selanjutnya, hasil dari pengujian 60 data latih campur (laki-laki dan perempuan) ditunjukkan pada tabel 5.7, tabel 5.8 untuk 90 data latih campur, dan tabel 5.9 untuk 120 data latih campur.

Tabel 5.7 Pengujian terhadap 60 Data Latih Campur

Jumlah Data Latih Campur	Nilai K	Nilai <i>f-measure</i>
60	1	0.8
	2	0.8
	3	0.8
	4	0.8
	5	0.7
	6	0.7
	7	0.7
	8	0.8
	9	0.8
	10	0.8

Pada tabel 5.7 dapat terlihat bahwa nilai *f-measure* terendah didapatkan pada nilai $k=5$ sampai dengan $k=7$ dengan nilai *f-measure* = 0.7, sedangkan untuk nilai *f-measure* tertinggi berada pada nilai $k=1$ sampai dengan $k=4$ dan $k=8$ sampai dengan $k=10$, dengan nilai *f-measure* = 0.8.

Tabel 5.8 Pengujian terhadap 90 Data Latih Campur

Jumlah Data Latih	Nilai K	Nilai <i>f-measure</i>
90	1	0.8
	2	0.8
	3	0.8
	4	0.8
	5	0.8
	6	0.8
	7	0.7
	8	0.7
	9	0.7
	10	0.8

Pada tabel 5.8 dapat terlihat bahwa nilai *f-measure* terendah didapatkan pada nilai $k=7$ sampai dengan $k=9$ dengan nilai *f-measure* = 0.7, sedangkan untuk nilai *f-measure* tertinggi berada pada nilai $k=1$ sampai dengan $k=6$ dan $k=10$ dengan nilai *f-measure* = 0.8.

Tabel 5.9 Pengujian terhadap 120 Data Latih Campur

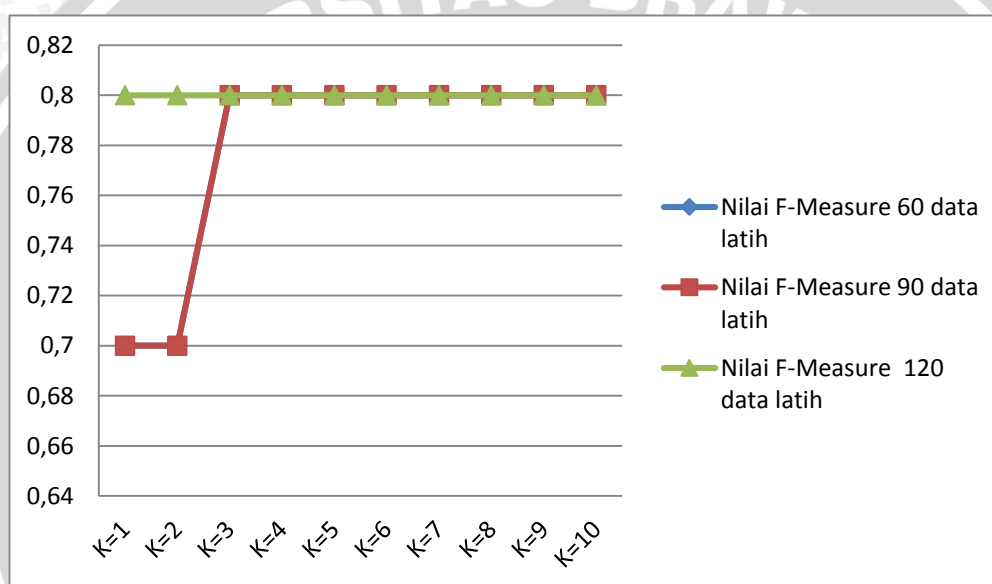
Jumlah Data Latih	Nilai K	Nilai f -measure
120	1	0.7
	2	0.7
	3	0.8
	4	0.8
	5	0.8
	6	0.8
	7	0.8
	8	0.8
	9	0.8
	10	0.8

Pada tabel 5.9 dapat terlihat bahwa nilai f -measure terendah didapatkan pada nilai $k=1$ sampai dengan $k=2$ dengan nilai f -measure = 0.7, sedangkan untuk nilai f -measure tertinggi berada pada nilai $k=3$ sampai dengan $k=10$ dengan nilai f -measure = 0.8.

5.3 Analisa Hasil

5.3.1 Analisa Hasil Pengaruh Nilai K terhadap Data Latih

Pada hasil uji coba menggunakan 60, 90, dan 120 pada data latih laki-laki yang terlihat pada tabel 5.1, tabel 5.3, dan tabel 5.5 bahwa nilai *f-measure* yang dihasilkan tiap-tiap nilai k semakin naik dan stabil. Untuk nilai *f-measure* tertinggi pada 60, 90, dan 120 data latih rata-rata berada pada nilai $k=10$. Berdasarkan tabel 5.1, tabel 5.3, dan tabel 5.5 dapat dibuat grafik hubungan antara nilai k dengan nilai *f-measure*. Grafik nilai ini ditunjukkan pada Gambar 5.1.

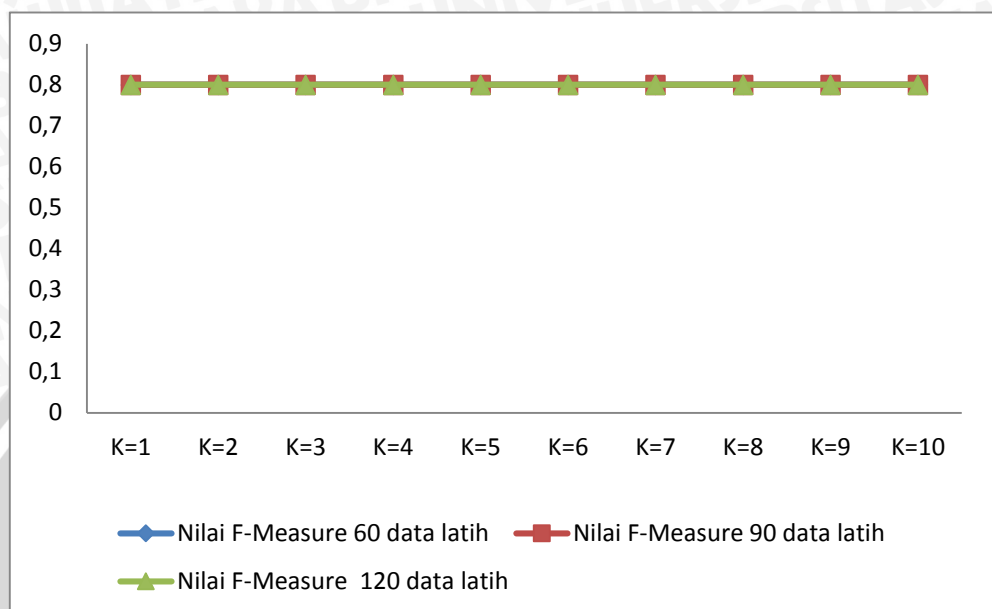


Gambar 5.1 Grafik Pengaruh Nilai K pada 60,90,120 Data Latih Laki-laki

Pada Gambar 5.1 terlihat nilai *f-measure* semakin naik dan stabil. Dimulai dari nilai nilai $k=1$ dan $k=2$, dimana nilai *f-measure*nya merupakan nilai terendah pada 60 dan 90 data latih. Nilai *f-measure* tertinggi rata-rata berada pada nilai $k=10$ dengan nilai *f-measure* = 0.8. Nilai $k=3$ sampai dengan $k=10$ memiliki nilai tetangga terdekat dengan data uji yang jumlahnya lebih banyak dibandingkan dengan nilai $k=1$ dan $k=2$, hal inilah yang menyebabkan nilai membershipnya naik, dan mempengaruhi kestabilan nilai *f-measure*.

Pada hasil uji coba menggunakan 60, 90, dan 120 pada data latih perempuan yang terlihat pada tabel 5.2, tabel 5.4, dan tabel 5.6 bahwa nilai *f-measure* yang

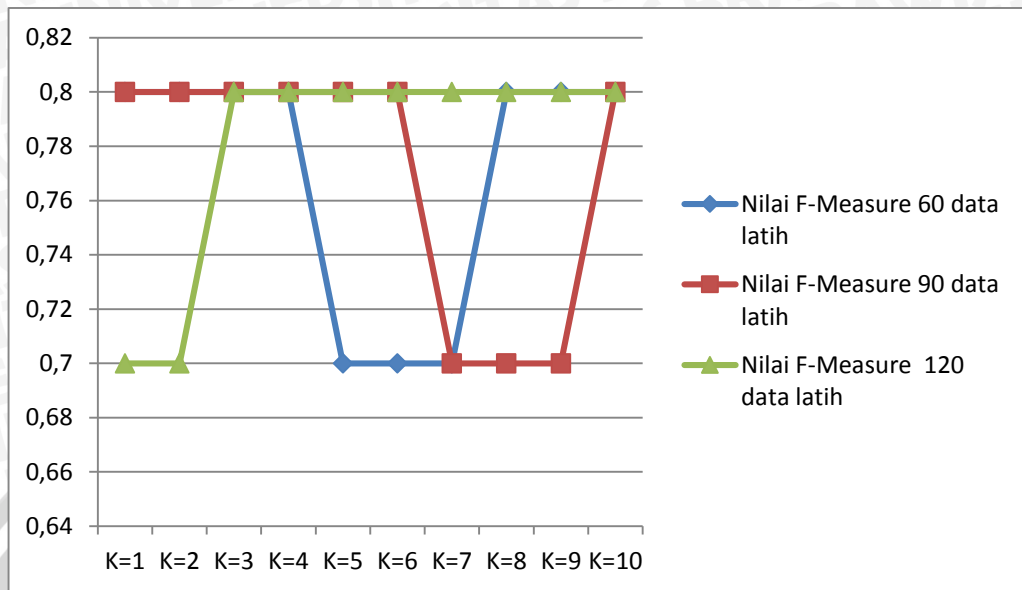
dihasilkan tiap-tiap nilai k adalah stabil tidak pernah naik dan turun. Grafik nilai ini ditunjukkan pada Gambar 5.2



Gambar 5.2 Pengaruh Nilai K terhadap 60,90,120 Data Latih Perempuan

Pada Gambar 5.2 terlihat nilai *f-measure* stabil tidak pernah naik dan tidak pernah turun. Dimulai dari nilai k=1 sampai dengan k=10, dimana nilai *f-measure*nya yaitu 0.8.

Pada hasil uji coba menggunakan 60, 90, dan 120 pada data latih campur yaitu perempuan dan laki-laki yang terlihat pada tabel 5.7, tabel 5.8, dan tabel 5.9 bahwa nilai *f-measure* yang dihasilkan tiap-tiap nilai k adalah nail turun pernah naik dan turun. Grafik nilai ini ditunjukkan pada Gambar 5.3



Gambar 5.3 Pengaruh Nilai K terhadap 60,90,120 Data Latih Campur

Pada Gambar 5.3 terlihat nilai *f-measure* naik turun pada data latih 60 $k=1$ sampai $k=4$ nilai *f-measure* stabil berada pada angka 0.8 kemudian pada $k=5$ sampai dengan $k=7$ nilai *f-measure* turun menjadi 0.7 dan naik lagi pada $k=8$ sampai $k=10$. Pada data latih 90 $k=1$ sampai $k=6$ nilai *f-measure* berada pada angka 0.8 namun pada $k=7$ sampai dengan $k=9$ nilai *f-measure* turun menjadi 0.7 dan naik lagi pada $k=10$. Pada latih 120 nilai *f-measure* terus naik yaitu dimulai dari $k=1$ dan $k=2$ dengan nilai *f-measure* = 0.7 kemudian $k=3$ sampai $k=10$ nilai *f-measure* = 0.8.

Dapat diartikan bahwa, jumlah data latih yang semakin bertambah sangat mempengaruhi nilai terendah dari nilai *f-measure*. Pengaruh k juga mempengaruhi hasil nilai *f-measure*, dimana pada $k=10$ nilai *f-measure* tertinggi, dan $k=1$ nilai *f-measure* terendah, dan ditambahkan, semakin banyak nilai k maka semakin memperkecil *noise* yang ada sehingga nilai *f-measure* semakin tinggi. Banyaknya jumlah data juga mempengaruhi besar nilai *noise* yang dihasilkan, yang disebabkan oleh banyaknya variasi data yang ada.

BAB VI

PENUTUP

6.1 Kesimpulan

Kesimpulan yang didapatkan pada penelitian ini adalah sebagai berikut :

1. Metode *Fuzzy K-Nearest Neighbor* dapat digunakan untuk memprediksi tingkat resiko penyakit gagal ginjal dengan 6 parameter yaitu umur, kadar urea, kadar BUN, kadar kreatinin, kadar *uric acid* dan kadar trigliserida. Hasil yang didapat dari pengelompokkan pada penelitian ini adalah nilai *f-measure* dari prediksi tiap data ke dalam data penyakit gagal ginjal.
2. Hasil perhitungan nilai *f-measure* prediksi tertinggi yang didapat yaitu saat nilai $k=10$ dengan nilai *f-measure* = 0.8 dari sebaran data latih 60, 90, maupun 120.
3. Nilai *f-measure* untuk data perempuan stabil yaitu 0.8 dikarenakan data latih pada perempuan lebih bagus dibandingkan data latih pada laki-laki.

6.2 Saran

Beberapa saran untuk pengembangan lebih lanjut dapat diberikan oleh penulis adalah :

1. Melakukan prediksi tingkat resiko penyakit gagal ginjal dengan data latih yang lebih banyak dan data latih yang lebih optimal untuk mengetahui apakah sistem memiliki nilai *f-measure* yang lebih baik.
2. Menggunakan data yang tidak acak dan nilai *range* yang tidak terlalu dekat agar mendapat nilai *f-measure* yang tinggi.

DAFTAR PUSTAKA

- [ABR-13] Abraham, Nelson. 2013. *"Penyakit Gagal Ginjal"*. <http://www.direktori-artikel.web.id/2013/08/penyakit-gagal-ginjal.html> [15 Februari 2014].
- [ARA – 2010] Arapoglou, Roi, Kostas Kolomvatsos and Stathes Hadjiefthymiades. 2010. *Buyer Agent Decision Process Based on Automatic Fuzzy Rules Generation Methods*. http://p-comp.di.uoa.gr/pubs/WCCI_f427.pdf.
- [ARD-11] Ardra. 2011. *"Penyakit Asam Urat"*. <http://ardra.biz/kesehatan/asam-urat/penyakit-asam-urat-uric-acid/> [3 Maret 2014].
- [FIR-12] Firmansyah, Ricky Yudha, Diana Rahmawati, Haryanto. 2012. *Sistem Pakar Diagnosis Penyakit Ginjal Menggunakan Metode Naïve Bayes Classifier*. Skripsi. Malang.
- [FLO-11] Florin Gorunescu, *Data Mining: Concepts, Models and Techniques*, Springer, 2011.
- [FUA-13] Fuadi, Taufik. 2013. *Accuracy Measure: Precision, Recall & F-Measure*. Bahan Kuliah Data Mining: FMIPA Universitas Syiah Kuala.
- [HER-08] Hermaduanty, Ninki. 2008. *Sistem Pendukung Keputusan Berbasis SMS untuk Menemukan Status Gizi dengan Metode K-Nearest Neighbor*. Skripsi. Yogyakarta.
- [JIA-06] Jiawei Han and Micheline Kamber. 2006. *Data Mining: Concepts and Techniques* Second Edition. Elsevier.
- [KEL – 85] Keller, M. James, Michael R Gray, James A. Givens. 1985. *A Fuzzy K-Nearest Neighbor*. IEEE Transactions On Sistem, Man And Cybernetics, Vol. SMC-15 NO 4.

- [KUS-03] Kusumadewi, Sri. 2003. *"Artificial Intelligence (Teknik dan Aplikasinya)"*. Graha Ilmu.Yogyakarta.
- [KUS-10] Kusumadewi, Sri.Purnomo, Hari. 2004. Aplikasi Logika Fuzzy untuk Pendukung Keputusan. Graha Ilmu. Yogyakarta.
- [NIL – 96] J.Nilsson, Nill. *"Introduction To Machine Learning"*. 1996. Stanford University: Stanford, CA 94305.
- [NUS – 98] Nuswantari. 1998. *"Pengertian Umur"*. <http://digilib.unimus.ac.id/files/disk1/121/jtptunimus-gdl-sitituslih-6010-2-babii.pdf>.
- [OLV-11] Olvista. 2011. *"Pengertian Tligiserida"*. <http://olvista.com/kesehatan/apa-itu-triglicerida/> [4 Maret 2014].
- [PRA-12] Prasetyo, Eka. 2012. Fuzzy K-NN In Every Class Untuk Klasifikasi Data. Universitas Pembangunan Nasional.
- [PRI – 05] Priyono, Agus Priyono, Muhammad Ridwan, Ahmad Jais Alias, Riza Atiq, O. K. Rahmat, Azmi Hassan & Mohd. Alauddin Mohd. Ali. 2005. *Generation Of Fuzzy Rules With Subtractive Clustering*, Jurnal Teknologi, 43(D) Dis. Universitas Teknologi Malaysia.
- [PUS-08] Puspasari, Maslikha. 2008. *Prediksi Tingkat Resiko Penyakit Jantung Koroner (PJK) Menggunakan Metode Fuzzy K-Nearest Neighbor (FK-NN)*. Skripsi. Malang.
- [RIN – 06] Rindiastuti. 2006. *"Gagal Ginjal"*. http://www.simassehat.com/Info_sehat/Seminar/Gagal%20Ginjal%20Kronik.pdf.

- [SPI-13] Spiritia, Yayasan. 2013. *“Tes Fungsi Ginjal”*. <http://spiritia.or.id/li/pdf/LI136.pdf>.
- [TAL-03] Tala, Fadillah Z. 2003. *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. Master of Logic Project. Institute for Logic, Language and Computation. Universiteit van Amsterdam. The Netherlands.
- [THO-07] Thomas, Dileep. 2007. *“Data Mining Classification”*. <http://www.dataminingarticles.com/info/classification/> [12 Oktober 2014].
- [VIT-08] Vitahealth. 2008. *“Gagal Ginjal”*. <http://gagal-ginjal.com/> [3 Maret 2014].
- [WAR-13] Warianto, Chaidar. 2011. *“Gagal Ginjal”*. http://skp.unair.ac.id/repository/Guru-Indonesia/GagalGinjal_ChaidarWarianto_20.pdf.
- [WIJ-13] Wijaya, Mohammad Tri. 2013. *Implementasi Fuzzy K-Nearest Neighbour (FK-NN) untuk Prediksi Kelas Kebakaran Hutan*. Skripsi. Malang.
- [WIS-13] Wisdaryanto, Ardhy. 2013. *Penerapan Metode Fuzzy K-Nearest Neighbor (FK- NN) untuk Pengklasifikasian Spam Email*. Skripsi. Malang.
- [ZHA-09] Zhang, Juan, Yi Niu, dan Huabei Nie. 2009. *“Web Document Classification Based on Fuzzy KNN Algorithm”*. International Conference on Computational Intelligence and Security.

LAMPIRAN A

HASIL WAWANCARA INFORMAN

(Tenaga Kesehatan Laboratorium Klinik Sejahtera Kota Probolinggo)

1. Nurul Maghfirah : Di klinik ini penyakit apa yang sering dialami pasien setelah melakukan tes darah ?
2. Tenaga Kesehatan : Ya banyak mbak, cuman penyakit yang sering dialami pasien dari sepuluh penyakit terbanyak yaitu penyakit gagal ginjal.
3. Nurul Maghfirah : Lalu bagaimana cara mengidentifikasi pasien terkena penyakit gagal ginjal pak ?
4. Tenaga Kesehatan : Untuk menentukan identifikasinya dengan melihat nilai dari kadar urea, kadar BUN, kadar kreatinin, kadar *uric acid* dan kadar trigliserida yang diperoleh dari hasil tes darah pasien.
5. Nurul Maghfirah : Lalu bagaimana caranya melihat bahwa pasiennya memiliki gejala penyakit gagal ginjal ?
6. Tenaga Kesehatan : Begini, untuk menentukan pasien terserang gagal ginjal itu ada 3 tingkatan yaitu tingkatan pertama normal dengan hasil tes darah pasien tidak melebihi batas normal. Tingkatan kedua yaitu sedang pasien dikatakan beresiko mengalami penyakit gagal ginjal namun dalam stadium yang ringan. Tingkatan ketiga yaitu tinggi pasien mengalami gagal ginjal dalam stadium yang berat dan hasil tes darahnya melebihi batas normal.
7. Nurul Maghfirah : Bagaimana caranya agar kita mengetahui bahwa pasien termasuk tingkatan normal, sedang dan tinggi, apakah ada batasan-batasannya?

8. Tenaga Kesehatan : Iya pasti ada, dengan melihat batasan nilai yang sudah ditentukan pada masing-masing kadar yang digunakan untuk diagnosa penyakit gagal ginjal. Kadar yang saya maksud disini yaitu ada kadar urea, kadar BUN, kadar kreatinin, kadar *uric acid* dan kadar trigliserida. Berikut ini batasan-batasan nilai untuk mendiagnosa penyakit gagal ginjal.

1. Untuk Laki-laki :

Kadar Urea	: Normal	: 10-49 mg/dl
	Sedang	: 50-60 mg/dl
	Tinggi	: lebih dari 61 mg/dl
Kadar BUN	: Normal	: 5-22 mg/dl
	Sedang	: 23-32 mg/dl
	Tinggi	: lebih dari 33 mg/dl
Kadar Kreatinin	: Normal	: 0.7-1.2 mg/dl
	Sedang	: 1.3-1.6 mg/dl
	Tinggi	: lebih dari 1.7 mg/dl
Kadar Uric Acid	: Normal	: 3.4-6.9 mg/dl
	Sedang	: 7.0-7.5 mg/dl
	Tinggi	: lebih dari 7.5 mg/dl
Kadar Trigliserida	: Normal	: < 200 mg/dl
	Sedang	: 201-250 mg/dl
	Tinggi	: > 250 mg/dl

2. Untuk Perempuan :

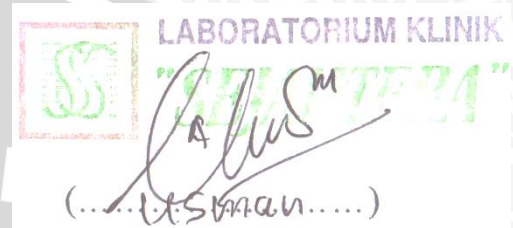
Kadar Kreatinin	: Normal	: 0.6-1 mg/dl
	Sedang	: 1.1-1.4 mg/dl
	Tinggi	: > 1.4 mg/dl
Kadar Uric Acid	: Normal	: 2.4-5.6 mg/dl
	Sedang	: 5.7-6.0 mg/dl
	Tinggi	: lebih dari 6.0 mg/dl

9. Nurul Maghfirah : Berdasarkan data batasan nilai tersebut lalu kadar apa yang paling mempengaruhi pasien yang menderita penyakit gagal ginjal pak, atau apakah semua berpengaruh ?

10. Tenaga Kesehatan : Sebenarnya tidak semua kadar yang saya sebutkan tadi berpengaruh untuk mengidentifikasi pasien menderita penyakit gagal ginjal, tetapi kita juga harus tau semua kadar yang telah ditentukan sesuai dengan prosedur namun ada kadar dalam darah yang sangat berpengaruh dalam menentukan diagnosa penyakit gagal ginjal ada 3 yaitu kadar urea, kadar kreatinin, dan kadar *uric acid*. Apabila diketahui kadar urea seseorang melebihi batas normal, namun kadar BUN, kadar kreatinin, kadar *uric acid* dan kadar trigliserida normal, maka seseorang tersebut dapat dikatakan dalam keadaan normal dan sehat. Namun apabila diketahui kadar urea, kadar kreatinin dan kadar *uric acid* seseorang melebihi batas normal maka

orang tersebut dapat dikatakan menderita penyakit gagal ginjal stadium berat.

Narasumber : Usman



LAMPIRAN B

Data Penyakit Gagal Ginjal

a. Data Penyakit Gagal Ginjal Untuk Data Laki-Laki Tahun 2013

No.	Nama	Umur	Urea	BUN	Kreatinin	Uric Acid	Trigliserida	Tingkat Resiko
1.	Tuji	69	28.9	16.2	1	5.1	155	1
2.	Yusuf	54	33.2	16.7	0.8	4.5	167	1
3.	Jamal	59	10	6	0.7	3.5	145	1
4.	Legawa	56	17.2	7	0.7	3.4	145	1
5.	Solihin	48	53	24.5	1.5	7.2	203	2
6.	Adi Susanto	60	12.3	6	0.8	5	119	1
7.	Abdi Walman	55	17.8	14.8	1	3.8	150	1
8.	Tri Joko	40	15.5	7.2	0.7	5.3	156	1
9.	Masyfufah	62	56	25.3	1.4	7.1	243	2
10.	Pudiono	50	20.8	20.4	0.9	5.5	152	1
11.	Mukhlis	42	68.2	35.6	1.8	7.6	252	3
12.	Sugiatno	70	55	24.4	1.6	7.3	203	2
13.	Yanto Purnomo	49	27.1	15.6	0.9	4.9	187	1
14.	Sutopo	58	56.3	24.1	1.4	7.1	205	2
15.	Gofur	56	53.8	26.8	1.5	7.2	209	2
16.	Arif Budi	39	64.2	35.1	1.9	7.6	278	3
17.	Rifki	48	16.9	21.3	1.1	3.7	165	1
18.	Wan Fiqih	65	52.8	26.2	1.4	7.3	249	2
19.	Ridwan	46	53.2	30.3	1.5	7.3	202	2
20.	Eko	49	65	35.2	1.8	7.7	257	3
21.	Deni	43	33.6	15.5	0.9	5.7	161	1
22.	Fahmi	45	53.4	24.7	1.5	7.1	209	2
23.	Rizal	58	57.2	26.8	1.6	7.3	201	2
24.	Imam P.	33	67.2	34.9	1.9	7.9	293	3
25.	Zakfar	50	50.3	25.3	1.5	7.2	208	2
26.	Agus	41	62.9	36.8	1.8	7.9	275	3
27.	Tan Kim Nio	91	63.5	36.1	1.9	7.6	298	3
28.	Suwardi	39	34.5	10.6	0.9	5.8	169	1
29.	Indra W.	60	63.2	34.9	1.8	10.1	265	3
30.	Arifin	67	54.8	32.6	1.4	7.1	204	2
31.	Won Fie Lan	95	30.5	16.5	0.9	5.2	156	1
32.	Ferdinan Susanto	62	55.6	28.3	1.3	47.2	224	2
33.	Hermanto	47	43.2	33.7	1.1	3.9	155	1
34.	Said	55	61.2	35.1	1.8	8.3	289	3

No.	Nama	Umur	Urea	BUN	Kreatinin	Uric Acid	Trigliserida	Tingkat Resiko
35.	Alex	40	55.8	27.3	1.5	7.4	208	2
36.	Angga	58	57.2	24	1.4	7.1	222	2
37.	Yogi	55	44.2	8.5	1.1	3.8	177	1
38.	Samian	57	57	25.4	1.4	7.2	232	2
39.	Agus W.	44	62.3	39.3	1.8	8.7	319	3
40.	Ismarta	60	63.6	34.9	1.8	7.8	276	3
41.	Gatot	45	62.1	35.2	1.9	8.7	291	3
42.	Eko Winanto	37	11.2	20	0.7	6	123	1
43.	Agus Asmani	42	63.6	33.5	1.9	9.4	285	3
44.	Suprianto	48	61.9	38.2	1.9	7.9	254	3
45.	Sony W	50	23.9	17	6.1	5.5	160	1
46.	Maskur	65	55.9	28.4	1.3	7.3	223	2
47.	Sholeh	57	38.5	15.8	1.2	5.8	162	1
48.	Gozali	57	64.2	34.8	1.8	7.9	361	3
49.	Fajar	54	59.2	24.5	1.3	7.4	213	2
50.	Sunardi	57	14.8	18.7	0.9	5.1	157	1
51.	Pujiasih	68	52.4	33.8	1.5	7.2	206	2
52.	Misnadi	56	17.3	15.6	1.1	6.2	188	1
53.	Tunggul Hadi	81	56.3	25.4	1.4	7.2	201	2
54.	Gandi	46	53.2	24.8	1.6	7	201	2
55.	Sugeng	56	62.8	37.2	1.8	7.9	276	3
56.	Johan	53	18.2	6.9	0.8	6	154	1
57.	Dawanah	53	66.2	38.4	1.9	8.7	274	3
58.	Samsuri	30	38.9	18.9	1.1	5.9	159	1
59.	Ananta Wijaya	49	55.2	28.3	1.4	7	205	2
60.	Rafi	76	57.2	27.8	1.5	7.3	204	2
61.	Paiman	54	56.8	29.6	1.5	7.2	209	2
62.	Sodik	57	43.2	16.2	0.7	3.4	150	1
63.	Soleh	67	52.8	27.3	1.6	7.5	204	2
64.	Anton	55	53	26.4	1.5	7	201	2
65.	Suhari	47	16.2	20.9	0.7	3.4	134	1
66.	Harjanto	69	55.8	29	1.3	7.1	205	2
67.	Yayok	52	62.6	35.3	1.8	7.9	257	3
68.	Sugianto	60	26.9	19.8	0.9	5.1	153	1
69.	H. Suyono	54	21.4	15.4	0.9	4.5	167	1
70.	Fauzi	32	54.9	26.9	1.5	7	242	2
71.	Taji	41	58.7	34.9	1.9	7.2	221	2
72.	Kresno	49	65.9	1.8	7.6	7.6	265	3
73.	Hidayat	71	57.2	25.3	1.4	7.1	201	2

No.	Nama	Umur	Urea	BUN	Kreatinin	Uric Acid	Trigliserida	Tingkat Resiko
74.	Supi	78	36.1	17.8	1	5.6	158	1
75.	Djohan	75	50.8	26.3	1.3	7	250	2
76.	Anang	35	66.3	35.3	1.9	8	289	3
77.	Budi	57	55.2	26.3	1.6	7.5	204	2
78.	Saiful	47	53.3	24.6	1.5	7	206	2
79.	Samsul	66	61.6	34.1	1.9	8.7	264	3
80.	Eddy	45	20.4	15.9	0.7	3.2	167	1
81.	Djoko	59	58.2	26.3	1.5	7.2	245	2
82.	Andik	53	62.3	33.9	1.8	8.6	252	3
83.	Darmawan	51	65.9	33.8	1.8	7.7	280	3
84.	Hariyono	58	55.9	26.7	1.5	7.2	209	2
85.	Siswanto	53	68.2	37.2	1.9	7.5	258	3
86.	Iwan	28	62.8	34.6	1.9	7.9	256	3
87.	Sunar	51	66.3	38.2	1.9	8	259	3
88.	Karto Wibowo	70	67.2	35.2	1.8	7.9	280	3
89.	Yudi	32	63.1	35.1	1.8	9.8	258	3
90.	Poniman	73	82.5	36.7	2	9.3	276	3
91.	Maksudi	68	65.1	34.1	1.9	10.1	256	3
92.	Hermanto	62	26.4	15.4	0.9	4.7	167	1
93.	Moh Bisri	61	62.7	38.3	2.1	7.8	277	3
94.	H. Nasir	49	30.8	17.5	0.8	3.5	159	1
95.	Rofiq	58	69.1	39.3	1.8	8.3	255	3
96.	Rosi	46	12.5	21.8	0.9	3.8	156	1
97.	Ainur	55	50.3	25.4	1.5	7	201	2
98.	Hendris	54	42.1	21.3	0.7	1.1	134	1
99.	Putra	40	51.6	24	1.5	7	201	2
100.	Rubi	23	26.8	16.3	0.7	3.6	154	1
101.	Sanusi	58	62.9	34.9	2	8.1	260	3
102.	Askandar	66	43.8	15.7	1.2	3.2	152	1
103.	Yohanes	33	47.7	17.3	1.1	5.1	159	1
104.	Sumarjono	64	45	62.8	34.4	1.8	260	3
105.	Mashuri	55	52.4	24.3	1.6	7.3	204	2
106.	Tunggal Hadi	80	62.9	34.6	1.9	7.9	267	3
107.	Ilyas	45	23.4	14.7	0.8	3.8	164	1
108.	Bowo	61	62.8	33.2	1.9	7.8	251	3
109.	Adi Hartanto	53	54.3	26.3	1.3	7	214	2
110.	Soni	60	35.2	16.9	1.2	5.8	158	1
111.	Balok Budiarto	66	18.5	16	1.1	4.4	159	1
112.	Suryadi	57	40.2	5	0.7	6.4	109	1

No.	Nama	Umur	Urea	BUN	Kreatinin	Uric Acid	Trigliserida	Tingkat Resiko
113.	Jinoto	50	62.8	37.2	1.9	7.7	278	3
114.	Sugiharto	52	66.7	33	3	8	254	3
115.	Nasir	49	62.8	37.2	1.9	8.7	461	3
116.	Wong Nyuk	70	11.2	15.2	1	3.5	102	1
117.	Harto	41	67.2	36.3	1.9	7.9	281	3
118.	Fuad	46	56.2	27.3	1.6	7.2	234	2
119.	Antok	47	53.5	27.1	1.5	7	214	2
120.	Wiyanto	62	63.1	33.9	1.8	10.5	268	3
121.	Suherman	60	67.8	35.2	1.8	7.6	278	3
122.	Abdullah	42	34.2	16.2	0.7	6.3	169	1
123.	Sumar	53	56.2	25.2	1.4	7	203	2
124.	Nawar	56	32.8	12.7	0.8	4.9	154	1
125.	Purwoto	50	50.9	27.8	1.4	7.1	245	2
126.	Candra	28	42.9	21.5	1.1	6.2	132	1
127.	Parman	70	66.4	35.3	3.2	8.6	347	3
128.	Moch. Zuber	29	52.1	27.2	1.3	7	201	2
129.	Yudianto	56	66.3	35.2	1.8	7.5	255	3
130.	Ardiek	29	24.2	17.2	0.7	6.1	165	1
131.	Fauzan	55	52.8	24.6	1.5	7.3	211	2
132.	Sauri	62	42.8	12.6	0.7	5	167	1
133.	Basri	52	18.9	16.2	0.7	3.6	169	1
134.	Silaturrahman	41	69.7	36.3	1.8	9.4	268	3
135.	Jamili	49	55.9	24.1	1.6	7.1	202	2
136.	Agung	23	12.6	6.9	0.7	1	105	1
137.	Gunawan	44	58.8	33.4	1.8	8.1	316	3
138.	Achmad Taufik	44	62.7	33.8	1.8	9.4	265	3
139.	Sunarya	54	53.8	24.4	1.3	7.2	209	2
140.	Ari	46	25.7	12.4	1	5.3	156	1
141.	Subani	58	61.5	34.8	1.8	9.7	267	3
142.	Agus Edi	25	64.2	35.9	1.9	8.6	279	3
143.	Yudo	70	57	28.7	1.4	7	201	2
144.	Yuwar	51	53.9	28.7	1.3	7.3	204	2
145.	Sumai	50	12.8	20.1	1	6.3	108	1
146.	Zulkifli	50	54.9	23.2	1.5	7	242	2
147.	Bachriansyah	48	68.4	35.5	1.9	7.7	256	3
148.	Subali	56	57.7	24.9	1.3	7.3	240	2
149.	Hadi	52	11.1	18.3	0.7	5.7	168	1
150.	Soekresno	49	64.4	35.4	1.8	7.6	254	3

Jumlah Data : 150

Keterangan Tingkat Resiko :

- 1 = Normal
- 2 = Sedang
- 3 = Tinggi

Urea :

- Normal : 10-49 mg/dl
- Rendah : 50-60 mg/dl
- Tinggi : lebih dari 61 mg/dl

BUN :

- Normal : 5-22 mg/dl
- Rendah : 23-32 mg/dl
- Tinggi : lebih dari 33 mg/dl

Kreatinin :

- Normal laki-laki : 0.7-1.2 mg/dl
- Rendah laki-laki : 1.3 - 1.6 mg/dl
- Tinggi laki-laki : Lebih dari 1.7 mg/dl

Uric Acid :

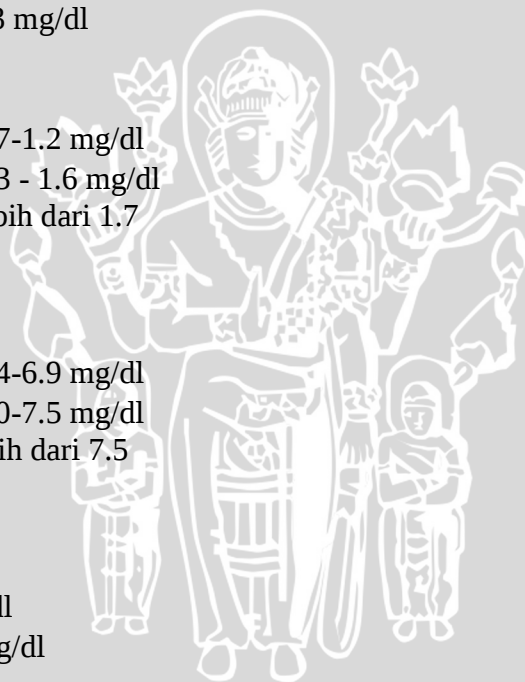
- Normal laki-laki : 3.4-6.9 mg/dl
- Rendah laki-laki : 7.0-7.5 mg/dl
- Tinggi laki-laki : lebih dari 7.5 mg/dl

Trigliserida :

- Normal : < 200 mg/dl
- Rendah : 201-250 mg/dl
- Tinggi : > 250 mg/dl

Umur :

- Muda : < 35 tahun
- Parobaya : 35 - 55 tahun
- Tua : > 55 tahun



b. Data Penyakit Gagal Ginjal untuk Perempuan Tahun 2013

No.	Nama	Umur	Urea	BUN	Kreatinin	Uric Acid	Trigliserida	Tingkat Resiko
1.	Qodriyah	55	62	34.8	1.5	6.2	256	3
2.	Ani Rahayu	59	30	12.6	0.9	3.7	153	1
3.	Nurul	39	55.2	23.9	1.1	6	250	2
4.	Atminah	79	33.8	15.7	0.7	5.6	162	1
5.	Ismaria	48	63.8	35.8	1.5	6.4	255	3
6.	Atiek	65	54.6	23	1.2	6	234	2
7.	Faida	52	65.2	41.3	1.5	7	256	3
8.	Kristin	45	62.1	38.2	1.4	7.1	289	3
9.	Masyudah	59	40.2	13.7	0.9	4.7	156	1
10.	Reni	47	64.1	38.1	1.6	7	299	3
11.	Banun	53	68.1	35.6	1.7	8	288	3
12.	Paulina	56	55.9	26.3	1.1	6	235	2
13.	Yeni	54	54.6	27.3	1.3	6	236	2
14.	Zulfa	59	64.2	36.9	1.5	7	253	3
15.	Resti	42	66.7	35.6	1.5	8	291	3
16.	Ratna Tri	47	43	21.2	0.8	5	155	1
17.	Fani	36	52.1	23.5	1.2	5.8	202	2
18.	Amalia	48	24.1	14.8	0.9	4.1	167	1
19.	Silvi	45	68.9	36.2	1.5	7	298	3
20.	Mafia	48	66.6	37.1	1.8	8.2	255	3
21.	Tutug	48	57.8	25.7	1.2	5.8	218	2
22.	Farida	71	49.7	23.1	1	5.4	159	1
23.	Nuning	39	52.9	28.6	1.4	5.9	212	2
24.	Asmuni	59	34.7	17.6	0.9	4.5	135	1
25.	Rusiati	63	61.4	33.4	1.6	7	263	3
26.	Isro'atul	46	55.6	28.7	1.1	5.9	203	2
27.	Putri	39	36.7	14.8	0.7	5.3	175	1
28.	Yulia	46	65.2	35.9	1.6	7.8	288	3
29.	Titik	47	58.5	23.3	1.3	5.6	201	2
30.	Asmiati	48	63.2	27.9	1.2	5.8	219	2
31.	Sumarsih	53	65.9	35.3	1.5	6.8	319	3
32.	Fitria	49	54.6	26.5	1.1	5.9	249	2
33.	Yunda	52	62.8	36.3	1.6	7.2	256	3
34.	Zubaidah	56	62.5	37.8	1.5	7	256	3
35.	Nurul	49	22.9	10.6	0.7	3.6	153	1
36.	Lely	38	55	28.7	1.2	6	209	2
37.	Fatimah	52	21.1	17.5	0.9	5.1	157	1
38.	Ami Khasanah	49	55.1	26.4	1.3	6	208	2

No.	Nama	Umur	Urea	BUN	Kreatinin	Uric Acid	Trigliserida	Tingkat Resiko
39.	Khotimatul	58	62.9	33.6	1.7	6	257	3
40.	Husna	55	61.2	35.1	1.5	7.4	253	3
41.	Khoiriyah	53	63.6	37.2	1.5	8.2	267	3
42.	Holila	34	21.3	15	0.7	5.3	158	1
43.	Siti	47	68.2	36	1.5	7.6	278	3
44.	Salamah	55	62.8	33.1	1.7	8.9	289	3
45.	Endang Sinaningrum	72	18.8	10.6	0.8	3.6	159	1
46.	Rusmiati	47	22.6	15.8	0.8	4.2	147	1
47.	Ninik	53	32.8	17.4	0.9	3.4	154	1
48.	Halimah	85	68.6	37.8	1.5	6.8	261	3
49.	Nurhayati	53	33.6	20.1	0.8	5	135	1
50.	Devi	43	62.7	34.2	1.6	7.5	267	3
51.	Kusmiati	56	64.8	35	1.5	7	298	3
52.	Susana	58	10	7	1	2.5	109	1
53.	Pujiana	45	52.4	29.2	1.3	6	234	2
54.	Astutik	56	12.8	20.9	0.6	5	119	1
55.	Sallyfallin	59	36.8	16.5	0.7	4.8	156	1
56.	Fenny Margaretta	61	55	25.6	1.4	5.9	213	2
57.	Kartika	35	62.6	37.9	1.6	6.3	254	3
58.	Erni	45	26.7	17.6	0.8	5	167	1
59.	Umi Salamah	47	53.9	29.5	1.4	5.9	222	2
60.	Sri Rahayuningsih	41	51.7	27.8	1.1	6	209	2
61.	Nanik	57	63.8	36.1	1.5	7.4	256	3
62.	Sindy	45	27.9	8.9	1	3.9	132	1
63.	Sherly	52	51.5	25.3	1.3	6	219	2
64.	Tatik	63	16.3	16.2	0.6	5.1	101	1
65.	Pujiasih	68	56.5	25.8	1.3	6	228	2
66.	Niken	45	62.8	37.3	1.5	7.5	290	3
67.	I gustining	55	66.8	33.9	1.6	7	267	3
68.	Lili Harmini	62	46.2	17.8	1	5.4	157	1
69.	Lila	49	33.2	16.9	0.9	5.6	159	1
70.	Laila	39	33.9	22	0.8	4.6	165	1
71.	Jamaliyah	47	64.3	35.3	1.5	6.8	259	3
72.	Ayu	55	52.6	26.7	1.4	6	244	2
73.	Susan Budiman	42	38.2	10.7	0.7	3.9	156	1
74.	Wahyuni	55	67	34.9	1.5	6.4	298	3
75.	Magda	36	59.7	28.5	1.3	5.9	218	2

No.	Nama	Umur	Urea	BUN	Kreatinin	Uric Acid	Trigliserida	Tingkat Resiko
76.	Tutik Hermawati	46	17.2	8.2	1	0.8	101	2
77.	Nora	51	65.2	37.2	1.6	7.2	291	3
78.	Marpuah	52	15.8	18.7	1	5	154	1
79.	Nurhasanah	57	55.6	27.6	1.2	5.7	230	2
80.	Anni	60	44.2	15.4	1	5.3	159	1
81.	Buyah	38	68.2	34.5	1.5	6.4	316	3
82.	Dyas	56	53.5	28.3	1.2	6	207	2
83.	Ulfa	39	63.1	33.5	1.6	7.2	265	3
84.	Reysa	38	55.2	27.6	1.2	5.7	209	2
85.	Nindi	45	53.2	23.6	1.3	6	228	2
86.	Sri Ratnaningsih	56	24.8	14.7	0.8	5.6	154	1
87.	Sumiati	54	33.9	15.7	0.7	4	154	1
88.	Dannia	45	66.8	35	1.6	7	267	3
89.	Robbiatus	48	63.6	39.2	1.5	7.2	287	3
90.	Meivita	46	24.4	16.8	1	5.6	159	1
91.	Salimah	50	38.1	17.3	0.9	3	164	1
92.	Cicik	34	26.9	15.2	0.9	4.9	156	1
93.	Wiwik	43	25.6	17.1	0.8	4.1	169	1
94.	Ana Cornelia	40	63.6	35.4	1.8	6.3	268	3
95.	Sri Astutik	38	33.8	15.9	1	5.2	167	1
96.	Yayumi	46	65.3	38.2	1.6	6.8	278	3
97.	Rini	29	53.8	27.6	1.3	5.8	226	2
98.	Rosita	52	52.6	25.4	1.1	6	225	2
99.	Nining	51	57.7	27.9	1.4	5.9	236	2
100.	Ninik	46	63.3	35.9	1.6	6.5	260	3
101.	Halila	52	62.7	27.5	1.5	6.1	265	3
102.	Lily K.	40	12.9	20.8	1	2.9	129	1
103.	Yuni	45	64.7	27.1	1.2	5.8	226	2
104.	Turina	60	12.9	21.9	0.8	5.5	133	1
105.	Seninti	47	55.3	26.4	1.4	5.7	239	2
106.	Retno	53	56.1	28.2	1.2	5.8	205	2
107.	Vida	48	51.7	29.3	1.1	5.7	237	2
108.	Rumiyati	54	19.1	17.5	0.8	3.8	188	1
109.	Elmi	39	29.3	15.5	0.8	3.8	154	1
110.	Yelmi	52	23.8	13.9	1	5.2	113	1
111.	Endang	53	63.9	34.7	1.5	6.8	299	3
112.	Rida	45	62.1	38.6	1.5	7.8	271	3
113.	Imroatul	67	55.3	28.9	1.2	6	246	2
114.	Ririn	30	51.4	26.4	1.3	5.8	202	2

No.	Nama	Umur	Urea	BUN	Kreatinin	Uric Acid	Trigliserida	Tingkat Resiko
115.	Novi	28	57.1	23.9	1.4	5.9	249	2
116.	Lusiana	52	65.4	36.2	1.6	6.1	269	3
117.	Fina	42	55.3	27.8	1.3	5.8	203	2
118.	Palupi	52	12.8	18.3	1	5	106	1
119.	Latifa	56	58.9	29.3	1.2	5.7	243	2
120.	Susmiati	50	39.9	13.9	1	3.8	200	1
121.	Najwa	37	63.8	39.3	3.6	11.3	287	3
122.	Badriyah	48	50.4	25.3	1.3	6	248	2
123.	Dyah	51	16.8	17.2	1	2.7	106	1
124.	Juwariyah	43	63.4	34.9	1.6	6.9	278	3
125.	Raikhatul	56	64.7	35.4	1.5	8.6	264	3
126.	Uli	54	58.9	27.2	1.3	6	217	2
127.	Veni	58	51.2	28.2	1.1	6	205	2
128.	Ida	55	58.3	29.2	1.2	5.7	239	2
129.	Mujiana	57	27.4	16.5	0.7	5.4	165	1
130.	Tanti	27	15.4	15.9	0.7	4.5	167	1
131.	Mutawassaula	38	20.5	13.9	0.7	4.5	167	1
132.	Azizah	56	51.2	28.1	1.1	5.8	231	2
133.	Sunarti	56	64.8	37.2	1.5	6.3	271	3
134.	Yunita	59	13.2	19.9	1	5	135	1
135.	Indri	56	68.2	34.9	1.6	7	326	3
136.	Sarti	73	64.1	37.6	1.5	7.2	266	3
137.	Supayah	43	55.8	24.9	1.3	5.9	213	2
138.	Heni Setyaningsih	57	57.8	23.2	1.4	5.8	221	2
139.	Yayuk	43	64.8	39.7	1.5	6.4	289	3
140.	Tri Wardhani	30	52.6	27.2	1.4	6	205	2
141.	Tyas	46	57.5	28.8	1.2	6	209	2
142.	Siti Widiastuti	56	20.6	12.8	0.7	3.2	156	1
143.	Luluk	35	24.9	11.9	0.8	3.8	167	1
144.	Maia	53	54.2	27.8	1.1	5.7	243	2
145.	Juana	63	58.9	24.3	1.3	6	206	2
146.	Jihan	38	55.6	26.2	1.2	5.8	201	2
147.	Miranti	32	64.2	37.8	1.6	6.7	258	3
148.	Indah	52	26.6	14.8	0.8	5.3	198	1
149.	Maimunah	59	66.4	33.7	1.5	6.3	258	3
150.	Dewi Ratna	45	55.4	25.3	1.3	6	228	2

Jumlah Data : 150

Keterangan Tingkat Resiko :

1 = Normal

2 = Sedang

3 = Tinggi

Urea :

Normal : 10-49 mg/dl

Rendah : 50-60 mg/dl

Tinggi : lebih dari 61 mg/dl

BUN :

Normal : 5-22 mg/dl

Rendah : 23-32 mg/dl

Tinggi : lebih dari 33 mg/dl

Kreatinin :

Normal Perempuan : 0.6-1.0
mg/dl

Rendah Perempuan : 1.1 - 1.4
mg/dl

Tinggi laki-laki : Lebih dari 1.4
mg/dl

Uric Acid :

Normal Perempuan : 2.4-5.6
mg/dl

Rendah Perempuan : 5.7-6.0
mg/dl

Tinggi Perempuan : lebih dari 6.0
mg/dl

Trigliserida :

Normal : < 200 mg/dl

Rendah : 201-250 mg/dl

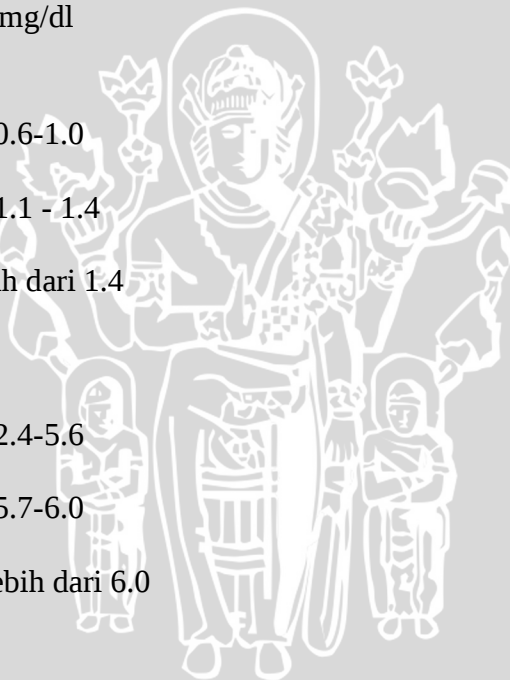
Tinggi : > 250 mg/dl

Umur :

Muda : < 35 tahun

Parobaya : 35 - 55 tahun

Tua : > 55 tahun



LAMPIRAN C

1. *SourceCode* Proses normalisasi Nilai Setiap Atribut

```

1.      public class Fuzzy {
2.      public Fuzzy() {
3.      }
4.      public static void main(String args[]) {
5.      Fuzzy f = new Fuzzy();
6.      }
7.      double MF_UMUR(double x) {
8.
9.      double ret = 0;
10.     ///System.out.println("MF_UMUR = x : "+x);
11.     if (x < 35) {
12.         if (x > 35) {
13.             ret = 0;
14.         } else if (x >= 0 && x < 35) {
15.             ret = (35 - x) / 35;
16.         } else {
17.             ret = 1;
18.         }
19.         ///System.out.print("\t->MUDA\t");
20.     } else if (x >= 35 && x <= 55) {
21.         if (x < 35 || x>55) {
22.             ret = 0;
23.         } else if (x >= 35 && x <= 45) {
24.             ret = (x-35) / (45-35);
25.         } else if (x > 45 && x <= 55) {
26.             ret = (55-x) / (55-45);
27.         } else {
28.             ret = 1;
29.         }
30.         ///System.out.print("\t-
31. >PAROBAYA\t");
32.     } else if (x > 55) {
33.         if (x < 55) {
34.             ret = 0;
35.         } else if (x > 55 && x <= 60) {
36.             ret = (x-55) / (60-55);
37.         } else {
38.             ret = 1;
39.         }
40.         ///System.out.print("\t->TUA\t");
41.     }
42.     ///System.out.println();
43.     return ret;
44.
45. }
46.
47. double MF_UREA(double x) {
48.
49. double ret = 0;
50. ///System.out.println("MF_UREA x : "+x);
51. if (x>=10 && x <=49) {
52.     if (x > 49) {

```

```

53.         ret = 0;
54.     } else if (x >= 10 && x <= 49) {
55.         ret = (49 - x) / 49;
56.     } else {
57.         ret = 1;
58.     }
59.     ////System.out.print("\t->NORMAL\t");
60. }else if (x >= 50 && x <= 60) {
61.     if (x < 50 || x>60) {
62.         ret = 0;
63.     } else if (x >= 50 && x < 55) {
64.         ret = (x-40 ) / (55-50);
65.     } else if (x >= 55 && x <= 60) {
66.         ret = (60-x ) / (60-55);
67.     } else {
68.         ret = 1;
69.     }
70.     ////System.out.print("\t->RENDAH\t");
71. }else if (x > 61) {
72.     if (x < 61) {
73.         ret = 0;
74.     } else if (x >= 61 && x <= 71) {
75.         ret = (x-61) / (71-61);
76.     } else {
77.         ret = 1;
78.     }
79.     ////System.out.print("\t->TINGGI\t");
80. }
81. ////System.out.println();
82. return ret;
83.
84. }
85.
86. double MF_BUN(double x) {
87.
88.     double ret = 0;
89.     ////System.out.println("MF_BUN x : "+x);
90.     if (x>5 && x <=22) {
91.         if (x > 22) {
92.             ret = 0;
93.         } else if (x >= 5 && x <= 22) {
94.             ret = (22 - x) / 22;
95.         } else {
96.             ret = 1;
97.         }
98.         ////System.out.print("\t->NORMAL\t");
99.     }else if (x >= 23 && x <= 32) {
100.         if (x < 23 || x>32) {
101.             ret = 0;
102.         } else if (x >= 23 && x < 28) {
103.             ret = (x-23 ) / (28-23);
104.         } else if (x >= 28 && x <= 32) {
105.             ret = (32-x ) / (32-28);
106.         } else {
107.             ret = 1;
108.         }
109.         //System.out.print("\t->RENDAH\t");

```

```
110.         }else if (x >= 33) {
111.             if (x < 33) {
112.                 ret = 0;
113.             } else if (x >= 33 && x <= 35) {
114.                 ret = (x-33) / (35-33);
115.             } else {
116.                 ret = 1;
117.             }
118.             //System.out.print("\t->TINGGI\t");
119.         }
120.         //System.out.println();
121.         return ret;
122.
123.     }
124.
125.     double MF_KREA_L(double x) {
126.
127.         double ret = 0;
128.         //System.out.println("MF_KREA_L x : "+x);
129.         if (x>0.7 && x <=1.2) {
130.             if (x > 1.2) {
131.                 ret = 0;
132.             } else if (x >= 0.7 && x <= 1.2) {
133.                 ret = (1.2 - x) / 1.2;
134.             } else {
135.                 ret = 1;
136.             }
137.             //System.out.print("\t->NORMAL\t");
138.         }else if (x >= 1.3 && x <= 1.6) {
139.             if (x < 1.3 || x>1.6) {
140.                 ret = 0;
141.             } else if (x >= 1.3 && x < 1.4) {
142.                 ret = (x-1.3) / (1.4-1.3);
143.             } else if (x >= 1.4 && x <= 1.6) {
144.                 ret = (1.6-x) / (1.6-1.4);
145.             } else {
146.                 ret = 1;
147.             }
148.             //System.out.print("\t->RENDAH\t");
149.         }else if (x >= 1.7) {
150.             if (x < 1.7) {
151.                 ret = 0;
152.             } else if (x >= 1.7 && x <= 2) {
153.                 ret = (x-1.7) / (2-1.7);
154.             } else {
155.                 ret = 1;
156.             }
157.             //System.out.print("\t->TINGGI\t");
158.         }
159.         //System.out.println();
160.         return ret;
161.
162.     }
163.     double MF_KREA_P(double x) {
164.
165.         double ret = 0;
166.         //System.out.println("MF_KREA_P x : "+x);
```

```
167.         if (x>0.6 && x <=1) {
168.             if (x > 1) {
169.                 ret = 0;
170.             } else if (x >= 0.6 && x <= 1) {
171.                 ret = (1 - x) / 1;
172.             } else {
173.                 ret = 1;
174.             }
175.             //System.out.print("\t->NORMAL\t");
176.         } else if (x >= 1.1 && x <= 1.4) {
177.             if (x < 1.1 || x>1.4) {
178.                 ret = 0;
179.             } else if (x >= 1.1 && x < 1.2) {
180.                 ret = (x-1.1) / (1.2-1.1);
181.             } else if (x >= 1.2 && x <= 1.4) {
182.                 ret = (1.4-x) / (1.4-1.2);
183.             } else {
184.                 ret = 1;
185.             }
186.             //System.out.print("\t->RENDAH\t");
187.         } else if (x >= 1.4) {
188.             if (x < 1.4) {
189.                 ret = 0;
190.             } else if (x >= 1.4 && x <= 2) {
191.                 ret = (x-1.4) / (2-1.4);
192.             } else {
193.                 ret = 1;
194.             }
195.             //System.out.print("\t->TINGGI\t");
196.         }
197.         //System.out.println();
198.         return ret;
199.     }
200.
201.     double MF_ACID_L(double x) {
202.         double ret = 0;
203.         //System.out.println("MF_ACID_L x : "+x);
204.         if (x>3.5 && x <=6.9) {
205.             if (x > 6.9) {
206.                 ret = 0;
207.             } else if (x >= 3.5 && x <= 6.9) {
208.                 ret = (6.9 - x) / 6.9;
209.             } else {
210.                 ret = 1;
211.             }
212.             //System.out.print("\t->NORMAL\t");
213.         } else if (x >= 7 && x <= 7.5) {
214.             if (x < 7 || x>7.5) {
215.                 ret = 0;
216.             } else if (x >= 7 && x < 7.2) {
217.                 ret = (x-7) / (7.2-7);
218.             } else if (x >= 7.2 && x <= 7.5) {
219.                 ret = (7.5-x) / (7.5-7.2);
220.             } else {
221.                 ret = 1;
222.             }
223.         }
```

```
124.         }
125.         //System.out.print("\t->RENDAH\t");
126.     }else if (x >= 7.5) {
127.         if (x < 7.5) {
128.             ret = 0;
129.         } else if (x >= 7.5 && x <= 8) {
130.             ret = (x-7.5) / (8-7.5);
131.         } else {
132.             ret = 1;
133.         }
134.         //System.out.print("\t->TINGGI\t");
135.     }
136.     //System.out.println();
137.     return ret;
138.
139. }
140.
141. double MF_ACID_P(double x) {
142.
143.     double ret = 0;
144.     //System.out.println("MF_ACID_P x : "+x);
145.     if (x>2.4 && x <=5.6) {
146.         if (x > 5.6) {
147.             ret = 0;
148.         } else if (x >= 2.4 && x <= 5.6) {
149.             ret = (5.6 - x) / 5.6;
150.         } else {
151.             ret = 1;
152.         }
153.         //System.out.print("\t->NORMAL\t");
154.     }else if (x >= 5.7 && x <= 6) {
155.         if (x < 5.7 || x>6) {
156.             ret = 0;
157.         } else if (x >= 5.7 && x <= 5.8) {
158.             ret = (x-5.7) / (5.8-5.7);
159.         } else if (x >= 5.8 && x <= 6) {
160.             ret = (6-x) / (6-5.8);
161.         } else {
162.             ret = 1;
163.         }
164.         //System.out.print("\t->RENDAH\t");
165.     }else if (x >= 6) {
166.         if (x < 6) {
167.             ret = 0;
168.         } else if (x >= 6 && x <= 7) {
169.             ret = (x-6) / (7-6);
170.         } else {
171.             ret = 1;
172.         }
173.         //System.out.print("\t->TINGGI\t");
174.     }
175.     //System.out.println();
176.     return ret;
177.
178. }
179. double MF_TRIG(double x) {
180.
```

```
181.         double ret = 0;
182.         //System.out.println("MF_TRIG x : "+x);
183.         if (x <= 200) {
184.             if (x > 200) {
185.                 ret = 0;
186.             } else if (x >= 0 && x <= 200) {
187.                 ret = (200 - x) / 200;
188.             } else {
189.                 ret = 1;
190.             }
191.             //System.out.print("\t->NORMAL\t");
192.         } else if (x >= 201 && x <= 250) {
193.             if (x < 201 || x > 250) {
194.                 ret = 0;
195.             } else if (x >= 201 && x < 225) {
196.                 ret = (x-201) / (225-201);
197.             } else if (x >= 225 && x <= 250) {
198.                 ret = (250-x) / (250-225);
199.             } else {
200.                 ret = 1;
201.             }
202.             //System.out.print("\t->RENDAH\t");
203.         } else if (x > 250) {
204.             if (x < 250) {
205.                 ret = 0;
206.             } else if (x > 250 && x <= 300) {
207.                 ret = (x-250) / (300-250);
208.             } else {
209.                 ret = 1;
210.             }
211.             //System.out.print("\t->TINGG\t");
212.         }
213.         //System.out.println();
214.         return ret;
215.     }
216.
217. }
```

2.Sourcecode Proses Klasifikasi dengan Algoritma Fuzzy K-Nearest Neighbor

No.	Source Code
1.	import java.util.ArrayList;
2.	import javax.swing.JOptionPane;
3.	import javax.swing.table.DefaultTableModel;
4.	
5.	public class FrmMain extends javax.swing.JFrame {
6.	MyQConn conn;
7.	
8.	String sQry = "SELECT `id_diagnosa`, `nama`, `umur`,
9.	`urea`, `bun`, `krea`, `uric`, `trig`, `tr`, `jk` FROM
10.	`tb_diagnosa` " ;
11.	String[] sHeader = { "ID", "Nama", "Umur", "Urea", "BUN",
12.	"Kreatin", "Uric Acid", "Trigliserida", "Tingkat Resiko"
13.	};
14.	String[] sColumn = { "id_diagnosa", "nama", "umur",
15.	"urea", "bun", "krea", "uric", "trig", "tr", "jk" };
16.	String[][] sKey =null;
17.	public FrmMain() {
18.	conn=new MyQConn();
19.	initComponents();
20.	}
21.	
22.	@Override
23.	public void show() {
24.	showData();
25.	super.show(); //To change body of generated
26.	methods, choose Tools Templates.
27.	}
28.	@SuppressWarnings("unchecked")
29.	double[][] valTrain;
30.	
31.	private void
32.	btnViewActionPerformed(java.awt.event.ActionEvent
33.	evt) {
34.	showData();
35.	}
36.	private void
37.	tbDataMouseClicked(java.awt.event.MouseEvent evt)
38.	{
39.	int ij= tbData.getSelectedRow();
40.	edUmur.setText(sKey[ij][2]);
41.	edUrea.setText(sKey[ij][3]);
42.	edBun.setText(sKey[ij][3]);
43.	edKrea.setText(sKey[ij][5]);
44.	edUric.setText(sKey[ij][6]);
45.	edTri.setText(sKey[ij][7]);
46.	edUmur.setText(sKey[ij][2]);
47.	
48.	}

```

49.         private void
50.         btnProsesActionPerformed(java.awt.event.ActionEvent
51.         evt) {
52.
53.             int NData =
54.             Integer.parseInt(edK2.getText());
55.             if (NData > sKey.length) {
56.                 NData = sKey.length;
57.             }
58.             String[][] sKey_ = new
59.             String[NData][sKey[0].length+1] ;
60.
61.             for (int i = 0; i < NData; i++) {
62.                 sKey_[i] = sKey[i];
63.             }
64.             for (int ij = 0; ij < sKey_.length; ij++) {
65.                 valTrain = new
66.                 double[sKey_.length][7];
67.                 double[] targetTrain = new
68.                 double[sKey_.length] ;
69.                 Fuzzy fuzz = new Fuzzy();
70.                 for (int i = 0; i < sKey_.length; i++)
71.                 {
72.                     double vUmur =
73.                     fuzz.MF_UMUR(Double.parseDouble(sKey_[i][2]));
74.                     double vUrea =
75.                     fuzz.MF_UREA(Double.parseDouble(sKey_[i][3]));
76.                     double vBun =
77.                     fuzz.MF_BUN(Double.parseDouble(sKey_[i][4]));
78.                     double vKrea = 0;
79.                     double vUric = 0;
80.
81.                     if (sKey_[i][9].equals("L")) {
82.                         vKrea =
83.                         fuzz.MF_KREA_L(Double.parseDouble(sKey_[i][5]));
84.                         vUric
85.                         =fuzz.MF_ACID_L(Double.parseDouble(sKey_[i][6]));
86.                     }else {
87.                         vKrea =
88.                         fuzz.MF_KREA_P(Double.parseDouble(sKey_[i][5]));
89.                         vUric
90.                         =fuzz.MF_ACID_P(Double.parseDouble(sKey_[i][6]));
91.
92.                     }
93.
94.                     double vTri =
95.                     fuzz.MF_TRIG(Double.parseDouble(sKey_[i][7]));
96.                     double vTarget =
97.                     (Double.parseDouble(sKey_[i][8]));
98.                     valTrain[i][0] = vUmur;
99.                     valTrain[i][1] = vUrea;
100.                    valTrain[i][2] = vBun;

```

```

101.         valTrain[i][3] = vKrea;
102.         valTrain[i][4] = vUric;
103.         valTrain[i][5] = vTri;
104.         valTrain[i][6] = vTarget;
105.         targetTrain[i] = vTarget;
106.
107.     }
108.     double vUmur =
109.         fuzz.MF_UMUR(Double.parseDouble(sKey_[ij][2]));
110.         double vUrea =
111.         fuzz.MF_UREA(Double.parseDouble(sKey_[ij][3]));
112.         double vBun =
113.         fuzz.MF_BUN(Double.parseDouble(sKey_[ij][4]));
114.         double vKrea = 0;
115.         double vUric = 0;
116.
117.         if (sKey_[ij][9].equals("L")) {
118.             vKrea =
119.             fuzz.MF_KREA_L(Double.parseDouble(sKey_[ij][5]));
120.             vUric
121.             =fuzz.MF_ACID_L(Double.parseDouble(sKey_[ij][6]));
122.         }else {
123.             vKrea =
124.             fuzz.MF_KREA_P(Double.parseDouble(sKey_[ij][5]));
125.             vUric
126.             =fuzz.MF_ACID_P(Double.parseDouble(sKey_[ij][6]));
127.         }
128.         System.out.println("Data ke- " + ij);
129.         double vTri =
130.
131.         fuzz.MF_TRIG(Double.parseDouble(edTri.getText().tr
132.         im()));
133.         double [] valDist = new
134.         double[valTrain.length];
135.         for (int i = 0; i < valTrain.length;
136.         i++) {
137.
138.         double tUmur = Math.pow(vUmur - valTrain[i][0],2);
139.         double tUrea = Math.pow(vUrea - valTrain[i][1],2);
140.         double tBun = Math.pow(vBun - valTrain[i][2],2);
141.         double tKrea = Math.pow(vKrea - valTrain[i][3],2);
142.         double tUric = Math.pow(vUric - valTrain[i][4],2);
143.         double tTri= Math.pow(vTri - valTrain[i][5],2);
144.         double d = Math.sqrt(
145.         Math.sqrt(tUmur+tUrea+tBun+tKrea+tUric+tTri));
146.         valDist[i] = d;
147.         System.out.print(d+"\t");
148.
149.     }
150.
151.

```

Kode program untuk mengurutkan hasil *euclidean distance*

```

152.
153.
154.      System.out.println(" " );
155.
156.      for(int i=0;i<valDist.length-1;i++)
157.      {
158.          for(int j=1;j<valDist.length;j++)
159.          {
160.              double a= valDist[j] ;
161.              double b= valDist[j-1];
162.              if(a>b)
163.              {
164.                  double aTemp=valDist[j];
165.                  valDist[j]=valDist[j-1];
166.                  valDist[j-1]=aTemp;
167.                  aTemp=targetTrain[j];
168.
169.                  targetTrain[j]=targetTrain[j-1];
170.                  targetTrain[j-1]=aTemp;
171.              }
172.          }
173.      }
174.
175.

```

Mengambil hasil *euclidean distance* sesuai dengan nilai k yang dimasukkan dan menentukan tingkat resiko

```

176.
177.
178.
179.      int K = Integer.parseInt(edK1.getText().trim());
180.      ArrayList<Double> targetK = new ArrayList<>();
181.      ArrayList<Integer> targetCounq = new ArrayList<>();
182.
183.      for (int i = 0; i < K; i++) {
184.          double d = targetTrain[i];
185.
186.
187.      if (targetK.contains(d)){
188.          int idx = targetK.indexOf(d);
189.
190.          targetCounq.set(idx, targetCounq.get(idx)+1) ;
191.      }else{
192.          targetK.add(d) ;
193.          targetCounq.add(1) ;
194.      }
195.
196.      }

```

menghitung niali *F-measure* dan nilai kinerja sistem

```

194.
195.
196.         int maxCount = 0;
197.         int idSol =0;
198.         int countAll=0;
199.         for (int i = 0; i < targetCounq.size(); i++) {
200.             Integer integer = targetCounq.get(i);
201.             if (integer > maxCount){
202.                 maxCount = integer;
203.                 idSol = i;
204.             }
205.             countAll+=integer;
206.             System.out.println(targetK.get(i) + " : "+integer);
207.         }
208.         double TP = targetCounq.get(idSol);
209.         double TN = targetCounq.get(idSol) +
110.         targetCounq.get(idSol) ;
111.
112.         double FN = countAll - targetCounq.get(idSol);
113.         double FP = 0;
114.
115.         double recall = TP/(TP+FN);
116.         double prec = TP/(TP+FP);
117.         double fmes = (2*recall*prec)/ (prec+recall);
118.
119.         double kSis = ((TP+TN) / (TP+TN+FP+FN)) * 100;
120.
121.         edTP1.setText(String.valueOf(TP));
122.         edTN1.setText(String.valueOf(TN));
123.         edFP1.setText(String.valueOf(FP));
124.         edFN1.setText(String.valueOf(FN));
125.
126.         edR1.setText(String.valueOf(recall));
127.         edP1.setText(String.valueOf(prec));
128.
129.         edFM1.setText(String.valueOf(fmes));
130.
131.         edKinerjal.setText(String.valueOf(kSis));
132.         String[]Target = new String [] {
133.             "", "Tinggi", "Rendah", "Normal";
134.             System.out.println( Target[ (int)Math.round(
135.                 targetK.get(idSol)) ] );
136.                 sKey_[ij][sKey_[0].length-1] = ""+(
137.                 Math.round( targetK.get(idSol)) ) ;
138.             }
139.             String[] sHeaders = {"ID", "Nama",
140.                 "Umur", "Urea", "BUN", "Kreatin", "Uric
141.                 Acid", "Trigliserida", "Tingkat Resiko", "Hasik" };
142.             tbData1.setModel(new
143.                 DefaultTableModel(sKey_, sHeaders));
144.
145.
146.         }

```

```

147.         private void
148.             edFM1ActionPerformed(java.awt.event.ActionEvent
149. evt) {
150.
151.         }
152.
153.         private void
154.             edKinerjaActionPerformed(java.awt.event.ActionEve
155. nt evt) {
156.
157.         }
158.
159.         private void
160.             edK1ActionPerformed(java.awt.event.ActionEvent evt) {
161.
162.         }
163.         private void
164.             edK2ActionPerformed(java.awt.event.ActionEvent evt) {
165.
166.         }
167.
168.         private void
169.             tbData1MouseClicked(java.awt.event.MouseEvent evt)
170. {
171.
172.         }
173.
174.         private void
175.             edKActionPerformed(java.awt.event.ActionEvent evt) {
176.
177.         }
178.         private void
179.             edKinerjaActionPerformed(java.awt.event.ActionEvent
180. evt) {
181.
182.         }
183.         private void
184.             edFMActionPerformed(java.awt.event.ActionEvent evt) {
185.
186.         }
187.         private void
188.             edTingkatResikoActionPerformed(java.awt.event.Acti
189. onEvent evt) {
190.
191.         }
192.
193.         private void
194.             btnUlangiActionPerformed(java.awt.event.ActionEvent
195. evt) {

```

```

196.         edUmur.setText("");
197.         edUrea.setText("");
198.         edBun.setText("");
199.         edKrea.setText("");
200.         edUric.setText("");
201.         edTri.setText("");
202.         edTingkatResiko.setText("");
203.         edFM.setText("");
204.         edFN.setText("");
205.         edFP.setText("");
206.         edKinerja.setText("");
207.         edTN.setText("");
208.         edTP.setText("");
209.         edR.setText("");
210.         edP.setText("");
211.     }
212.
213.     private void
214.     btnProsesActionPerformed(java.awt.event.ActionEvent
215.     evt) {
216.         // "id_diagnosa", "nama", "umur", "urea",
217.         "bun", "krea", "uric", "trig", "tr", "jk"
218.
219.         if (edUmur.getText().isEmpty() ||
220.             edUrea.getText().isEmpty() ||
221.             edBun.getText().isEmpty() ||
222.             edKrea.getText().isEmpty() ||
223.             edUric.getText().isEmpty() ||
224.             edTri.getText().isEmpty()) {
225.
226.             JOptionPane.showMessageDialog(rootPane, "Isikan
227.             Semua Data...");
228.             return;
229.         }
230.
231.         valTrain = new double[sKey.length][7];
232.         double[] targetTrain = new double[sKey.length];
233.         Fuzzy fuzz = new Fuzzy();
234.         for (int i = 0; i < sKey.length; i++) {
235.             double vUmur =
236.             fuzz.MF_UMUR(Double.parseDouble(sKey[i][2]));
237.             double vUrea =
238.             fuzz.MF_UREA(Double.parseDouble(sKey[i][3]));
239.             double vBun =
240.             fuzz.MF_BUN(Double.parseDouble(sKey[i][4]));
241.             double vKrea = 0;
242.             double vUric = 0;
243.
244.
245.             if (sKey[i][9].equals("L")) {
246.                 vKrea =
247.                 fuzz.MF_KREA_L(Double.parseDouble(sKey[i][5]));
248.                 vUric
249.                 =fuzz.MF_ACID_L(Double.parseDouble(sKey[i][6]));

```

```

250.         }else {
251.             vKrea =
252.             fuzz.MF_KREA_P(Double.parseDouble(sKey[i][5]));
253.             vUric
254.             =fuzz.MF_ACID_P(Double.parseDouble(sKey[i][6]));
255.
256.         }
257.         double vTri =
258.         fuzz.MF_TRIG(Double.parseDouble(sKey[i][7]));
259.         double vTarget =
260.         (Double.parseDouble(sKey[i][8]));
261.         valTrain[i][0] = vUmur;
262.         valTrain[i][1] = vUrea;
263.         valTrain[i][2] = vBun;
264.         valTrain[i][3] = vKrea;
265.         valTrain[i][4] = vUric;
266.         valTrain[i][5] = vTri;
267.         valTrain[i][6] = vTarget;
268.         targetTrain[i] = vTarget;
269.
270.     }
271.     String Jk =String.valueOf(
272.     cbJk.getSelectedItem().toString().charAt(0));
273.
274.     double vUmur =
275.
276.     fuzz.MF_UMUR(Double.parseDouble(edUmur.getText().t
277.     rim()));
278.     double vUrea =
279.
280.     fuzz.MF_UREA(Double.parseDouble(edUrea.getText().t
281.     rim()));
282.     double vBun =
283.
284.     fuzz.MF_BUN(Double.parseDouble(edBun.getText().tri
285.     m()));
286.     double vKrea = 0;
287.     double vUric = 0;
288.     if (Jk.equals("L")) {
289.         vKrea =
290.
291.         fuzz.MF_KREA_L(Double.parseDouble(edKrea.getText()
292.         .trim()));
293.         vUric
294.         =fuzz.MF_ACID_L(Double.parseDouble(edUric.getText(
295.         ).trim()));
296.     }else {
297.         vKrea =
298.
299.         fuzz.MF_KREA_P(Double.parseDouble(edKrea.getText()
300.         .trim()));
301.         vUric

```

```

302. =fuzz.MF_ACID_P(Double.parseDouble(edUric.getText().trim
303. ());
304.
305.     }
306.     double vTri =
307.
308.     fuzz.MF_TRIG(Double.parseDouble(edTri.getText().tr
309. im()));
310.         double []      valDist      =      new
311. double[valTrain.length];
312.         for (int i = 0; i < valTrain.length; i++)
313.     {
314.
315.         double tUmur = Math.pow(vUmur - valTrain[i][0],2);
316.         double tUrea = Math.pow(vUrea - valTrain[i][1],2);
317.         double tBun = Math.pow(vBun - valTrain[i][2],2);
318.         double tKrea = Math.pow(vKrea - valTrain[i][3],2);
319.         double tUric = Math.pow(vUric - valTrain[i][4],2);
320.         double tTri= Math.pow(vTri - valTrain[i][5],2);
321.         double d =
322.         Math.sqrt(
323.         Math.sqrt(tUmur+tUrea+tBun+tKrea+tUric+tTri));
324.             valDist[i] = d;
325.         }
326.
327.         for(int i=0;i<valDist.length-1;i++)
328.         {
329.             for(int j=1;j<valDist.length;j++)
330.             {
331.                 double a= valDist[j] ;
332.                 double b= valDist[j-1];
333.                 if(a>b)
334.                 {
335.                     double aTemp=valDist[j];
336.                     valDist[j]=valDist[j-1];
337.                     valDist[j-1]=aTemp;
338.
339.                     aTemp=targetTrain[j];
340.                     targetTrain[j]=targetTrain[j-1];
341.                     targetTrain[j-1]=aTemp;
342.                 }
343.             }
344.
345.         }
346.
347.     }
348.     int K =
349.     Integer.parseInt(edK.getText().trim());
350.     ArrayList<Double>      targetK      =      new
351. ArrayList<>();
352.     ArrayList<Integer>      targetCounq      =      new
353. ArrayList<>();
354.

```

```

355.         for (int i = 0; i < K; i++) {
356.             double d = targetTrain[i];
357.
358.             if (targetK.contains(d)) {
359.                 int idx = targetK.indexOf(d);
360.                 targetCounq.set(idx,
361. targetCounq.get(idx)+1) ;
362.             }else{
363.                 targetK.add(d);
364.                 targetCounq.add(1);
365.             }
366.         }
367.         int maxCount = 0;
368.         int idSol = 0;
369.         int countAll=0;
370.         for (int i = 0; i < targetCounq.size();
371. i++) {
372.             Integer integer = targetCounq.get(i);
373.             if (integer > maxCount){
374.                 maxCount = integer;
375.                 idSol = i;
376.             }
377.             countAll+=integer;
378.             System.out.println(targetK.get(i) + "
379. : "+integer);
380.         }
381.         double TP = targetCounq.get(idSol);
382.         double TN = targetCounq.get(idSol) +
383. targetCounq.get(idSol) ;
384.
385.         double FN = countAll -
386. targetCounq.get(idSol);
387.         double FP = 0;
388.
389.         double recall = TP/(TP+FN);
390.         double prec = TP/(TP+FP);
391.         double fmes = (2*recall*prec)/
392. (prec+recall);
393.
394.         double kSis = ((TP+TN) / (TP+TN+FP+FN)) *
395. 100;
396.
397.         edTP.setText(String.valueOf(TP));
398.         edTN.setText(String.valueOf(TN));
399.         edFP.setText(String.valueOf(FP));
400.         edFN.setText(String.valueOf(FN));
401.
402.         edR.setText(String.valueOf(recall));
403.         edP.setText(String.valueOf(prec));
404.
405.         edFM.setText(String.valueOf(fmes));
406.         edKinerja.setText(String.valueOf(kSis));

```

```

407. String[] Target = new String [] {
408.     "", "Tinggi", "Rendah", "Normal" };
409.
410. edTingkatResiko.setText( Target[ (int)Math.round(
411.     targetK.get(idSol)) ] );
412.
413. }
414.
415. private void
416.     edTriActionPerformed(java.awt.event.ActionEvent
417.     evt) {
418.
419.     }
420.
421. private void
422.     edUricActionPerformed(java.awt.event.ActionEvent
423.     evt) {
424.
425.     }
426.
427. private void
428.     edKreaActionPerformed(java.awt.event.ActionEvent
429.     evt) {
430.
431.     }
432.
433. private void
434.     edBunActionPerformed(java.awt.event.ActionEvent
435.     evt) {
436.
437.     }
438.
439. private void
440.     edUreaActionPerformed(java.awt.event.ActionEvent
441.     evt) {
442.
443.     }
444.
445. private void
446.     edUmurActionPerformed(java.awt.event.ActionEvent
447.     evt) {
448.
449.     }
450.
451. public static void main(String args[]) {
452.     /* Set the Nimbus look and feel */
453.     //

```

for

```

459.      (javax.swing.UIManager.LookAndFeelInfo info :
460.      javax.swing.UIManager.getInstalledLookAndFeels())
461.      {
462.          if
463.      ("Nimbus".equals(info.getName())) {
464.
465.      javax.swing.UIManager.setLookAndFeel(info.getClass
466.      Name());
467.          break;
468.      }
469.      }
470.      } catch (ClassNotFoundException ex) {
471.
472.      java.util.logging.Logger.getLogger(FrmMain.class.g
473.      etName()).log(java.util.logging.Level.SEVERE,
474.      null, ex);
475.      } catch (InstantiationException ex) {
476.
477.      java.util.logging.Logger.getLogger(FrmMain.class.g
478.      etName()).log(java.util.logging.Level.SEVERE,
479.      null, ex);
480.      } catch (IllegalAccessException ex) {
481.
482.      java.util.logging.Logger.getLogger(FrmMain.class.g
483.      etName()).log(java.util.logging.Level.SEVERE,
484.      null, ex);
485.      } catch
486.      (javax.swing.UnsupportedLookAndFeelException ex) {
487.
488.      java.util.logging.Logger.getLogger(FrmMain.class.g
489.      etName()).log(java.util.logging.Level.SEVERE,
490.      null, ex);
491.      }
492.      //</editor-fold>
493.
494.      /* Create and display the form */
495.      java.awt.EventQueue.invokeLater(new
496.      Runnable() {
497.          public void run() {
498.              new FrmMain().setVisible(true);
499.          }
500.      });
501.      }
502.      // Variables declaration - do not modify
503.      private javax.swing.JButton btnProses;
504.      private javax.swing.JButton btnProses1;
505.      private javax.swing.JButton btnUlangi;
506.      private javax.swing.JButton btnView;
507.      private javax.swing.ButtonGroup buttonGroup1;
508.      private javax.swing.JComboBox cbJk;
509.      private javax.swing.JFormattedTextField edBun;
510.
511.      private javax.swing.JFormattedTextField edFM;

```

```

511.         private javax.swing.JFormattedTextField edFM1;
512.         private javax.swing.JFormattedTextField edFN;
513.         private javax.swing.JFormattedTextField edFN1;
514.         private javax.swing.JFormattedTextField edFP;
515.         private javax.swing.JFormattedTextField edFP1;
516.         private javax.swing.JFormattedTextField edK;
517.         private javax.swing.JFormattedTextField edK1;
518.         private javax.swing.JFormattedTextField edK2;
519.     private javax.swing.JFormattedTextField edKinerja;
520.     private javax.swing.JFormattedTextField edKinerja1;
521.     private javax.swing.JFormattedTextField edKrea;
522.         private javax.swing.JFormattedTextField edP;
523.         private javax.swing.JFormattedTextField edP1;
524.         private javax.swing.JFormattedTextField edR;
525.         private javax.swing.JFormattedTextField edR1;
526.         private javax.swing.JFormattedTextField edTN;
527.         private javax.swing.JFormattedTextField edTN1;
528.         private javax.swing.JFormattedTextField edTP;
529.         private javax.swing.JFormattedTextField edTP1;
530.     private javax.swing.JFormattedTextField
531.     edTingkatResiko;
532.         private javax.swing.JFormattedTextField edTri;
533.         private javax.swing.JFormattedTextField edUmur;
534.         private javax.swing.JFormattedTextField edUrea;
535.         private javax.swing.JFormattedTextField edUric;
536.         private javax.swing.JComboBox jComboBox1;
537.         private javax.swing.JLabel jLabel1;
538.         private javax.swing.JLabel jLabel10;
539.         private javax.swing.JLabel jLabel11;
540.         private javax.swing.JLabel jLabel12;
541.         private javax.swing.JLabel jLabel13;
542.         private javax.swing.JLabel jLabel14;
543.         private javax.swing.JLabel jLabel15;
544.         private javax.swing.JLabel jLabel16;
545.         private javax.swing.JLabel jLabel17;
546.         private javax.swing.JLabel jLabel18;
547.         private javax.swing.JLabel jLabel19;
548.         private javax.swing.JLabel jLabel2;
549.         private javax.swing.JLabel jLabel20;
550.         private javax.swing.JLabel jLabel21;
551.         private javax.swing.JLabel jLabel22;
552.         private javax.swing.JLabel jLabel23;
553.         private javax.swing.JLabel jLabel3;
554.         private javax.swing.JLabel jLabel31;
555.         private javax.swing.JLabel jLabel32;
556.         private javax.swing.JLabel jLabel33;
557.         private javax.swing.JLabel jLabel34;
558.         private javax.swing.JLabel jLabel35;
559.         private javax.swing.JLabel jLabel36;
560.         private javax.swing.JLabel jLabel37;
561.         private javax.swing.JLabel jLabel38;
562.
private javax.swing.JLabel jLabel39;

```

```

563.         private javax.swing.JLabel jLabel4;
564.         private javax.swing.JLabel jLabel40;
565.         private javax.swing.JLabel jLabel41;
566.         private javax.swing.JLabel jLabel5;
567.         private javax.swing.JLabel jLabel6;
568.         private javax.swing.JLabel jLabel7;
569.         private javax.swing.JLabel jLabel8;
570.         private javax.swing.JLabel jLabel9;
571.         private javax.swing.JPanel jPanel1;
572.         private javax.swing.JPanel jPanel2;
573.         private javax.swing.JPanel jPanel3;
574.         private javax.swing.JPanel jPanel4;
575.         private javax.swing.JPanel jPanel6;
576.         private javax.swing.JPanel jPanel7;
577.         private javax.swing.JScrollPane jScrollPane1;
578.         private javax.swing.JScrollPane jScrollPane2;
579.         private javax.swing.JSeparator jSeparator1;
580.         private javax.swing.JSeparator jSeparator4;
581.         private javax.swing.JSplitPane jSplitPanel1;
582.         private javax.swing.JTabbedPane jTabbedPane1;
583.         private javax.swing.JTable tbData;
584.         private javax.swing.JTable tbData1;
585.         private javax.swing.JLabel txtJml;
586.         // End of variables declaration
587.
588.         private void showData() {
589.             conn.showinTable(sQry+ "      where      jk
590. ='" + (cbJk.getSelectedItemAt(0).toString().charAt(0)) +
591. "' order by RAND() " , tbData,sHeader,sColumn);
592.             String[][] sKey_ = conn.getKeyData();
593.             sKey_ = new String[
594. Integer.parseInt(jComboBox1.getSelectedItemAt(0).toSt
595. ring()) ][sKey_[0].length];
596.             int idx=0;
597.
598.             int[] jmlData =new int[] {0,0,0};
599.             for (int i = 0; i < sKey_.length; i++) {
600.                 String strings = sKey_[i][8];
601.                 if (strings.equals("1") && jmlData[0]
602. <
603. Integer.parseInt(jComboBox1.getSelectedItemAt(0).toSt
604. ring())/3) ){
605.                     sKey[idx] = sKey_[i];
606.                     idx++;
607.                     jmlData[0]++;
608.
609.                 }else if (strings.equals("2") &&
610. jmlData[1] <
611. Integer.parseInt(jComboBox1.getSelectedItemAt(0).toSt
612. ring())/3) ){
613.                     sKey[idx] = sKey_[i];
614.

```

idx++;

615.	jmlData[1]++;
616.	
617.	}else if (strings.equals("3") &&
618.	jmlData[2] < (
619.	Integer.parseInt(jComboBox1.getSelectedItem().toString())/3)){
620.	
621.	sKey[idx] = sKey_[i];
622.	idx++;
623.	
624.	jmlData[2]++;
625.	}
626.	if (idx % (
627.	Integer.parseInt(jComboBox1.getSelectedItem().toString())) ==0){
628.	
629.	break;
630.	}
631.	}
632.	
633.	tbData.setModel(new
634.	DefaultTableModel(sKey, sHeader));
635.	
636.	txtJml.setText("Jml. Data :
637.	"+String.valueOf(sKey.length));
638.	}
639.	}

3. Sourcecode Koneksi Database

No.	Source Code
1.	import com.mysql.jdbc.Connection;
2.	import com.mysql.jdbc.ResultSetMetaData;
3.	import com.mysql.jdbc.Statement;
4.	import java.sql.DriverManager;
5.	import java.sql.ResultSet;
6.	import javax.swing.JTable;
7.	import javax.swing.table.DefaultTableModel;
8.	
9.	public class MyQConn {
10.	
11.	Connection con = null;
12.	String url = "jdbc:mysql://localhost:3306/";
13.	String db = "db_fkk_gg";
14.	String driver = "com.mysql.jdbc.Driver";
15.	String user ="root";
16.	String pass ="";
17.	String[][] keyData=null;

```

18.         boolean connect(){
19.             try {
20.                 Class.forName(driver);
21.                 con         =         (Connection)
22.                 DriverManager.getConnection(url+db,user,pass);
23.                 keyData=null;
24.                 return true;
25.             } catch ( Exception ex) {
26.                 return false;
27.             }
28.         }
29.
30.         public Connection getCon() {
31.             if (!connect()) {
32.                 return null;
33.             }
34.             return con;
35.         }
36.
37.         public boolean runQuery(String qry){
38.             try {
39.                 if (!connect()) {
40.                     return false;
41.                 }
42.                 Statement st         =         (Statement)
43.                 con.createStatement();
44.                 int val = st.executeUpdate(qry);
45.                 st.close();
46.                 con.close();
47.                 return val==0 ? false : true;
48.             } catch ( Exception ex) {
49.                 return false;
50.             }
51.         }
52.         //panggil query
53.         public boolean cekQuery(String qry){
54.             try {
55.                 if (!connect()) {
56.                     return false;
57.                 }
58.                 Statement st         =         (Statement)
59.                 con.createStatement();
60.                 ResultSet val         =
61.                 st.executeQuery(qry);
62.                 val.next();
63.                 int ret = val.getRow();
64.                 st.close();
65.                 con.close();
66.
67.                 return ret ==0 ? false : true;
68.
69.

```

```

70.         } catch ( Exception ex) {
71.             return false;
72.         }
73.     }
74.
75.     public String[][] getKeyData() {
76.         return keyData;
77.     }
78.
79.     //ambil data hasil query
80.     public String[]  getData(String qry ){
81.         try {
82.
83.             if (!connect()) {
84.                 return null;
85.             }
86.
87.             Statement      st      = (Statement)
88. con.createStatement();
89.             ResultSet rs;
90.
91.             rs = st.executeQuery(qry);
92.
93.             int baris = 0;
94.
95.             while (rs.next()) {
96.                 baris = rs.getRow();
97.             }
98.             String[]      strData=new
99. String[baris];
100.
101.             int x = 0;
102.             rs.beforeFirst();
103.             while (rs.next()) {
104.                 strData[x] = rs.getString(1);
105.                 x++;
106.             }
107.
108.             st.close();
109.             rs.close();
110.             con.close();
111.             return strData;
112.         } catch ( Exception ex) {
113.             return null;
114.         }
115.
116.     }
117.     //tampil data pada table
118.     public      boolean      showinTable(String
119. qry,JTable      table,String[]      header,String[]
120. columns ){
121.         try {

```

```

122.
123.         if (!connect()) {
124.             return false;
125.         }
126.
127.
128.         Statement st = (Statement)
129.         con.createStatement();
130.         ResultSet rs;
131.         Object[][] dataTable = null;
132.         rs = st.executeQuery(qry);
133.         ResultSetMetaData meta =
134.         (ResultSetMetaData) rs.getMetaData();
135.
136.         int col = meta.getColumnCount();
137.         int baris = 0;
138.
139.         while (rs.next()) {
140.             baris = rs.getRow();
141.         }
142.         keyData = new String[baris][col];
143.         dataTable = new Object[baris][col];
144.         int x = 0;
145.         rs.beforeFirst();
146.         while (rs.next()) {
147.             for (int
148.             i=0;i<columns.length;i++){
149.                 dataTable[x][i] =
150.                 rs.getString(columns[i]);
151.                 keyData[x][i] =
152.                 String.valueOf( dataTable[x][i]);
153.             }
154.
155.             x++;
156.         }
157.         table.setModel(new
158.         DefaultTableModel(dataTable, header));
159.
160.         st.close();
161.         rs.close();
162.         con.close();
163.         return true;
164.     } catch (Exception ex) {
165.         return false;
166.     }
167.
168.     }

```