Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 6, No. 6, Juni 2022, hlm. 2754-2759 http://j-ptiik.ub.ac.id

Naïve Bayes untuk Klasifikasi Pergantian *Operating System* pada *Personal Computer* di Bank X

Syifa Namira Neztigaty[1], Dian Eka Ratnawati[2], Dany Primanita Kartikasari[3]

Program Studi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]syfnmr@student.ub.ac.id, [2]dian_ilkom@ub.ac.id, [3]dany.jalin@ub.ac.id

Abstrak

Bank X merupakan salah satu bank terbesar di Indonesia. Tim Desktop yang merupakan salah satu tim yang melakukan identifikasi resiko dan evaluasi secara periodik pada PC. Melalui evaluasi tersebut Tim Desktop menemukan banyak operating system pada PC di Bank X yang telah end-of-support. Kondisi ini menimbulkan resiko jika ada bug atau celah keamanan pada operating system dapat menyebabkan pencurian data nasabah atau data transaksi. Agar terhindar dari hal tersebut seluruh operating system pada PC yang telah end-of-support harus diganti ke operating system yang baru. Saat ini aktivitas pergantian sudah dilakukan, namun masih dilakukan secara manual dan belum efektif. Proses tersebut telah berhasil mengganti ±5.000 PC. Sehingga melalui hal tersebut diperlukan sistem untuk melakukan klasifikasi data PC. Melalui sistem, Tim Desktop akan mendapatkan hasil kelas dari PC. Hasil kelas tersebut dihitung menggunakan metode klasifikasi data mining, yaitu Naïve Bayes. Dari data atribut PC yang belum memiliki kelas, sistem akan melakukan pengolahan data. Data dihitung dengan menggunakan fungsi yang ada pada controller, dengan mengakses database data PC yang telah memiliki kelas. Informasi mengenai hasil klasifikasi ditampilkan pada laman hasil klasifikasi dan rekap data PC ditampilkan pada laman rekap data PC. Dari hasil pengujian metode ini diperoleh nilai akurasinya sebesar 92.8371%.

Kata kunci: *data mining, klasifikasi, naïve bayes, confusion matrix, operating system*

Abstract

Bank X is one of the largest banks in Indonesia. The Desktop Team, which is one of the teams that carry out risk identification and periodic evaluation on PCs. Through this evaluation, the Desktop Team found many operating systems on PCs at Bank X were end-of-support. This condition poses a risk if there are bugs or security holes in the operating system that can lead to data theft. In order to avoid this incident, the entire operating system of end-of-support PCs must be replaced with a newer operating system. Currently the operating system replacement activity has been carried out, but it is still done manually and has not been effective. The process has succeeded in replacing ±5,000 PCs. Through this system, the Desktop Team Desktop Team will get the class results from the PC. The data could be classified by using data mining classification method, namely Naïve Bayes. From the PC attribute data does not yet have a class, the system will perform data processing. Then the data is calculated using the existing functions on the controller, by accessing classified PC database. Information about the classification results is displayed on the prediction results page. From the results of testing the accuracy value is 92.8371%.

Keywords: *data mining, classification, nave bayes, confusion matrix, operating system*

1. PENDAHULUAN

Bank X merupakan salah satu bank terbesar di Indonesia saat ini. PC atau *Personal Computer* merupakan komponen sistem pendukung pencapaian target bisnis adalah perangkat PC atau *Personal Computer*. Saat ini populasi PC di Bank X mencapai ±50.000 PC.

Tim *Desktop* merupakan tim yang melakukan evaluasi terhadap PC di Bank X. Melalui evaluasi Tim *Desktop* menemukan banyak *operating system* pada PC yang telah *end-of-support*. Kondisi ini dapat menimbulkan

resiko jika ada *bug* atau celah keamanan pada *operating system* tersebut seperti virus atau malware yang dapat menyebabkan gangguan operasional maupun terjadi pencurian data nasabah atau data transaksi. Agar terhindar dari hal tersebut seluruh *operating system* pada PC yang telah *end-of-support* harus diganti ke operating system yang baru. Saat ini terdapat ±15.000 PC yang telah berhasil diganti. Namun aktivitas tersebut sangat lambat, karena masih dilakukan secara manual dan belum efektif.

Melalui latar belakang yang telah ada, penulis tertarik melakukan klasifikasi terhadap pergantian OS pada PC di Bank X dengan menggunakan metode *naïve bayes*. Pertimbangan penggunaan metode *naïve bayes* karena *naïve bayes* memiliki nilai akurasi yang tinggi, performansi yang baik, cepat, serta efisien jika diimplementasikan pada jumlah data yang besar (Coastera, Yusa, Lediwara, & Sari, 2012). Hasil klasifikasi tersebut akan diimplementasikan dalam bentuk web sebagai sistem informasi manajemen pergantian OS pada PC di Bank X. Sistem tersebut akan mengklasifikasikan pergantian *operating system* pada *personal computer* di Bank X mengacu pada data yang sudah ada.

2. DASAR TEORI

2.1 Data Mining

Data Mining adalah proses penemuan pola dari data yang besar yang disimpan pada repository dengan menggunakan teknologi pengenalan pola, teknik statistik, maupun matematika (Larose, 2005). Melalui data mining dapat ditemukan pola-pola tersembunyi dan juga informasi mengenai prediksi yang mungkin tidak terlihat sebelumnya (Siregar & Puspabhuana, 2018).

2.2 Naïve Bayes

Naïve Bayes ini dapat menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari data set yang ada (Sumpena & H, 2019). Penggunaan metode ini terbukti telah memiliki akurasi dan kecepatan yang tinggi saat data pada database mengaplikasikan metode tersebut dengan data jumlah besar(Aswendy, 2016). Berikut merupakan persamaan yang digunakan pada motode *naïve bayes*(Bustami, 2014):

$$P(H \mid X) = \frac{P(X \mid H) \times P(H)}{P(X)} \quad (1)$$

Keterangan :

X : Data dengan kelas yang belum diketahui

H : Hipotesis data X merupakan suatu kelas spesifik

P $(H|X)$: Probabilitas hipotesis H berdasar kondisi X (posteriori probability)

P (H) : Probabilitas hipotesis H (prior probability)

P $(X|H)$: Probabilitas X berdasarkan kondisi pada hipotesis H

P (X) : Probabilitas X

2.3 Confusion Matrix

Confusion matrix merupakan jumlah prediksi tidak tepat dan tepat yang terbentuk oleh model dengan membandingkan data uji pada hasil klasifikasi sebenarnya. (Syahputra et al., 2018). Berikut persamaan nilai akurasi dari confusion matrix (Han, Kamber, & Pei, 2012):

$$Accuracy = \frac{TN+TP}{TN+FN+FP+TP} \quad (2)$$

3. METODOLOGI PENELITIAN

Tahapan pada penelitian ini dapat dilihat pada Gambar 1.

Gambar 1. Metodologi Penelitian

Metodologi penelitian dimulai dari melakukan identifikasi masalah pada Bank X. Kemudian tahapan dilanjutkan dengan

pengumpulan studi literatur, setelah itu tahap pengumpulan data dengan data yang digunakan adalah data yang didapatkan dari Tim *Desktop* Bank X. Selanjutnya tahap pengolahan data dengan metode naïve bayes dan perancangan sistem dengan pembentukan diagram sistem. Tahap berikutnya yaitu implementasi sistem dan dilanjutkan dengan pengujian dan analisis terhadap hasil implementsi. Tahap terakhir berupa kesimpulan dan saran.

3.1 Analisis Kebutuhan

Pada tahap ini merupakan tahap identifikasi masalah pada Bank X. Masalah yang ditemukan terdapat banyak *personal computer* pada Bank X yang *operating system-nya* telah *end-of-support*. Melalui analisis ini terdapat beberapa fungsi yaitu, login, menambah data training, edit data training, melihat data training, melakukan klasifikasi, melihat rekap PC, melihat rekap PC, menguji performa, dan register user.

3.2 Pengolahan Data

Data tersebut merupakan data PC Bank X sejak tahun 2004 – 2018 dengan total 19.581 data. Data yang diberikan oleh pihak Bank X terdiri dari 11 atribut data, yaitu *Row Number, Device Name, Type, OS Name, Date,* Umur PC, *RAM, Processor Type, Available Disk, System Manufacturer,* dan Ganti OS.

3.3 Perancangan Sistem

3.1.1 Use Case Diagram

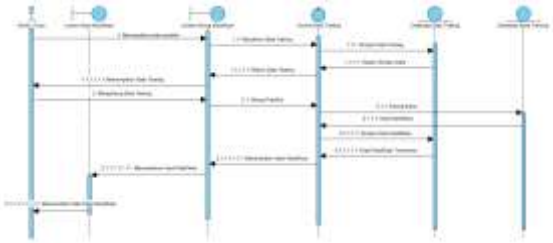
Gambar 2. *Use Case* Diagram

Gambaran *use case* pada sistem ini dapat dilihat pada Gambar 2. Selain itu penjelasan dari gambar dideskripsikan pada Tabel 1.

Tabel 1. Kebutuhan Fungsional

Kode	Nama	Deskripsi	Aktor
UCD-1	Login	Aktor dapat melakukan login ke sistem.	Admin, User
UCD-2	Edit data *training*	Sistem dapat mengubah data *training* yang di-*input*-kan oleh aktor.	Admin
UCD-3	Melihat data *training*	Sistem dapat menampilkan data *training* yang ingin dilihat oleh aktor.	User
UCD-4	Mengklasifikasi PC	Sistem dapat melakukan klasifikasi dari data yang di-*input*-kan oleh aktor.	Admin, User
UCD-5	Melihat hasil klasifi-kasi PC	Sistem dapat menampilkan data yang telah diklasifikasi yang ingin dilihat oleh aktor.	Admin, User
UCD-6	Menguji performa	Sistem dapat melakukan uji performa perhitungan dari data yang ingin diketahui oleh aktor.	Admin, User
UCD-7	Melihat rekap PC	Sistem dapat menampilkan data rekap PC yang ingin dilihat oleh aktor.	Admin, User
UCD-8	Register user	Sistem dapat menambahkan data user sesuai dengan yang di-*input*-kan oleh aktor.	Admin

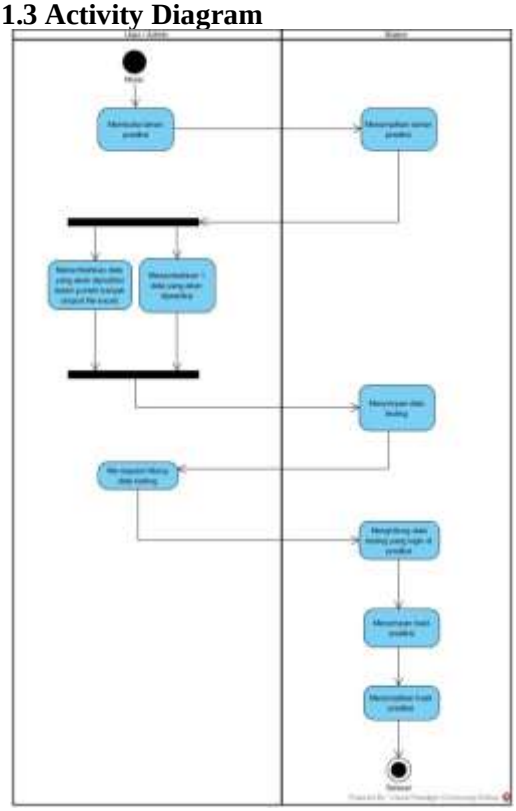
3.1.2 Sequence Diagram



Gambar 3. *Sequence* Diagram

Pada Gambar 3. merupakan gambaran pada laman hitung klasifikasi dimana admin/user akan melakukan klasifikasi dengan menginput data testing PC tersebut dan sistem akan melakukan perhitungan, kemudian sistem akan menyimpan data tersebut dan memperlihatkan data testing tersebut. Setelah itu aktor dapat melakukan hitung klasifikasi dengan data testing yang diinputkan sebelumnya. Sistem akan menghitung klasifikasi data testing dengan mengakses data training. Setelah berhasil dihitung hasil klasifikasi pada database data testing dan mengembalikan hasil klasifikasi tersebut ke laman hasil klasifikasi.

3.1.3 Activity Diagram



Gambar 4. Activity Diagram

Pada Gambar 4. merupakan gambaran pada laman hitung klasifikasi Activity diagram. Alur activity diagram tersebut dimulai dari sistem menampilkan laman hitung klasifikasi, dilanjutkan dengan menambahkan data yang akan diklasifikasi dimana data dapat di-*input*-kan satu per satu maupun dalam jumlah banyak menggunakan file excel, kemudian sistem menyimpan data tersebut sebagai data testing, dilanjutkan dengan aktor me-*request* hitung data testing tersebut, kemudian sistem menghitung data testing tersebut dan menyimpan hasil dari klasifikasi tersebut, dan dilanjutkan dengan menampilkan laman hasil klasifikasi, kemudian aktivitas selesai.

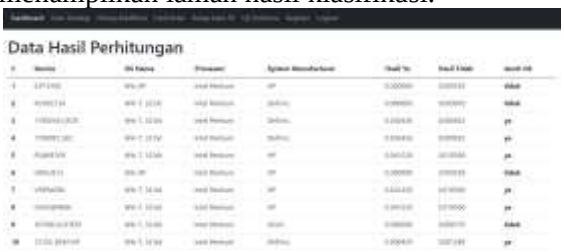
4. HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Implementasi sistem adalah hasil dari pengimplementasian sistem terhadap perancangan sistem. Implementasi berupa website yang dibangun menggunakan bahasa pemrograman HTML, PHP, dan CSS dengan penggunaan Laravel sebagai framework.

Gambar 5. Implementasi Laman Hitung Klasifikasi

Gambar 6. menunjukkan laman tambah data jika ngin menambahkan data baru pada data testing untuk diklasifikasi kelas. Data yang ditambahkan pada laman tambah data ini user diminta untuk memasukkan data *device name, type,* umur PC, *OS name, RAM, processor, system manufacturer, available harddisk size,* dan *date*. Setelah itu jika user menekan button "Tambah Data" data tersebut akan tersimpan ke database dan kembali ke laman hitung klasifikasi. Pada laman hitung klasifikasi data yang sudah dimasukkan sebelumnya dapat dilihat hasil klasifikasinya dengan menekan button "Hitung" pada Gambar 7. Setelah menekan button tersebut sistem akan menampilkan laman hasil klasifikasi.



Gambar 6. Implementasi Laman Hasil Klasifikasi

Pada laman ini ditampilkan laman uji performa metode, dimana user dapat

memasukkan nilai presentase data testing untuk diuji. Setelah memasukkan jumlah presentasenya jika user menekan *button* "Submit" sistem akan menampilkan seperti pada Gambar 8. Gambar 9 ini menampilkan hasil uji performa metode dengan tabel *confusion matrix* dan hasil akurasinya.

Gambar 7. Implementasi Laman Hasil Uji Performa

Gambar 8. Implementasi Laman Hasil Uji Performa

4.2 Pengujian dan Analisis

Confusion matrix merupakan salah satu metode pengujian yang dapat melakukan evaluasi pada metode ini. Data yang digunakan sebagai pegujian ini merupakan dataset yang didapatkan dari data testing yaitu data yang telah memiliki kelas. Total dataset yang digunakan sebanyak 8.931 data. Perhitungan *confusion matrix* menggunakan tipe *test precentage split*. Pengujian dilakukan dengan 3 skenario berbeda. Pertama dilakukan *precentage split* 80:20. Hasil uji tersebut dapat dilihat pada Gambar 9.

Gambar 9. Hasil Uji Performa *Percentage Split 80:20*

Skenario pertama *precentage split* 80:20, dengan 7.144 data training dan 1.787 data testing.

Tabel 2. Confusion Matrix *Precentage Split* 80:20

Actual class	***Predicted class***	
	Ya (*Positive*)	**Tidak (*Negative*)**
Ya (*Positive*)	797	9
Tidak (*Negative*)	119	862

Tabel 2. merupakan hasil dari perhitungan *confusion matrix* dengan tipe test *precentage split* 80:20. Dari Tabel 2. nilai akurasi dapat dicari dengan menggunakan persamaan (1).

$$Accuracy = \frac{TN+TP}{TN+FN+FP+TP}$$
$$Accuracy = \frac{862+797}{862+9+119+797}$$
$$= 0.928371$$

Berdasarkan perhitungan dari persamaan, nilai akurasi yang diporel 0.928371. Hasil tersebut sama dengan hasil pada gambar 9. yaitu 92.8371%.

Pengujian kedua dilakukan *precentage split* 75:25. Hasil uji tersebut dapat dilihat pada Gambar 10.

Gambar 10. Hasil Uji Performa *Percentage Split 75:25*

Skenario pertama *precentage split* 75:25, dengan 6.697 data training dan 2.234 data testing.

Tabel 3. Confusion Matrix *Precentage Split* 75:25

Actual class	***Predicted class***	
	Ya (*Positive*)	**Tidak (*Negative*)**
Ya (*Positive*)	1006	165
Tidak (*Negative*)	2	1061

Tabel 3. merupakan hasil dari perhitungan confusion matrix dengan tipe test *precentage split* 75:25. Dari Tabel 3. nilai akurasi dapat dicari dengan menggunakan persamaan (1).

$$Accuracy = \frac{TN+TP}{TN+FN+FP+TP}$$
$$Accuracy = \frac{1061+1006}{1061+165+2+1006}$$
$$= 0.925246$$

Berdasarkan perhitungan dari persamaan, nilai akurasi yang diporel 0.925246. Hasil tersebut sama dengan hasil pada gambar 10. yaitu 92.5246%.

Pengujian kedua dilakukan *precentage split* 70:30. Hasil uji tersebut dapat dilihat pada gambar 11.

Gambar 11. Hasil Uji Performa *Percentage Split 70:30*

Skenario pertama *precentage split* 75:25,

dengan 6.251 data training dan 2.680 data testing.

Tabel 4. Confusion Matrix *Precentage Split* 70:30

Actual class	***Predicted class***	
	Ya (*Positive*)	**Tidak (*Negative*)**
Ya (*Positive*)	1274	197
Tidak (*Negative*)	6	1203

Tabel 4. merupakan hasil dari perhitungan confusion matrix dengan tipe test *precentage split* 70:30. Dari Tabel 4. nilai akurasi dapat dicari dengan menggunakan persamaan (1).

$$Accuracy = \frac{TN+TP}{TN+FN+FP+TP}$$

$$Accuracy = \frac{1274+1203}{1274+197+6+1203}$$

$$= 0.924253$$

Berdasarkan perhitungan dari persamaan, nilai akurasi yang diporel 0.924253. Hasil tersebut sama dengan hasil pada gambar 12. yaitu 92.4253%.

Tabel 5. Perbandingan Nilai Akurasi

Percentage Split	**Nilai Akurasi**
Percentage Split 80:20	92.8371%
Percentage Split 75:25	92.5246%
Percentage Split 70:30	92.4253%

Pada Tabel 5 merupakan hasil akurasi tersebut *Percentage Split* 80:20 menunjukkan nilai akurasi 92.8371%, *Percentage Split* 75:25 menunjukkan nilai akurasi 92.5246%, dan *Percentage Split* 70:30 menunjukkan nilai akurasi 92.4253%. Melalui hasil uji tersebut didapatkan bahwa semakin besar presentase data training, semakin tinggi nilai akurasinya.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diambil berdasarkan hasil penelitian yaitu untuk melakukan klasifikasi pada data pergantian *operating system* pada *personal computer* sistem melakukan *preprocessing* pada data. Perhitungan pada sistem dilakukan melalui fungsi yang telah dibuat di-*controller*. Fungsi tersebut dibentuk dengan memanfaatkan persamaan Naïve Bayes.

Pengujian nilai akurasi menggunakan confusion matrix dengan tipe *percentage split* data. Hasil nilai akurasi tertinggi yang dicapai adalah 92.8371%, dengan menggunakan Percentage Split 80:20. Melalui hasil uji tersebut didapatkan bahwa semakin besar presentase data training, semakin tinggi nilai akurasinya.

5.2 Saran

Saran yang dapat diberikan untuk penelitian berikutnya pada penelitian ini meliputi, pemrosesan data dengan menggunakan metode lainnya selain naïve bayes, sehingga dapat dibandingkan metode mana dengan hasil akurasi terbaik diantara metode-metode tersebut, pengujian dengan tipe test yang berbeda, sehingga dapat membandingkan tipe percentage split dengan tipe test yang lain, dan mengembangkan sistem yang telah dibangun sehingga proses perhitungan klasifikasi pada sistem dapat dilakukan lebih cepat.

6. DAFTAR PUSTAKA

Aswendy. (2016). Analisis Data Iklim Indonesia Menggunakan Aplikasi Weka Dengan Metode Klasifiksi. *Teknologi Rekayasa*, *21*(3), 217–228.

Bustami. (2014). *Penerapan Algoritma Naive Bayes Untuk Mengklasifikasi Data Nasabah Asuransi*. *8*(1), 884–898. https://doi.org/10.26555/jifo.v8i1.a2086

Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques. In *Data Mining: Concepts and Techniques*. https://doi.org/10.1016/C2009-0-61819-5

Larose, D. T. (2005). Discovering Knowledge in Data: An Introduction to Data Mining. In *Discovering Knowledge in Data: An Introduction to Data Mining*. https://doi.org/10.1002/0471687545

Siregar, A. M., & Puspabhuana, A. (2018). *DATA MINING: Pengolahan Data Menjadi Informasi dengan RapidMiner - Amril Mutoi Siregar, S.Kom., M.Kom. DAN Adam Puspabhuana, S.Kom., M.Kom. - Google Books*. Retrieved from https://books.google.co.id/books?id=rTlmDwAAQBAJ&printsec=frontcover&dq=data+mining+adalah&hl=en&sa=X&ved=2ahUKEwj39Y7Iz__rAhXPbX0KHaUTAfAQ6AEwAHoECAMQAg#v=onepage&q=data mining adalah&f=false

Sumpena, J., & H, N. K. (2019). Analisis Prediksi Kelulusan Siswa PKBM Paket C dengan Metoda Algoritma Naïve Bayes. *Tedc*, *13*(2), 127–133.