

PENERAPAN METODE *K-NEAREST NEIGHBOR* DALAM MENDETEKSI KERUSAKAN SSD

Abil Khasani, Haris Yuana, Zunita Wulansari

Teknik Informatika, Universitas Islam Balitar Blitar
Jl. Majapahit No.2- 4, Sananwetan, Sananwetan, Blitar, Indonesia
nyuubbest@gmail.com

ABSTRAK

Masalah yang sering dihadapi pengguna teknologi *Solid State Drive (SSD)* adalah kesulitan untuk mengetahui bahwa *SSD* mereka sedang dalam kondisi rusak atau tidak. Sedangkan pengguna belum melakukan persiapan pengamanan data penting, namun tiba-tiba *SSD* sudah mengalami rusak berat. Untuk pengembalian data penting bisa jadi mengalami kesulitan untuk *back up* data dan mungkin terjadi tidak bisa mengambil data penting yang di dalam *SSD* yang sudah rusak. Karena *SSD* dengan kondisi rusak dalam rentang waktu beberapa hari atau beberapa pemakaian masih bisa digunakan tetapi kalau sudah rusak total maka tidak bisa diperbaiki lagi. Penelitian ini bertujuan mengurangi kerugian pengguna *SSD* dalam pengamanan data penting. Sehingga pengguna *SSD* jauh – jauh hari sudah melakukan persiapan pengamanan data penting dan tidak mengalami kerugian banyak. Penelitian ini menentukan kerusakan *SSD* dengan menerapkan metode *K-NEAREST NEIGHBOR* ke dalam sistem untuk menentukan status rusak atau tidak adanya kerusakan suatu *SSD*. Berdasarkan hasil pengujian nilai akurasi pada penelitian ini sebesar 80% dengan cara menghitung manual dan menggunakan aplikasi *Rapid Miner*.

Kata Kunci: Kerusakan *SSD*, *K-NEAREST NEIGHBOR*, *Rapid Miner*

1. PENDAHULUAN

Data mining adalah tahapan proses untuk memperoleh nilai tambah dari suatu kumpulan data berupa pengetahuan yang belum diketahui secara manual. Data mining dapat digunakan untuk memperoleh informasi dari data yang besar sehingga nantinya mendapatkan informasi yang bisa digunakan dalam mendiagnosa sebuah kerusakan. Salah satu metode dari data mining yang bisa digunakan untuk memprediksi yaitu metode *K-Nearest Neighbor (KNN)*. *KNN* adalah metode penyelesaian masalah yang sederhana dalam menentukan sebuah kerusakan data dari *Solid-State Drive* berdasarkan data kerusakan yang lama dengan melalui pendekatan berdasarkan jarak antara data terdekat. Metode ini dipilih berdasarkan studi literatur yang dilakukan, dimana metode *KNN* menghasilkan nilai keakuratan yang tinggi. Seperti pada penelitian (Syukri Mustafa & Wayan Simpen, 2019) tentang memprediksi pasien terkena penyakit diabetes menghasilkan nilai akurasi sebesar 93,33% dengan *error rate* 6,67%. Penelitian lain dengan bidang berbeda yaitu implemantasi *KNN* untuk klasifikasi bunga dengan ekstraksi fitur warna RGB dengan menghasilkan nilai akurasi sebesar 90%. Dalam mendeteksi sebuah kerusakan menjadi sebuah pengetahuan terdapat beberapa langkah. Salah satu langkah tersebut adalah pengumpulan gejala – gejala kerusakan perangkat keras yaitu *Solid-State Drive* dan data yang pernah terjadi atau bisa disimpulkan dengan kerusakan yang pernah terjadi terhadap *Solid-State Drive*. *Solid-State Drive (SSD)* adalah media penyimpanan yang menggunakan memori *non-volatile (flash)* untuk menyimpan dan mengakses data, tidak sama seperti *HDD*. Karena adanya teknologi penyimpanan yang mengalami perkembangan yang pesat, dan pengguna yang terus bertambah. Tetapi

seiring dengan perkembangan teknologi penyimpanan tidak diikuti dengan pengetahuan pengguna masalah teknis *SSD*. Gejala-gejala kerusakan *SSD* sangatlah mirip dengan kerusakan perangkat keras lain. Maka dilakukanlah untuk mendeteksi kerusakan *SSD* dengan metode *KNN*.

2. TINJAUAN PUSTAKA

Dalam bab kajian teoritis tentang teori-teori yang mendukung tema penelitian. Berikut adalah penjelasan tentang teori-teori yang mendukung penelitian.

2.1. Data Mining

Secara sederhana, data mining dapat diartikan mengekstrak atau menggali knowledge yang ada pada sekumpulan data (Nur Khormarudin, 2016). Menurut (Hermawati, 2013), Data mining adalah proses yang memperkerjakan satu atau lebih teknik pembelajaran komputer (*machine learning*) untuk menganalisis dan mengekstraksi pengetahuan secara otomatis. Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *database* (Thaniket dkk., 2019). Data mining biasanya menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database*. (Erwansyah, 2019) Data mining adalah serangkaian proses untuk menggali nilai tambahan berupa informasi yang selama ini tidak diketahui secara manual dari suatu basis data. Berikut adalah tahapan proses data mining :

1. *Preprocessing*. *Preprocessing* merupakan pengelompokan suatu data yang nantinya akan mempermudah dalam pengambilan data.

2. *Transformation*. Pada proses ini data akan digabungkan untuk melakukan proses mining dengan cara peringkasan.
3. *Data Mining*. Pada proses ini informasi dalam data akan dipilih menggunakan teknik, metode, atau algoritma tertentu sesuai dengan tujuan dari proses tersebut.
4. *Interpretation*. *Interpretation* merupakan proses menerjemahkan pola yang dihasilkan pada saat data mining, selain itu juga menguji apakah pola sesuai dengan hipotesa sebenarnya. Dapat disimpulkan bahwa data mining merupakan proses pencarian data secara otomatis untuk menemukan pola atau model dari suatu *database*.

2.2. Algoritma K-Nearest Neighbor

Penerapan algoritma *K-Nearest Neighbor* pada penelitian ini digunakan untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. *K-Nearest Neighbor* merupakan pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama dengan berdasarkan pencocokan bobot dari jumlah fitur yang ada. Algoritma *Nearest Neighbor* berdasarkan pada proses pembelajaran menggunakan analogi. Menurut (Rahmat Dian Nugraha dkk., 2020) *K-Nearest Neighbor* (KNN) merupakan metode yang termasuk kelompok dalam pengklasifikasian data yang sederhana dan mudah untuk pengimplementasian, efektif pada data yang lebih besar, dan dapat mengklasifikasikan data dengan tepat. Menurut (R. Nola and S. M Nurtanzis Sutoyo, 2016) rumus untuk menghitung bobot kemiripan (*similarity*) dengan *Nearest Neighbor* adalah sebagai berikut.

$$(T, S) = \frac{\sum_{i=1}^n f(T, S) * W_i}{W_i} \quad (1)$$

Keterangan :

t = kasus baru

s = kasus yang ada dalam penyimpanan

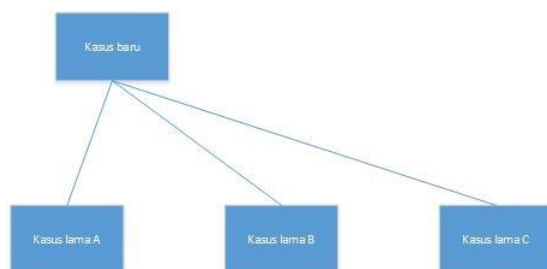
n = jumlah atribut setiap kasus

i = atribut individu (1 sampai dengan *n*)

f = fungsi *similarity* atribut 1 antara kasus *t* dan kasus *s*

w = bobot yang diberikan pada atribut ke *i*

Kedekatan biasanya berada pada nilai 0 sampai nilai 1. Nilai 0 artinya kedua kasus tidak mirip, sedangkan nilai 1 berarti kasus mutlak mirip.



Gambar 1. Ilustrasi Kedekatan Kasus Flowchart

Seperti tampak pada gambar 1 di atas terdapat 3 data lama yaitu A, B, dan C. Ketika ada data baru, maka data lama yang terdekat dengan data baru bisa dijadikan solusi dalam menghadapi permasalahan yang ada. Dari gambar di atas, kasus lama A yang akan digunakan karena memiliki jarak yang paling pendek.

2.3. Menentukan matrix jarak

Dalam menentukan titik data mana yang paling dekat dengan kueri tertentu, maka jarak antara titik kueri dan titik lainnya perlu dihitung. Salah satu cara yang bisa digunakan untuk membentuk batasan keputusan yaitu dengan menggunakan matrix jarak ini. Untuk menemukan titik serupa terdekat, kita bisa menggunakan perhitungan jarak seperti *Euclidean distance*, *Hamming distance*, *Manhattan distance*, dan *Minkowski distance*.

2.4. Mendefinisikan nilai K

Nilai *K* pada algoritma KNN berperan penting dalam menentukan berapa banyak tetangga yang akan diperiksa untuk menentukan klasifikasi dari titik kueri tertentu. Sebagai contoh yaitu jika *K*=1 maka *instance* akan ditugaskan kelas yang sama dengan tetangga paling dekat. Dengan mendefinisikan *K* bisa menjadi tindakan penyeimbang karena nilai yang berbeda akan menyebabkan *overfitting* atau *underfitting* nilai. Nilai *K* yang rendah memiliki varian yang tinggi namun bias yang rendah. Sedangkan nilai *K* yang tinggi dapat menyebabkan varian rendah dan bias yang tinggi.

Pilihan *K* bergantung pada data input karena data yang memiliki lebih banyak *outlier* atau *noise* kemungkinan akan bekerja lebih baik dengan nilai *k* yang lebih tinggi. *Outlier* adalah kasus atau data yang memiliki karakteristik unik yang terlihat sangat berbeda jauh dari observasi-observasi lainnya dan muncul dalam bentuk nilai ekstrim baik untuk sebuah variabel tunggal atau kombinasi (Ghozali, 2011). Secara keseluruhan angka ganjil sangat dianjurkan dalam pemilihan nilai karena untuk menghindari ikatan dalam klasifikasi. Strategi *cross validation* juga dapat digunakan untuk membantu kita memilih *K* yang optimal untuk data set yang kita miliki.

2.5. Alur cara kerja algoritma KNN

1. Menentukan parameter *K*= jumlah tetangga terdekat.
2. Melakukan pembobotan
3. Hitung jarak antara data yang akan dievaluasi dengan semua data pelatihan.
4. Urutkan jarak dari nilai paling terbesar ke yang terkecil
5. Tetapkan titik data baru ke kategori yang jumlah tetangganya. Paling banyak.
6. Model sudah siap

2.6. SSD (Solid State Drive)

SSD atau *Solid-State Drive* merupakan media penyimpanan data yang menggunakan memori tidak menguap (*nonvolatile memory*) sebagai media dan

tidak menggunakan media penyimpanan eksternal konvensional. Lain halnya dengan memori yang mudah menguap (*volatile memory*) seperti RAM, data yang sudah tersimpan pada SSD tidak akan hilang meskipun daya listriknya tidak ada. Menurut (Pranoto, 2020) penyimpanan SSD menggunakan *non volatile memory* yaitu memori yang datanya dapat ditulis dan dihapus, tetapi tetap ada walaupun dalam kondisi mati (*off*) karena SSD tidak menggunakan *disk magnetis* seperti *Harddisk* tradisional. Sedangkan menurut (Ramdhan, R.A, Y.Prayudi, dan B, 2017) SSD singkatan dari *Solid State Drive* atau *Solid State Disk*, adalah perangkat penyimpan data yang menggunakan rangkaian IC sebagai memori untuk menyimpan data atau informasi. SSD bisa juga disebut versi canggih dari *USB Flash Drive* dengan kapasitas yang lebih besar dan bisa juga sebagai pengganti *harddisk* yang biasa digunakan pada perangkat komputer.

Berikut beberapa gejala SSD yang akan mengalami kerusakan pada umumnya.

2.6.1. Error Bad Block

Yang dimaksud dari *Bad Block* pada SSD adalah situasi yang ada dimana komputer mencoba untuk membaca atau menyimpan *file*, tetapi membutuhkan waktu yang sangat lama dan akhirnya tidak tersimpan, sehingga akhirnya sistem menyerah dengan mengeluarkan notifikasi error. Gejala yang biasa terjadi saat *Bad Block* :

- Sebuah *file* tidak bisa terbaca atau ditulis di SSD
- File* sistem PC butuh perbaikan
- Aplikasi yang berjalan sering *Freeze* atau *Crash*
- Sering error ketika memindahkan *File*
- Sering lemot ketika membuka *file* besar

Bila komputer mengalami gejala-gejala yang sudah disebutkan di atas maka dianjurkan mencoba aplikasi *Hard Tune Pro* atau *Hard Disk Sentinel* dan diperiksa apakah ada error di *Bad Block*. Jika ada maka harus mempersiapkan *backup*.

2.6.1.1. File tidak bisa di Read atau Write

Ada 2 efek yang akan ditemui ketika SSD terjangkit *Bad Block* :

- Sistem akan mendeteksi *Bad Block* ketika *write* data *drive* dan oleh karena itulah ia akan menolak untuk *write* data.
- Sistem akan mendeteksi *Bad Block* setelah data di *write* dan oleh karena itu ia akan menolak *Read* data.

Pada situasi pertama data yang tidak pernah di *write* jadi data tidak korup. Biasanya sistem akan menyelesaikan secara otomatis. Dalam situasi ini Langkah yang harus diambil yaitu menggunakan *drive* lain untuk menyimpan *file* atau *copy* ke *cloud storage* atau *restart* komputer dan coba *save* ulang data tersebut.

Di situasi kedua, sayangnya data yang tersimpan tetap tidak bisa diambil. Ada beberapa metode yang bisa memperbaiki SSD dari *file*, tapi jangan terlalu

berharap. Karena *Bad Block* berarti data yang tersimpan di setiap *block* hilang selamanya.

2.6.1.2. File Sistem Membutuhkan Perbaikan



Gambar 2. Screenshoot saat mengalami *Recovery File*

Gambar di atas merupakan notifikasi error yang kadang-kadang akan muncul ketika tidak mematikan komputer dengan benar. Namun jika notifikasi tersebut muncul maka jangan dibiarkan saja karena hal tersebut bisa saja terjadi dari *Bad Block* atau kabel konektor dari SSD.

3. METODE PENELITIAN

3.1. Identifikasi Permasalahan

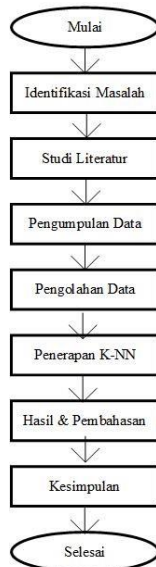
Berdasarkan hasil observasi yang dilakukan, masalah yang sering dihadapi pengguna teknologi *Solid State Drive (SSD)* adalah kesulitan untuk mengetahui bahwa SSD mereka sedang dalam kondisi rusak atau tidak. Kebanyakan dari pengguna SSD belum melakukan persiapan pengamanan data penting, namun tiba-tiba SSD sudah mengalami rusak berat. SSD yang sudah mengalami rusak total tidak bisa diperbaiki lagi, sehingga perlu dilakukan persiapan pengamanan data penting agar tidak mengalami kerugian yang banyak.

3.2. Jenis Penelitian

Jenis penelitian yang digunakan dalam penelitian ini adalah penelitian kuantitatif. Metode penelitian kuantitatif adalah salah satu jenis penelitian yang spesifikasinya adalah sistematis, terencana, dan terstruktur dengan jelas sejak awal hingga pembuatan desain penelitiannya.

3.3. Alur Metode Penelitian

Tahap pertama yang harus dilakukan yaitu mengidentifikasi masalah yang dihadapi yaitu tentang seberapa efektif sebuah metode dilakukan untuk mendiagnosa kerusakan pada SSD. Selanjutnya yaitu pencarian informasi terkait dengan teori-teori yang mendasari masalah pada penelitian yang dilakukan. Pada tahap pengumpulan data diperlukan data hasil wawancara dengan narasumber yaitu Junior's Komputer tentang diagnosa dan kerusakan pada laptop. Selanjutnya yaitu dilakukan pengolahan data yang sudah didapat pada saat wawancara. Kemudian dilakukan penerapan KNN yang akan dijelaskan pada bab hasil dan penjelasan. Langkah terakhir yang dilakukan yaitu menarik kesimpulan dari penelitian ini.



Gambar 3. Gambar Skema Tahapan Penelitian

3.4. Instrumen Pengumpulan Data

Jenis data yang digunakan dalam penelitian, yaitu data primer. Data primer adalah data dari penelitian yang diperoleh secara langsung dari narasumber sebagai pengetahuan dasar. Data ini diperoleh dari teknisi di Junior's Komputer yaitu tentang kerusakan-kerusakan SSD yang pernah terjadi dari tahun 2019-2020 yang nantinya akan diolah dalam penelitian ini. Berikut ini beberapa tahapan dalam pengumpulan data untuk memperoleh informasi yaitu :

3.4.1. Wawancara

Pada tahap ini penulis mengajukan beberapa pertanyaan kepada salah satu teknisi Junior's Komputer untuk mengetahui data dan informasi mengenai gejala yang dialami suatu SSD.

3.4.2. Observasi / Pengamatan

Observasi merupakan metode pengumpulan informasi atau data melalui pengamatan terhadap objek secara langsung maupun tidak langsung. Pada penelitian ini penulis melakukan pengamatan pengumpulan data tentang diagnose SSD di tempat servis.

3.5. Analisis Data

Tahapan analisis data ini meliputi penjabaran data yang diperoleh dari hasil wawancara pada Toko Junior's Komputer selama 2 tahun.

Tabel 1. Gejala yang diperoleh

No	Kode Gejala	Daftar Gejala
1	K01	SSD lama saat membuka drive
2	K02	Tidak bisa read dan write
3	K03	Sering recovery file
4	K04	Cras ketike booting
5	K05	Error Bad Block
6	K06	Tidak bisa write

No	Kode Gejala	Daftar Gejala
7	K07	SSD tidak muncul di windows tetapi terdeteksi di BIOS
8	K08	Komputer sering mati saat dipakai terlau lama
9	K09	File sering corrupt
10	K10	Saat membuka file besar mengalami recovery file
11	K11	Indikator SSD menyala tapi tidak terbaca di windows
12	K12	Health SSD menurun 30%
13	K13	Tidak bisa masuk ke sistem windows
14	K14	Alus listrik yang tidak stabil

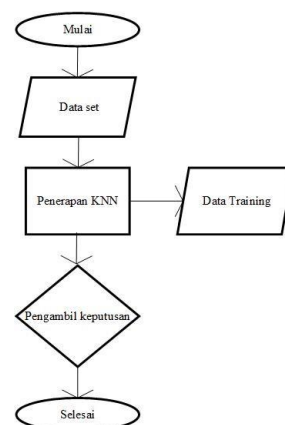
Pada tabel 1 sudah dituliskan gejala apa saja yang muncul pada SSD.

Tabel 2. Data Kasus Kerusakan SSD

No	Merek SSD	Status SSD
1	V-gen	Rusak
2	Gloway	Tidak
3	Sandisk	Rusak
4	Adata	Rusak
5	Midias	Rusak
...	...	...
46	Midias	Rusak
47	V-gen 240	Rusak
48	Sa860500	Rusak
49	Blue512	Rusak
50	Kingstone	Rusak

Pada tabel 2 disebutkan data kerusakan SSD yang pernah terjadi di tempat servis Junior's Komputer.

3.6. Rancangan Model



Gambar 3. Flowchart Penerapan KNN

Flowchart di atas menjelaskan tentang proses berjalannya penerapan aplikasi untuk mendeteksi sebuah kerusakan SSD, berikut adalah penjelasan dari setiap tahapan.

- Data set adalah data yang akan diuji hasil dari kuisioner pengguna SSD
- Penerapan KNN pada flowchart di atas memiliki beberapa alur proses sebagai berikut.



Gambar 4. Alur Proses Penerapan KNN

Hal pertama yang dilakukan adalah menentukan nilai K, kemudian melakukan pembobotan. Jika sudah dilakukan maka menghitung jarak yang akan dievaluasi. Kemudian urutkan jarak nilai paling terbesar ke paling terkecil. Langkah terakhir yaitu memilih tetangga paling banyak.

- c. Data training adalah data kerusakan SSD yang didapatkan di tempat servis
- d. Pengambil keputusan adalah pencocokan terhadap data yang sudah dikelola dari KNN dan Pakar

4. HASIL DAN PEMBAHASAN

4.1. Penerapan Penelitian

a. Preprocessing

Data gejala yang sudah dikumpulkan dilakukan pengelompokan dan diberikan kode pada setiap data yang sudah disebutkan pada tabel 1. Kemudian gejala tersebut diberikan nilai pembobotan dari setiap gejala yang ada seperti pada tabel berikut.

Tabel 3. Pembobotan Data Gejala Lama

No	Kode Gejala	Bobot Kerusakan	Daftar Gejala
1	K01	0,3	SSD lama saat membuka drive
2	K02	0,6	Tidak bisa read dan write
3	K03	0,3	Sering recovery file
4	K04	0,4	Cras ketike booting
5	K05	0,9	Error Bad Block
6	K06	0,6	Tidak bisa write
7	K07	0,4	SSD tidak muncul di windows tetapi terdeteksi di BIOS

4.2.2. Pembobotan

No	Kode Gejala	Bobot Kerusakan	Daftar Gejala
8	K08	0,5	Komputer sering mati saat dipakai terlalu lama
9	K09	0,3	File sering corrupt
10	K10	0,4	Saat membuka file besar mengalami recovery file
11	K11	0,6	Indikator SSD menyala tapi tidak terbaca di windows
12	K12	0,4	Health SSD menurun 30%
13	K13	0,7	Tidak bisa masuk ke sistem windows
14	K14	0,7	Alus listrik yang tidak stabil

Data yang ada pada tabel 3 adalah data gejala lama yang sudah diberikan nilai pembobotan kerusakan.

b. Transformation

Data gejala dan kerusakan SSD akan mengalami proses penggabungan data atau mentransformasikan data ke dalam proses dengan melakukan peringkasan dimana data akan dikelompokkan sesuai dengan atribut yang digunakan untuk proses data sebagai kriteria data yang menjadi target dalam proses data.

Tabel 4. Transformasi Data

No	Kode SSD	Gejala	Status
1	D01	K5,K12,K3,K8,K6	Rusak
2	D02	K11,K9,K8	Tidak Rusak
3	D03	K1,K4,K13	Rusak
4	D04	K9,K14,K1	Rusak
5	D05	K14,K9,K3,K7	Tidak Rusak
6	D06	K10,K2,K14	Rusak
7	D07	K6,K3,K9,K13	Rusak
8	D08	K5,K11,K9,K8	Rusak
9	D09	K2,K9,K8,K7	Rusak
10	D10	K14,K03,K13,K8	Tidak Rusak

Pada tabel di atas telah dikelompokkan SSD apa saja yang mengalami kerusakan atau tidak dilihat dari gejala yang dimunculkan.

4.2. Penerapan KNN

4.2.1. Menentukan Nilai K

Dalam menentukan nilai K tidak dibutuhkan rumus, melainkan menggunakan aplikasi *rapid miner* untuk mengetahui hasil maksimal adalah K3. Pilihan K akan sangat bergantung pada data input karena data dengan lebih banyak *outlier* atau *noise* kemungkinan akan berkinerja lebih baik dengan nilai K yang lebih tinggi. Secara keseluruhan, disarankan untuk memilih nilai K berupa angka ganjil untuk menghindari ikatan dalam klasifikasi

Kasus kerusakan SSD Samsung 970 Evo 500GB dan V-Gen 252GB.

Tabel 5. Kerusakan SSD Samsung 970 Evo 500GB

No	SSD	Status	K 1	K 2	K 3	K 4	K 5	K 6	K 7	K 8	K 9	K 10	K 11	K 12	K 13	K 14
1	S01			√			√							√		√
2	S02								√		√		√		√	√

Tabel di atas merupakan data kerusakan SSD Samsung 970 Evo 500GB yang didapatkan dari *quisioner* pengguna SSD tersebut.

1. Pembobotan SSD Samsung 970 Evo 500GB
 - a) Kode D01 s/d D10 dengan S01

Tabel 6. Kerusakan SSD Kode D01 s/d D10 dengan S01

No	SSD	Status	K 1	K 2	K 3	K 4	K 5	K 6	K 7	K 8	K 9	K 10	K 11	K 12	K 13	K 14
1	D01	Rusak	√							√				√		
	S01			√			√							√		√
Pembobotan			0	1	0	0	1	0	0	0	0	0	0	1	0	0
2	D02	Tidak			√								√			
	S01			√			√							√		
Pembobotan			0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9	D09	Rusak			√		√	√		√				√		
	S01			√			√							√		√
Pembobotan			0	0	0	0	1	0	0	0	0	0	0	1	0	0
10	D10	Tidak								√	√		√			
	S01			√			√							√		√
Pembobotan			0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabel di atas merupakan cara untuk mencari nilai kemiripan SSD, setelah mencari nilai bobot di SSD baru kemudian nilai bobot tersebut akan dihitung dengan rumus *similarity* yang ada di C. Perhitungan jarak akan devaluasi.

S01 adalah SSD Samsung yang belum terdeteksi di tempat servis Junior's Komputer. D1 s/d D10 adalah SSD V-Gen yang sudah terdeteksi di tempat servis Junior's Komputer.

4.2.3. Jarak Jumlah K SSD V-Gen 252GB

Tabel 7. Kerusakan SSD Kode D01 s/d D10 dengan S02

No	SSD	Status	K 1	K 2	K 3	K 4	K 5	K 6	K 7	K 8	K 9	K 10	K 11	K 12	K 13	K 14
1	D01	Rusak	√							√				√		
	S02								√		√		√		√	√
Pembobotan			0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	D02	Tidak			√								√			
	S02								√		√		√		√	√
Pembobotan			0	0	0	0	0	0	0	0	0	0	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9	D09	Rusak			√		√	√		√				√		
	S02								√		√		√		√	√
Pembobotan			0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	D10	Tidak								√	√		√			
	S02								√		√		√		√	√
Pembobotan			0	0	0	0	0	0	0	0	1	0	1	0	0	0

Tabel di atas merupakan cara untuk mencari nilai kemiripan SSD, setelah mencari nilai bobot di SSD baru kemudian nilai bobot tersebut akan dihitung dengan rumus *similarity* yang ada di C. Perhitungan jarak akan devaluasi.

S02 adalah SSD Samsung yang belum terdeteksi di tempat servis Junior's Komputer. D01 s/d D10 adalah SSD V-Gen yang sudah terdeteksi di tempat servis Junior's Komputer.

4.2.4. Mengurutkan Jarak Dari Nilai Terbesar ke yang Terkecil

1) Samsung 970 Evo 500GB

Tabel 26. Mengurutkan Jarak Dari Nilai Terbesar

Samsung 970 Evo 500GB		
D01	$0,307 \times 100\% = 30,7\%$	6
D02	0	9
D03	$0,409 \times 100\% = 40,9\%$	5
D04	$0,428 \times 100\% = 42,8\%$	4
D05	$0,235 \times 100\% = 23,5\%$	7
D06	$0,88 \times 100\% = 88\%$	1
D07	0	8
D08	$0,789 \times 100\% = 78,9\%$	2
D09	$0,481 \times 100\% = 48,1\%$	3
D10	0	10

Tabel di atas merupakan tabel yang berisi nilai persen dari kemiripan kerusakan SSD Samsung dengan data SSD yang sudah ada di tempat servis. Sebagai contoh Samsung 970 mengalami gejala kerusakan sama dengan SSD V-Gen di tabel 2 adalah

2) V-Gen 252GB

Tabel 27. Mengurutkan Jarak Dari Nilai Terbesar

V-Gen 252 GB		
D01	0	0
D02	$0,666 \times 100\% = 66,6\%$	2
D03	$0,454 \times 100\% = 45,5\%$	4
D04	$0,285 \times 100\% = 28,5\%$	6
D05	0	9
D06	$0,4 \times 100\% = 40\%$	5
D07	$1 \times 100\% = 100\%$	1
D08	$0,210 \times 100\% = 21\%$	7
D09	0	8
D10	$0,642 \times 100\% = 64,2\%$	3

30,7%. Sedangkan yang dicari adalah nilai persen paling besar yaitu SSD D06 adalah SSD Eyotal. Data SSD Eyota rusak maka dapat disimpulkan bahwa SSD Samsung 970 rusak. Dan data yang ada pada SSD V-Gen 252Gb adalah kebalikannya

4.2.5. Memilih Tetangga Paling Banyak

1) Samsung 970 Evo 500GB

Tabel 28. Memilih Tetangga Paling Banyak

Data Testing	Jarak Antar Data	Status	Urutan Jarak
D01	$0,307 \times 100\% = 30,7\%$	Rusak	6
D02	0	Tidak	9
D03	$0,409 \times 100\% = 40,9\%$	Rusak	5
D04	$0,428 \times 100\% = 42,8\%$	Rusak	4
D05	$0,235 \times 100\% = 23,5\%$	Rusak	7
D06	$0,88 \times 100\% = 88\%$	Rusak	1
D07	0	Tidak	8
D08	$0,789 \times 100\% = 78,9\%$	Rusak	2
D09	$0,481 \times 100\% = 48,1\%$	Rusak	3
D10	0	Tidak	10

Pada tabel di atas warna biru merupakan data training dengan status rusak, warna hijau menandakan data training dengan status tidak rusak,

sedangkan warna kuning menandakan K. Kalau warna kuningnya 3 maka K=3 sedangkan warna kuningnya 1 maka K=1.

2) V-Gen 252GB

Tabel 29. Memilih Tetangga Paling Banyak

Data Testing	Jarak Antar Data	Status	Urutan Jarak
D01	0	Rusak	0
D02	$0,666 \times 100\% = 66,6\%$	Tidak	2
D03	$0,454 \times 100\% = 45,5\%$	Rusak	4
D04	$0,285 \times 100\% = 28,5\%$	Rusak	6
D05	0	Rusak	9
D06	$0,4 \times 100\% = 40\%$	Rusak	5
D07	$1 \times 100\% = 100\%$	Tidak	1
D08	$0,210 \times 100\% = 21\%$	Rusak	7
D09	0	Rusak	8
D10	$0,642 \times 100\% = 64,2\%$	Tidak	3

Pada tabel di atas warna biru merupakan data training dengan status rusak, warna hijau menandakan data training dengan status tidak rusak,

sedangkan warna kuning menandakan K. Kalau warna kuningnya 3 maka K=3 sedangkan warna kuningnya 1 maka K=1.

4.3. Solusi

Tabel 30. Solusi

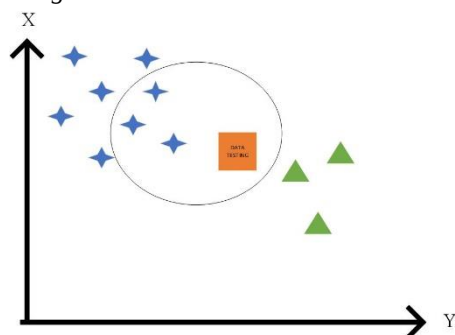
No	Solusi untuk SSD yang didiagnosa rusak
1	SSD tidak bisa diperbaiki jadi harus beli baru bila rusak
2	SSD kalau sudah didiagnosa rusak, SSD bisa tetap jalan tetapi siapkan <i>backup</i> data
3	Carilah SSD yang <i>writenya</i> lumayan banyak. Biasanya SSD yang bermerek bagus <i>writenya</i> banyak. Contohnya Samsung, Seagate, Western Digital, Adata, SSD Corsair, dan lain-lain

Tabel di atas merupakan solusi untuk mengetahui SSD mengalami kerusakan atau tidak tanpa harus pergi ke tempat servis. Contohnya yaitu SSD masih bisa dipakai tetapi setelah dilakukan tes pada aplikasi dengan gejala yang sudah dialami maka komputer menampilkan peringatan bahwa SSD tersebut rusak, maka solusinya adalah seperti yang tertera di tabel. Solusi terakhir yang bisa diberikan adalah membeli atau mengganti SSD lama ke yang terbaru.

4.4. Hasil KNN dari perhitungan

Hasil dari perhitungan KNN sebagai berikut.

a. Samsung 970 Evo 500Gb



Gambar 5. Hasil KNN Samsung 970 Evo 500Gb

Keterangan :

X = class data SSD rusak

Y = class data SSD tidak rusak

Simbol bintang = data yang rusak

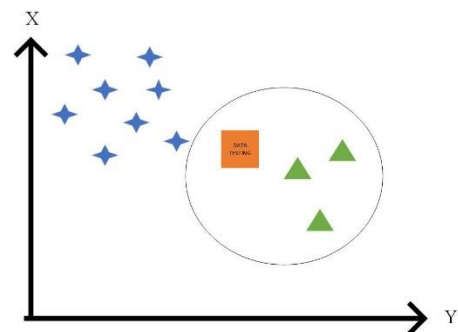
Symbol segitiga = data set yang

Simbol persegi = data testing

Simbol lingkaran = jumlah nilai K3

Gambar 5 adalah hasil deteksi SSD Samsung 970 Evo 500Gb dengan menggunakan algoritma KNN. Bisa disimpulkan bahwa SSD Samsung 970 evo 500Gb dengan jarak tetangga terdekatnya dalam menggunakan nilai K3 adalah SSD tersebut rusak. Karena nilai k pada algoritma KNN mendefinisikan berapa banyak tetangga yang akan diperiksa untuk menentukan klasifikasi titik kueri tertentu.

b. V-gen 252 Gb



Gambar 6. Hasil KNN V-gen 252Gb

Keterangan :

X = class data SSD rusak

Y = class data SSD tidak rusak

Simbol bintang = data yang rusak

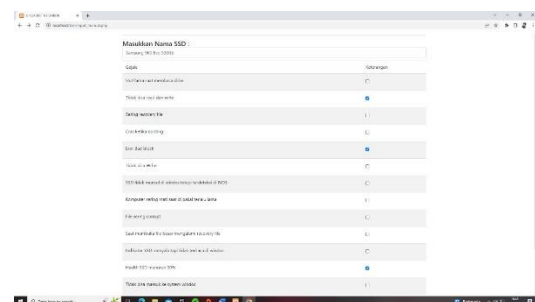
Symbol segitiga = data set yang

Simbol persegi = data testing

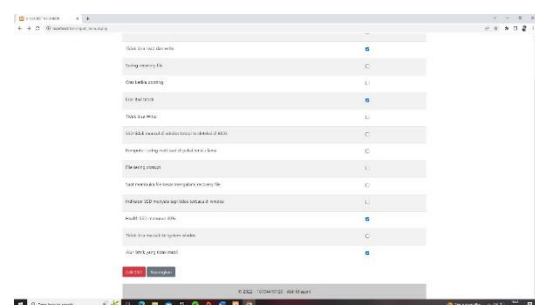
Simbol lingkaran = jumlah nilai K3

Gambar 6 adalah hasil deteksi SSD V-gen 252Gb dengan menggunakan algoritma KNN. Bisa disimpulkan bahwa SSD V-gen 252Gb dengan jarak tetangga terdekatnya dalam menggunakan nilai K3 adalah SSD tersebut rusak. Karena nilai k pada algoritma KNN mendefinisikan berapa banyak tetangga yang akan diperiksa untuk menentukan klasifikasi titik kueri tertentu.

4.5. Implementasi KNN dengan menggunakan php



Gambar 7. Tampilan Depan



Gambar 8. Tampilan Depan

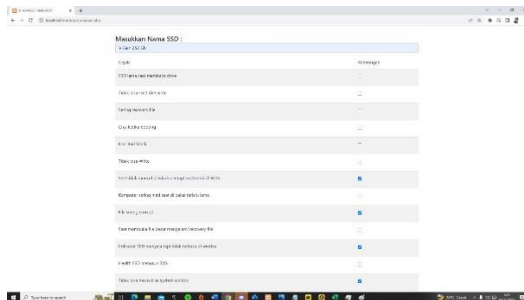
Pada gambar 7 dan 8 adalah tampilan pertanyaan gejala saat mau mendeteksi sebuah SSD pada tipe

Samsung 960 Evo 500Gb. Pada menu tersebut berisi kolom nama SSD dan gejala apa saja yang dialami. Kemudian baru bisa di cek.

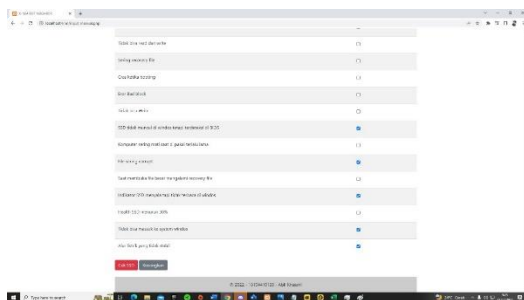


Gambar 9. Tampilan Hasil

Pada gambar 9 adalah tampilan hasil dari pertanyaan gejala yang dialami pada gambar 7 dan 8. Hasil akhir dari aplikasi yaitu *SSD Samsung 970 Evo 500GB* mengalami kerusakan dengan gejala *Error Bad Block*, Tidak bisa *read* dan *write*, *Health SSD* menurun 30%, dan alur listrik yang tidak stabil.



Gambar 10. Tampilan Depan



Gambar 11. Tampilan Depan

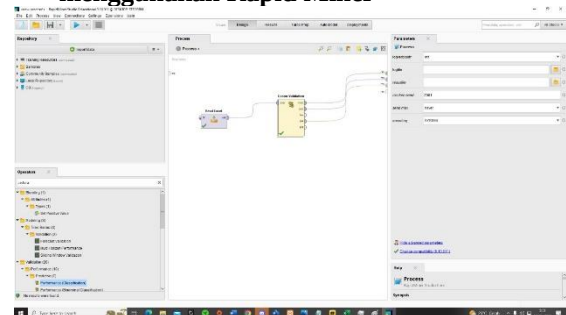
Pada gambar 10 dan 11 adalah pertanyaan saat mau masuk mendeteksi sebuah SSD pada tipe V-gen 252Gb. Pada menu tersebut berisi kolom nama SSD dan gejala apa saja yang dialami. Kemudian baru bisa di cek.



Gambar 12. Tampilan Hasil

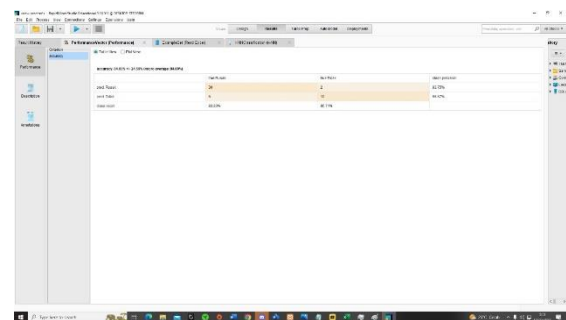
Pada gambar 12 adalah tampilan hasil dari pertanyaan gejala yang berada pada gambar 10 dan 11. Hasil akhir dari aplikasi yaitu *SSD V-Gen 252GB* tidak rusak hanya saja mengalami gejala *indicator SSD menyala tapi tidak terbaca di windows*.

4.6. Pengujian akurasi algoritma KNN menggunakan Rapid Miner

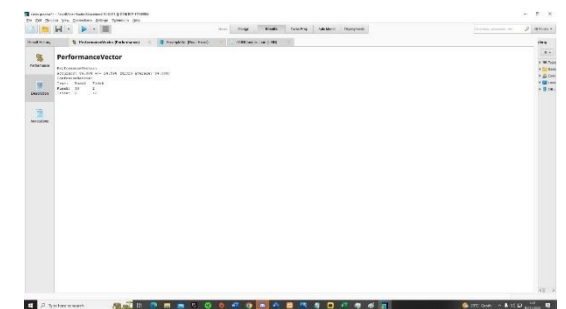


Gambar 13. Desain Pengujian Akurasi

Gambar 13 adalah desain pengujian akurasi dari algoritma KNN di rapid miner. Pengujian data SSD yang di tempat servis dengan menggunakan algoritma KNN menggunakan cross validasi.



Gambar 14. Hasil Pengujian Akurasi



Gambar 15. Hasil Pengujian Akurasi

Pada gambar 14 dan 15 menunjukkan hasil pengujian akurasi algoritma KNN dalam mendeteksi kerusakan SSD sebesar 84%.

5. KESIMPULAN DAN SARAN

Implementasi algoritma *K-NEAREST NEIGHBOR* dapat digunakan baik dan cocok dalam proses mengklasifikasi kerusakan SSD. Yang sebelumnya menentukan rusak tidak rusaknya SSD dengan cara manual yang sangat lama seperti pengujian komputer atau laptop selama berkali kali menggunakan *software harddisk sentinel*. Menyalakan laptop apakah ada eror saat menjalankan aplikasi - aplikasi berat dan saat mendeteksi dengan cara manual itu memakan waktu lama bisa sehari-hari. Sekarang dengan adanya algoritma *K-NEAREST NEIGHBOR* untuk mempercepat mendeteksi kerusakan SSD dan sudah di lakukan 20x pengujian hasil test tersebut sangat mirip dengan dengan teknisi yang melakukan mendeteksi secara manual. Seperti saat pengujian SSD Samsung 960 Evo 500GB dengan gejala *error bad block*, tidak bisa read dan write, health ssd menurun, alur listrik tidak stabil. Hasil dari manual sama menggunakan aplikasi sama ialah rusak. Maka disimpulkan bahwa algoritma *K-NEAREST NEIGHBOR* sangat bisa mendeteksi kerusakan SSD. Hasil dari pengujian algoritma *K-NEAREST NEIGHBOR* untuk mengetahui kerusakan SSD dengan menggunakan RAPID MINER sebesar 84% dan sudah diterapkan di aplikasi berbasis web dengan php.

Untuk meningkatkan dan menyempurnakan penelitian yang telah dilakukan. Peneliti memberikan saran yaitu dengan adanya penelitian ini dapat dikembangkan dengan cara menggabungkan ataupun membandingkan dengan algoritma klasifikasi yang lainnya agar mendapatkan hasil prediksi yang telah bagus.

DAFTAR PUSTAKA

- [1] Erwansyah, K. (2019). Implementasi Data Mining Untuk Menganalisa Hubungan Data Penjualan Produk Bahan Kimia Terhadap Persediaan Stok Barang Menggunakan Algoritma FP (Frequent Pattern) Growth Pada PT . Grand Multi Chemicals. *Jurnal Teknologi Sistem Informasi Dan Sistem Komputer TGD (J-SISKO TECH)*, 2(2), 30–40.
- [2] Ghazali, I. (2011). *Aplikasi Analisis Multivariate dengan Program SPSS*. (Edisi Ket). Semarang : Badan Penerbit Universitas Diponegoro.
- [3] Hermawati, F. A. (2013). *Data Mining*. Yogyakarta : Andi.
- [4] Nur Khormarudin, A. (2016). Teknik Data Mining: Algoritma K-Means Clustering. *Jurnal Ilmu Komputer*, 1–12.
<https://ilmukomputer.org/category/datamining/>
- [5] Pranoto, W. (2020). *Penerapan Metode Live Forensics Untuk Akuisisi Pada Solid State Drive (SSD) NVMe Fungsi TRIM*.
- [6] R. Nola and S. M Nurtanzis Sutoyo. (2016). Case Based Reasoning Untuk Mendeteksi Kerusakan Harddisk. *Jurnal Sustainable*, 5(01), 15–21.
- [7] Rahmat Dian Nugraha, A., Auliasari, K., & Agus Pranoto, Y. (2020). IMPLEMENTASI METODE K-NEAREST NEIGHBOR (KNN) UNTUK SELEKSI CALON KARYAWAN BARU (Studi Kasus : BFI Finance Surabaya). *JATI (Jurnal Mahasiswa Teknik Informatika)*, 4(2), 14–20.
<https://doi.org/10.36040/jati.v4i2.2656>
- [8] Ramdhan, R.A, Y.Prayudi, dan B, S. (2017). Implementasi dan Analisis Forensika Digital pada Fitur TRIM Solid State Drive. *Teknomatika*, 9(2).
- [9] Syukri Mustafa, M., & Wayan Simpen, I. (2019). Implementasi Algoritma K-Nearest Neighbor (KNN) Untuk Memprediksi Pasien Terkena Penyakit Diabetes Pada Puskesmas Manyampa Kabupaten Bulukumba. *Februari, 2019*(1), 1–10.
- [10] Thaniket, R., Kusriani, & Luthf, E. T. (2019). Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Algoritma Support Vector Machine. *JURNAL FATEKSA: Jurnal Teknologi Dan Rekayasa*, 13(2), 69–83.