# KNN vs Naive Bayes Untuk Deteksi Dini Putus Kuliah Pada Profil Akademik Mahasiswa

**Vina Zahrotun Kamila*[1], Eko Subastian[2]**
[1]Program Studi Teknik Informatika, Universitas Mulawarman, Samarinda
[2]Program Studi Pendidikan Komputer, Universitas Mulawarman, Samarinda
e-mail: ***[1]vinakamila@fkti.unmul.ac.id**, [2]ekosebastian1989@gmail.com

***Abstrak***

*Penelitian ini membahas bagaimana perbandingan KNN dan Naive Bayes dalam memprediksi potensi putus kuliah pada mahasiswa. Data yang dijadikan variabel independen adalah data akademik yaitu nilai semester 1 hingga 6. Hasil dari penelitian ini diharapkan menjadi pedoman dalam menerapkan algoritma ke dalam sistem deteksi dini putus kuliah. Algoritma-algoritma ini diterapkan dengan library Scikit-learn pada Python. Nilai akurasi yang dihasilkan dari penelitian ini menunjukkan Naive Bayes (92%) lebih unggul dalam memprediksi status putus kuliah mahasiswa dibandingkan dengan algoritma KNN (85%). Namun perlu dilakukan penelitian lanjutan lagiuntuk menguji konsistensi dan akurasi pada data yang lebih besar dan lebih beragam.*

***Kata kunci****—KNN, Naive Bayes, Deteksi Putus Kuliah*

## 1. PENDAHULUAN

Berdasarkan laporan Statistik Pendidikan Tinggi Indonesia 2017 [1] dan 2018 [2] oleh PDDIKTI dari laman pddikti.ristekdikti.go.id, tingkat putus kuliah atau *dropout* di Indonesia mengalami peningkatan di tahun 2018. Tren peningkatan jumlah mahasiswa yang putus kuliah ini perlu ditekan dengan mewaspadai mahasiswa-mahasiswa yang memiliki potensi putus kuliah tersebut.

Karena pentingnya kewaspadaan tersebut, maka perlu dibangun suatu sistem deteksi dini mahasiswa putus kuliah, agar di semester 7 menjelang skripsi, mahasiswa-mahasiswa ini diberikan kontrol dan pengawasan lebih oleh pengelola program studi. Selain bertujuan menurunkan angka putus kuliah, deteksi dini ini diharapkan dapat meningkatkan kinerja program studi dan dapat meningkatkan poin akreditasi.

KNN (K-Nearest Neighbors) dan Naive Bayes merupakan algoritma yang termasuk dalam 10 algoritma datamining yang banyak digunakan dalam penelitian learning machine[3]. Pada penelitian sebelumnya, telah dilakukan prediksi predikat prestasi mahasiswa menggunakan KNN[4]. Prediksi dalam penelitian tersebut menggunakan faktor penentu jenis kelamin, umur, jenis tinggal, jumlah nilai mutu, dan jumlah satuan kredit SKS dengan hasil akurasi 82%. Penelitian lain berupa prediksi hasil pembelajaran mahasiswa dengan algoritma Naive Bayes dan C4.5[5]. Variabel yang digunakan antara lain jenis kelamin, umur, asal kota (domestik/ non-domestik), status sekolah (negeri/swasta), IPK semester 1-4, partisipasi kegiatan organisasi dan menghasilkan status kelulusan (terlambat/ tepat waktu/ cepat). Dalam penelitian ini algoritma Naive Bayes menghasilkan rata-rata akurasi dan presisi lebih tinggi dari algoritma C4.5. Beberapa algoritma data mining lain juga telah teruji setelah diterapkan sebagai algoritma prediksi dengan data akademik siswa/ mahasiswa [6][7].

Tujuan dari penelitian ini untuk mengetahui bagaimana perbandingan algoritma KNN dan Naive Bayes dalam memprediksi potensi putus kuliah pada mahasiswa program studi Pendidikan Komputer Universitas Mulawarman. Variabel yang digunakan adalah nilai-nilai

mata kuliah wajib yang diambil mahasiswa dari semester 1 sampai 6. Hasil perbandingan tersebut diharapkan dapat menjadi motivasi dalam menerapkan salah satu algoritma dalam sistem deteksi dini putus kuliah di lingkungan program studi Pendidikan Komputer Universitas Mulawarman. KNN dan Naive Bayes digunakan karena telah terbukti memiliki performansi yang baik dalam memprediksi prestasi siswa di Palestina[8]. Meskipun algoritma prediksi prestasi siswa/ mahasiswa biasanya dijalankan dengan Rapid Miner [8] atau Weka [9][10], penelitian ini memilih menggunakan library Scikit-learn untuk KNN dan Naive Bayes di Python.

Penggunaan KNN dan Naive Bayes dalam prediksi prestasi siswa memang bukan baru dalam penelitian data mining. Namun penelitin ini menekankan penggunaan variabel profil akademik, yaitu nilai-nilai mata kuliah (tanpa unsur non akademik seperti jenis kelamin atau umur) sebagai variabel independen sebagai kebaruan dalam penelitian. Dalam penelitian ini juga dapat juga dilihat apakah data profil akademik ini bisa dijadikan faktor penentu status potensi putus kuliah mahasiswa atau tidak.

## 2. METODE PENELITIAN

### *2.1 Data dan Variabel*

Data yang digunakan dalam penelitian ini adalah data akademik berupa nilai-nilai mata kuliah wajib semester 1 sampai 6 untuk mahasiswa angkatan 2013 hingga 2015. Dari data tersebut akan didapat status putus kuliah mahasiswa ‘aman’, ‘kurang aman’, ‘tidak aman’. Penetapan status ini berasal dari data mahasiswa putus kuliah dan data mahasiswa lulus dengan waktu tunggu skripsi (tepat waktu, terlambat, sangat terlambat). Data angkatan 2013-2014, sebagian data angkatan 2015 (untuk 2015 hanya yang lulus saja) merupakan data training, yang nantinya akan diujicobakan ke mahasiswa angkatan 2015 hingga 2017. Data mahasiswa lulus per bulan Juli 2019 adalah 23 orang yang merupakan 20,7% dari total of mahasiswa angkatan 2013 sampai 2015. Pengambilan keputusan didapat dari data yang tersedia. Mahasiswa yang putus kuliah sebelum memasuki semester ke-enamnya tidak dimasukkan dalam data training. Meskipun data yang diambil relatif kecil, namun hasil dari penelitian ini dapat menjadi tonggak awal untuk melakukan penelitian dengan data yang lebih banyak dan metode yang lebih beragam dalam memprediksi potensi kuliah mahasiswa.

Variabel dependen merupakan variabel yang diuji dan diukur dalam suatu penelitian. Variabel dependen bergantung dari variabel-variabel independen. Variabel independen dalam penelitian ini adalah nilai-nilai mata kuliah wajib semester 1 sampai 6. Data angkatan 2013 hingga 2015 dipilih karena pada tahun-tahun tersebut tidak terdapat perubahan kurikulum yang signifikan sehingga tidak ada perubahan nama atau jumlah mata kuliah wajib dan pilihan. Pada program studi Pendidikan Komputer, terdapat 45 mata kuliah wajib (termasuk PPL dan KKN) yang harus diambil agar dapat mengambil mata kuliah skripsi. Format data dalam penelitian dapa dilihat pada Tabel 1.

Tabel 1. Contoh Format Data

| **MKWajib1** | **MKWajib2** | **MKWajib3** | **...** | **MKWajib45** | **Status** |
|---|---|---|---|---|---|
| 3 | 4 | 3 | ... | 3 | Aman |
| 3.5 | 3.5 | 4 | ... | 4 | Hampir Putus Kuliah |
| ...dst | | | | | |

Nilai matakuliah yang dimaksud adalah nilai bobot (bukan nilai huruf) sesuai ketentuan akademik fakultas, dan belum dikalikan dengan jumlah SKS. Bagaimana representasi data training bisa dilihat pada Gambar 1.

| | A | B | C | D | E | F | AP | AQ | AR | AS | AT | AU | AV |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | MKW1 | MKW2 | MKW3 | MKW4 | MKW5 | MKW6 | MKW42 | MKW43 | MKW44 | MKW45 | Status | | |
| 2 | 3 | 3 | 4 | 3 | 3 | 3 | 3 | 4 | 3,5 | 3,5 | a | | |
| 3 | 3,5 | 3 | 2,5 | 3,5 | 3 | 3 | 3,5 | 4 | 3 | 3,5 | a | Ket : | |
| 4 | 3 | 3 | 3 | 2,5 | 2 | 3 | 3,5 | 4 | 3,5 | 3,5 | a | a : aman | |
| 5 | 3 | 2,5 | 2 | 2,5 | 3 | 3 | 3 | 4 | 3 | 3,5 | a | b : hampir DO | |
| 6 | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 4 | 3 | 3,5 | b | c : DO | |
| 7 | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 2 | 3 | 3,5 | b | | |
| 8 | 3 | 3 | 2 | 2 | 3 | 3 | 3 | 4 | 3 | 3,5 | a | | |
| 9 | 2,5 | 3 | 2 | 2 | 3 | 3 | 0 | 0 | 0 | 0 | c | | |
| 10 | 3 | 3 | 2,5 | 2,5 | 3 | 3 | 3 | 4 | 3,5 | 3,5 | a | | |
| 11 | 2,5 | 3 | 2,5 | 2,5 | 3,5 | 2,5 | 3 | 4 | 2,5 | 3 | a | | |
| 12 | 2,5 | 3 | 2,5 | 2,5 | 3,5 | 2,5 | 0 | 0 | 2,5 | 0 | c | | |
| 13 | 2,5 | 2,5 | 2,5 | 2,5 | 3,5 | 2,5 | 3 | 3 | 2,5 | 3 | b | | |
| 14 | 3 | 3 | 3 | 3,5 | 3,5 | 3 | 3,5 | 2 | 3 | 3,5 | b | | |
| 15 | 3 | 3 | 3 | 3,5 | 3 | 3 | 3 | 4 | 3 | 3,5 | a | | |
| 16 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 4 | 3 | 3,5 | b | | |
| 17 | 3 | 2,5 | 3 | 2,5 | 3 | 3 | 3 | 2 | 0 | 0 | c | | |
| 18 | 2,5 | 2,5 | 2,5 | 2,5 | 3,5 | 2,5 | 0 | 0 | 0 | 0 | c | | |
| 19 | 3 | 3,5 | 3,5 | 3,5 | 3 | 3 | 3 | 4 | 3 | 3,5 | a | | |
| 20 | 3,5 | 3 | 3 | 3,5 | 3,5 | 3 | 3,5 | 4 | 3 | 3,5 | b | | |
| 21 | 3,5 | 3,5 | 3 | 4 | 3,5 | 3 | 3 | 4 | 3 | 3,5 | a | | |
| 22 | 3 | 3 | 3 | 4 | 3,5 | 3 | 3 | 2 | 2,5 | 3 | b | | |
| 23 | 3,5 | 3,5 | 3 | 4 | 3,5 | 3 | 3 | 4 | 4 | 4 | a | | |

Gambar 1. Gambaran data yang digunakan dalam penelitian

*2. 2 Pengolahan Data*

Python telah terbukti memiliki tingkat permormansi yang baik pada segi *correct/incorrect instances*, *precision*, dan *recall* jika dibandingkan dengan menggunakan Weka[11]. Penelitian ini menggunakan library KNN dan Naive Bayes. Sebelum data training yang diproses, data tersebut dikonversi terlebih dahulu dalam format CSV. Selanjutnya data dalam format ini akan disimpan dan diproses dengn library numpy dan pandas. Library Scikit-learn adalah library utama yang digunakan untuk menjalankan algoritma *machine learning* beserta analisis datanya[12]. Library ini sangat berguna untuk para pengguna yang memiliki pemahaman yang kurang dalam pemrograman *machine learning*[13] karena hanya dengan kode-kode sederhana, para pengguna tersebut dapa melihat hasil serta analisis algoritma dengan lengkap. Sedangkan library matplotlib dapat digunakan untuk analisis dan pemodelan prediksi dengan diagram dan grafik.

Algoritma KNN dan Naive Bayes digunakan untuk pemrosesan data training dan membangun model prediksi agar dapat digunakan untuk memprediksi data baru dalam hal ini data mahasiswa angkatan 2015-2016. Dua algoritma ini dijalankan dengan library Scikit-learn.

*2.3 Algoritma KNN*

KNeighborsClassifier dari Scikit-learn digunakan dalam penelitian ini untuk mengimplementasikan proses pembelajaran knearest neighbors dari tiap tahap dalam data training, di mana k adalah jumlah *neighbor* berupa bilangan bulat yang telah ditentukan sebelumnya. Tahapan prosesnya meliputi :

a. Buka data training (dalam penelitian ini disimpan dalam bentuk csv)

b. Tentukan jumlah *neighbor* k

c. Untuk setiap data training:

i. Hitung Euclidean distance

$$d(p,q) = d(q,p) = \sqrt{(q_2 - p_1)^2 + (q_2 - p_1)^2 + \cdots + (q_n - p_n)^2}$$

$$Euclidean\ distance\ (d) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

ii. Tampilkan hasil perhitungan Euclidean distances dan urutkan

iii. Klasifikasikan *neighbor* terdekat berdasarkan nilai k (misal k=3 artinya jumlah *neighbor* terdekat adalah 3 atau 3 data yang hasil Euclidean distances terkecil tersebut dijadikan satu kelas)

*2.4 Algoritma Naive Bayes*

Pada library Scikitlearn, algoritma Naive bayes diterapkandengan fungsi GaussianNB yang di dalamnya terdapat perhitungan klasifikasi dengan Gaussian Naive Bayes. Perhitungan kemiripan atau kedekatan Gaussian ini dinyatakan dalam :

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Dengan parameter $\sigma_y$ dan $\mu_y$ diasumsikan dengan kemiripan atau kedekatan tertinggi[14].

## 3. HASIL DAN PEMBAHASAN

Hasil penerapan algoritma KNN untuk data training dapat dilihat pada Gambar 2 dan Gambar 3. Sedangkan hasil penerapan algoritma Naive Bayes dapat dilihat pada Gambar 4. Dari gambar tersebut kita dapat lihat bahwa library Scikit-learn juga menyediakan analisis akurasi (*accuracy*), presisi (*precision*), *recall*, *f1-score* dan *support*.

```
In [11]: X_train, X_test, y_train, y_test = train_test_split(X,y)

In [12]: print("Accuration = %0.2f" % accuracy_score(y_test, knn.predict(X_test)))
         print(classification_report(y_test, knn.predict(X_test)))
         print("Accuracy of 2NN: %.2f %%" %(100*accuracy_score(y_test,  knn.predict(X_test))))

         Accuration = 0.80
                       precision    recall  f1-score   support

                    a       0.88      1.00      0.93         7
                    b       0.00      0.00      0.00         1
                    c       1.00      0.50      0.67         2

            micro avg       0.80      0.80      0.80        10
            macro avg       0.62      0.50      0.53        10
         weighted avg       0.81      0.80      0.79        10

         Accuracy of 2NN: 80.00 %
```

Gambar 2 Hasil algoritma KNN k=2

```
In [11]: X_train, X_test, y_train, y_test = train_test_split(X,y)

In [12]: print("Accuration = %0.2f" % accuracy_score(y_test, knn.predict(X_test)))
         print(classification_report(y_test, knn.predict(X_test)))
         print("Accuracy of 1NN: %.2f %%" %(100*accuracy_score(y_test,  knn.predict(X_test))))

         Accuration = 1.00
                       precision    recall  f1-score   support

                    a       1.00      1.00      1.00         6
                    b       1.00      1.00      1.00         4

            micro avg       1.00      1.00      1.00        10
            macro avg       1.00      1.00      1.00        10
         weighted avg       1.00      1.00      1.00        10

         Accuracy of 1NN: 100.00 %
```

Ganbar 3 Hasil algoritma KNN k=1

```
In [12]: X_train, X_test, y_train, y_test = train_test_split(X,y)

In [13]: print("Accuration = %0.2f" % accuracy_score(y_test, gnb.predict(X_test)))
         print(classification_report(y_test, gnb.predict(X_test)))
         print("Accuracy of NaiveBayes: %.2f %%" %(100*accuracy_score(y_test, gnb.predict(X_test))))

         Accuration = 0.90
                       precision    recall  f1-score   support

                    a       1.00      0.80      0.89         5
                    b       0.80      1.00      0.89         4
                    c       1.00      1.00      1.00         1

            micro avg       0.90      0.90      0.90        10
            macro avg       0.93      0.93      0.93        10
         weighted avg       0.92      0.90      0.90        10

         Accuracy of NaiveBayes: 90.00 %
```

Gambar 4 Hasil Algoritma Naive Bayes

Dari 10 percobaan pengosongan secara acak dengan masing masing algoritma didapatkan nilai akurasi (*accuracy score*) yang dapat dilihat pada Tabel 2.

Tabel 2 Percobaan Pengosongan Acak

| Algoritma | Percobaan ke- | Nilai Akurasi |
|---|---|---|
| Naive Bayes | 1 | 80% |
| | 2 | 90% |
| | 3 | 100% |
| | 4 | 90% |
| | 5 | 100% |
| | 6 | 90% |
| | 7 | 100% |
| | 8 | 100% |
| | 9 | 80% |
| | 10 | 90% |
| KNN, K=2 | 1 | 90% |
| | 2 | 90% |
| | 3 | 70% |
| | 4 | 100% |
| | 5 | 80% |
| | 6 | 90% |
| | 7 | 70% |
| | 8 | 80% |
| | 9 | 90% |
| | 10 | 90% |

Rata-rata hasil akurasi dari percobaan yang dilakukan dapat dilihat di Tabel 3.

Tabel 3 Hasil Akurasi

| Algoritma | Nilai Akurasi |
|---|---|
| Naive Bayes | 92% |
| KNN, K=2 | 85% |

Dari hasil yang didapatkan, meskipun dengan data yang kurang, potensi dari kedua algoritma ini sangat baik untuk diterapkan ke dalam sistem deteksi dini putus kuliah di program studi Pendidikan Komputer. Meskipun begitu, perlu dilakukan penelitian lanjutan, dengan data yang lebih banyak dan lebih update, untuk memastikan konsistensi dan akurasi dari algoritma yang digunakan. Untuk penerapan prediksi potensi putus kuliah mahasiswa angkatan 2015-2016, dapat dilihat pada Gambar 5

```
In [10]: Tes = [[3,3,4,3,3,3,3.5,3,3.5,3,3.5,2.5,3.5,3.5,4,3.5,3.5,3,2.5,2,2,2,3.5,4,3,4,3.5,4,2.5,3.5,3,
         Y_pred = knn.predict(Tes)
         print("Hasil Prediksi : ", Y_pred)

         Hasil Prediksi :  ['b']
```

Gambar 5 Prediksi dengan data baru (hasil: status 2 “hampir putus kuliah”)

Setelah diujicobakan, sebagian data mahasiswa berada pada status potensi ‘hampir putus kuliah’. Hal ini berarti perlu diwaspadai dan dilakukan sebuah prosedur dan perlakuan baru agar mahasiswa dapat fokus untuk melanjukan kuliah dan skripsi.

Karena kurangnya data historis profil akademik mahasiswa, penelitian selanjutnya perlu dikembangkan dengan variabel lain yang lebih beragam.

## 4. KESIMPULAN

Penelitian ini telah menelaah penggunaan algoritma *machine learning* untuk deteksi mahasiswa berpotensi putus kuliah di Indonesia. Penerapannya fokus pada data akademik (nilai-nilai akademik) terutama nilai mata kuliah wajib. Penggunaan library KNN dan Naive Bayes dalam Python sangat mudah diaplikasikan dan menghasilkan akurasi yang cukup baik. Hasil menunjukkan nilai akurasi algoritma Naive Bayes lebih unggul daripada KNN dalam memprediksi status potensi putus kuliah mahasiswa. Namun, penelitian ini perlu ditingkatkan lagi dengan data mahasiswa yang lebih banyak dan beragam agar hasil akurasinyaa lebih meyakinkan dan konsistensinya tidak perlu ditanyakan lagi. Analisis yang dilakukan juga perlu diperdalam untuk mengisi kekurangan dan meningkatkan performansi dari algoritma yang digunakan.

## 5. SARAN

Penelitian selanjutnya perlu dilakukan kembali dengan data yang lebih banyak dan beragam. Perlu juga dilakukan perbandingan dengan algoritma lain untuk melihat perbandingan penggunaan variasi algoritma seperti SVM atau jaringan syaraf tiruan. Selain itu sebaiknya data profil non-akademik seperti latar belakang keluarga, status perkawinan, status pekerjaan dan motivasi belajar, perlu dipertimbangkan untuk menjadi faktor dan dimasukkan sebagai variabel independen dalam algoritma untuk meningkatkan akurasi.

## DAFTAR PUSTAKA

[1] Kemristekdikti, “Statistik Pendidikan Tinggi Indonesia 2017.” [Online]. Available: https://pddikti.ristekdikti.go.id/asset/data/publikasi/Statistik Pendidikan Tinggi Indonesia 2017.pdf. [Accessed: 07-Sep-2019].

[2] Kemristekdikti, “Statistik Pendidikan Tinggi Indonesi 2018.” [Online]. Available: https://pddikti.ristekdikti.go.id/asset/data/publikasi/Statistik Pendidikan Tinggi Indonesia 2018.pdf. [Accessed: 07-Sep-2019].

[3] X. Wu *et al.*, “Top 10 algorithms in data mining,” *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, Jan. 2008.

[4] M. Mustakim and G. Oktaviani, “Algoritma K-Nearest Neighbor Classification Sebagai Sistem Prediksi Predikat Prestasi Mahasiswa,” *J. Sains dan Teknol. Ind.*, vol. 13, no. 2, pp. 195–202, 2016.

[5] M. Wati, W. Indrawan, J. A. Widians, and N. Puspitasari, “Data mining for predicting students’ learning result,” in *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, 2017, pp. 1–4.

[6] I. Ognjanovic, D. Gasevic, and S. Dawson, “Using institutional data to predict student course selections in higher education,” *Internet High. Educ.*, vol. 29, pp. 49–62, Apr. 2016.

[7] A. M. Shahiri, W. Husain, and N. A. Rashid, “A Review on Predicting Student’s Performance Using Data Mining Techniques,” *Procedia Comput. Sci.*, vol. 72, pp. 414–422, Jan. 2015.

[8] I. A. A. Amra and A. Y. A. Maghari, “Students performance prediction using KNN and Naïve Bayesian,” in *2017 8th International Conference on Information Technology (ICIT)*, 2017, pp. 909–913.

[9] M. Mayilvaganan and D. Kalpanadevi, “Comparison of classification techniques for

predicting the performance of students academic environment," in *2014 International Conference on Communication and Network Technologies*, 2014, pp. 113–118.

[10] S. Taruna and M. Pandey, "An empirical analysis of classification techniques for predicting academic performance," in *2014 IEEE International Advance Computing Conference (IACC)*, 2014, pp. 523–528.

[11] J. Mitrpanont *et al.*, "A study on using Python vs Weka on dialysis data analysis," in *2017 2nd International Conference on Information Technology (INCIT)*, 2017, pp. 1–6.

[12] L. Buitinck *et al.*, "API design for machine learning software: experiences from the scikit-learn project," *arXiv Prepr. arXiv1309.0238*, 2013.

[13] G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller, "Scikit-learn," *GetMobile Mob. Comput. Commun.*, vol. 19, no. 1, pp. 29–33, Jun. 2015.

[14] "Naive Bayes." [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html. [Accessed: 06-Jul-2019].