PERCOBAAN IV

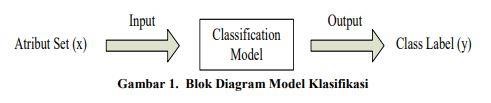
ALGORITMA *NAIVE BAYES*

1. TUJUAN
2. Praktikan mampu memahami algoritma *naïve bayes.*
3. Praktikan mampu mengimplementasikan algoritma *Naive Bayes.*

1. ALAT DAN BAHAN
2. Laptop
3. *Pycharm*
4. *Python*
5. Modul
6. *Microsoft Word*

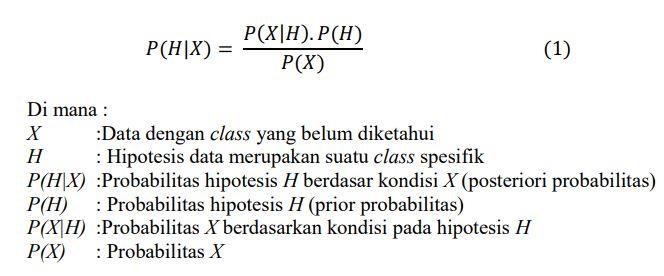
1. TEORI DASAR

*Naive Bayes* merupakan sebuahpengklasifikasianprobabilistiksederhana yang menghitung sekumpulanprobabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari *dataset* yang diberikan.Algoritma mengunakan teorema *Bayes* dan mengasumsikan semua atribut independen atau tidak saling ketergantungan yang diberikan oleh nilai pada variabel kelas.

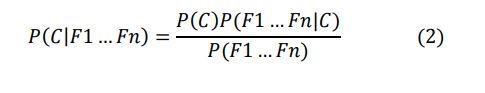


Definisi lain mengatakan *Naive Bayes* merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya. *Naive Bayes* didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai output. Dengan kata lain, diberikan nilai output, probabilitas mengamati secara bersama adalah produk dari probabilitas individu. Keuntungan penggunaan *Naive Bayes* adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (*Training* *Data*) yang kecil untuk menentukan estimasi paremeter yang diperlukan dalam proses pengklasifikasian. *Naive Bayes* sering bekerja jauh lebih baik dalam kebanyakan situasi dunia nyata yang kompleks dari pada yang diharapkan.

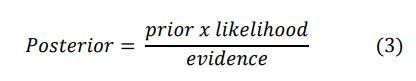
Persamaan dari teorema Bayes adalah



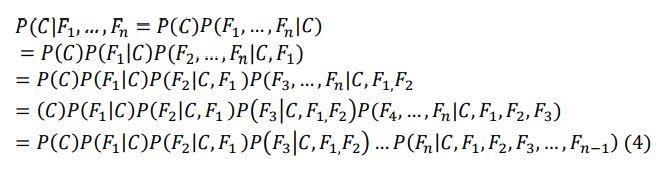
Untuk menjelaskan metode *Naive Bayes*, perlu diketahui bahwa proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut. Karena itu, metode *Naive Bayes* di atas disesuaikan sebagai berikut:



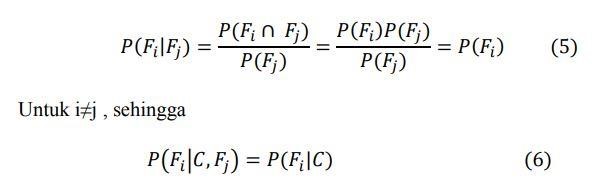
Di mana Variabel C merepresentasikan kelas, sementara variabel F1 ... Fn merepresentasikan karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi. Maka rumus tersebut menjelaskan bahwa peluang masuknya sampel karakteristik tertentu dalam kelas C (Posterior) adalah peluang munculnya kelas C (sebelum masuknya sampel tersebut, seringkali disebut *prior*), dikali dengan peluang kemunculan karakteristik, karakteristik sampel pada kelas C (disebut juga *likelihood*), dibagi dengan peluang kemunculan karakteristik-karakteristik sampel secara global (disebut juga *evidence*). Karena itu, rumus di atas dapat pula ditulis secara sederhana sebagai berikut:



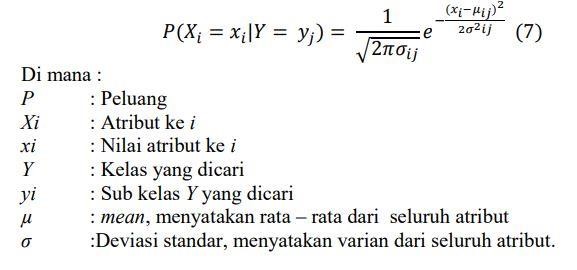
Nilai *Evidence* selalu tetap untuk setiap kelas pada satu sampel. Nilai dari posterior tersebut nantinya akan dibandingkan dengan nilai-nilai posterior kelas lainnya untuk menentukan ke kelas apa suatu sampel akan diklasifikasikan. Penjabaran lebih lanjut rumus *Bayes* tersebut dilakukan dengan menjabarkan  menggunakan aturan perkalian sebagai berikut:



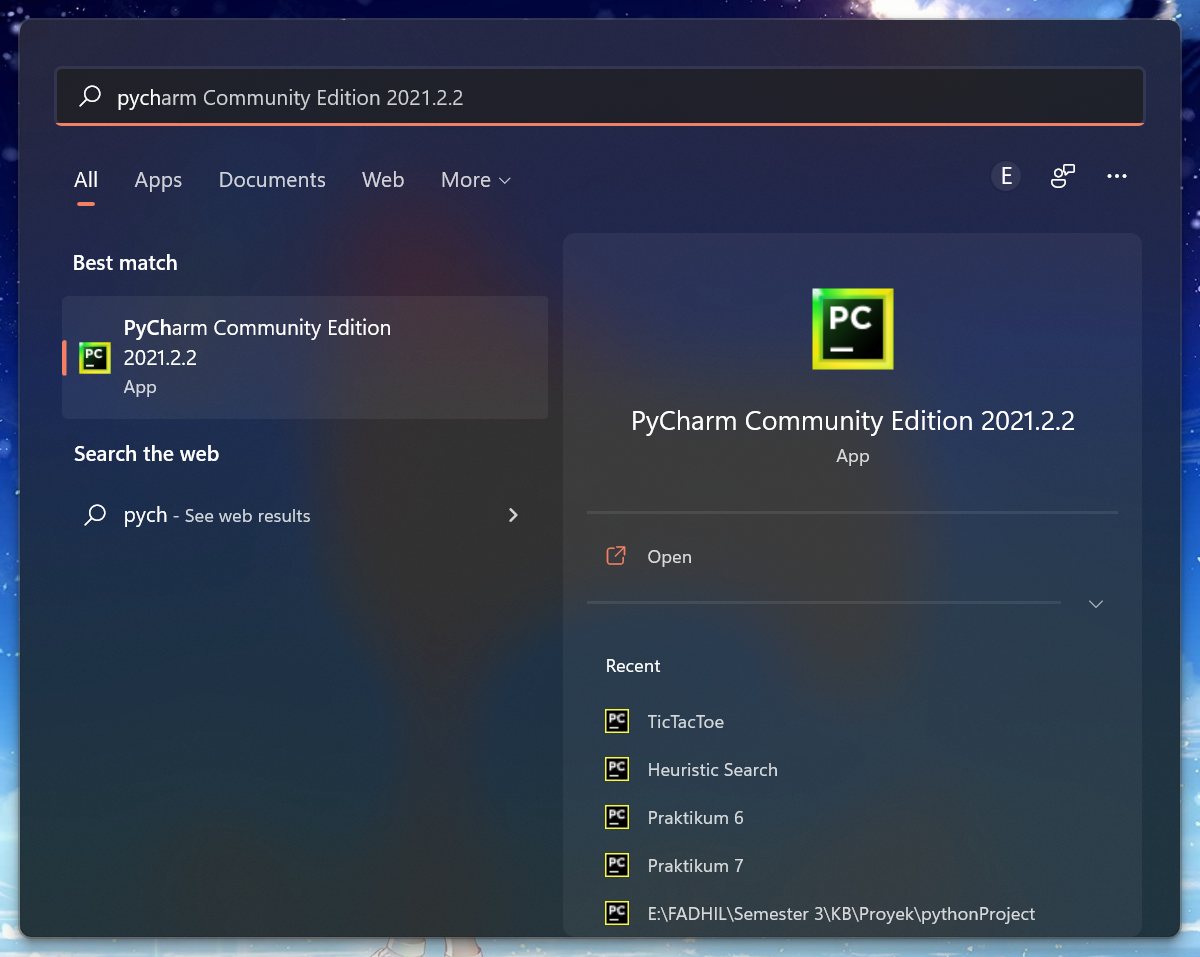
Dapat dilihat bahwa hasil penjabaran tersebut menyebabkan semakin banyak dan semakin kompleksnya faktor - faktor syarat yang mempengaruhi nilai probabilitas, yang hampir mustahil untuk dianalisa satu persatu. Akibatnya, perhitungan tersebut menjadi sulit untuk dilakukan. Di sinilah digunakan asumsi independensi yang sangat tinggi (naif), bahwa masing-masing petunjuk (F1,F2...Fn) saling bebas (independen) satu sama lain. Dengan asumsi tersebut, maka berlaku suatu kesamaan sebagai berikut:



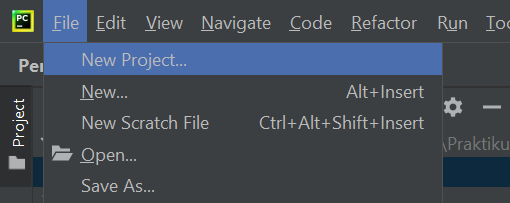
Persamaan di atas merupakan model dari teorema *Naive Bayes* yang selanjutnya akan digunakan dalam proses klasifikasi. Untuk klasifikasi dengan data kontinu digunakan rumus Densitas Gauss :



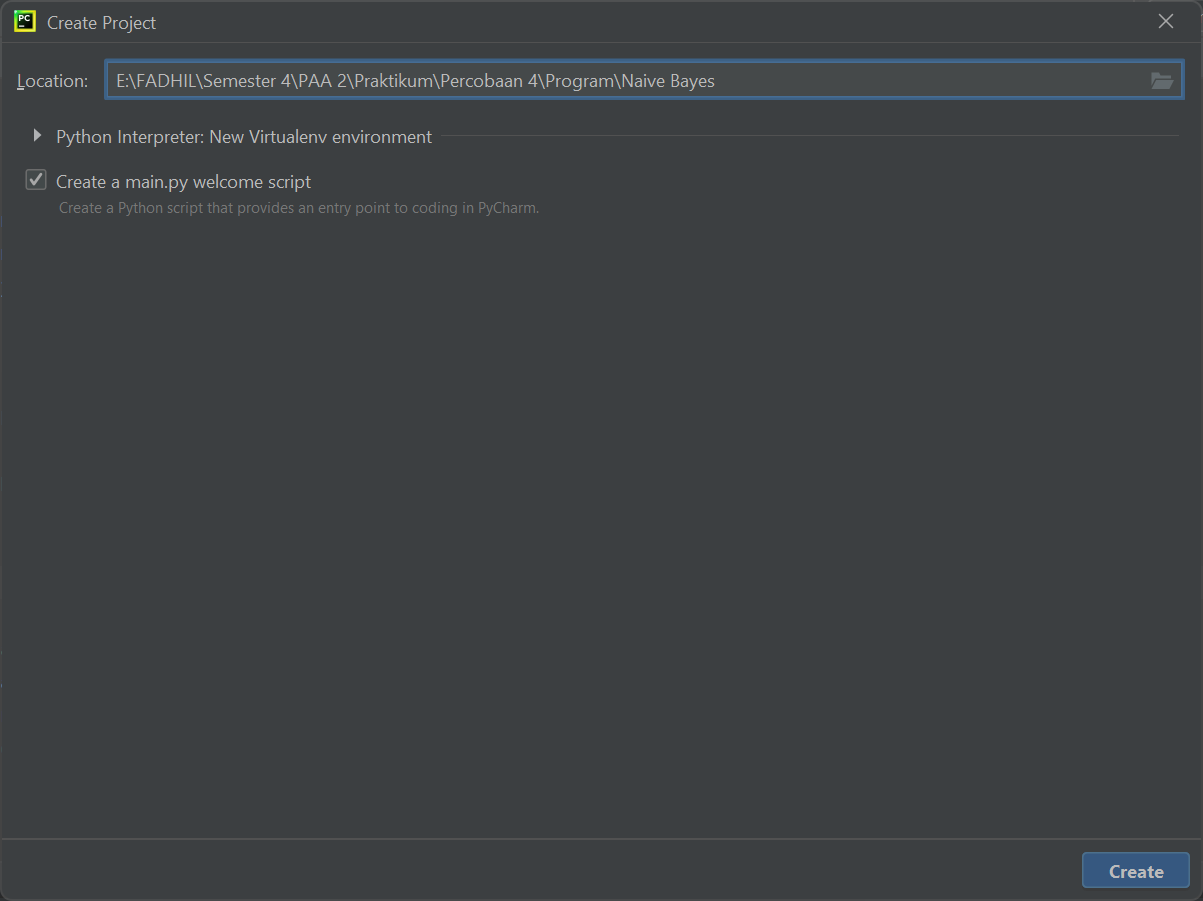
1. LANGKAH KERJA
2. Buka aplikasi *pycharm.*



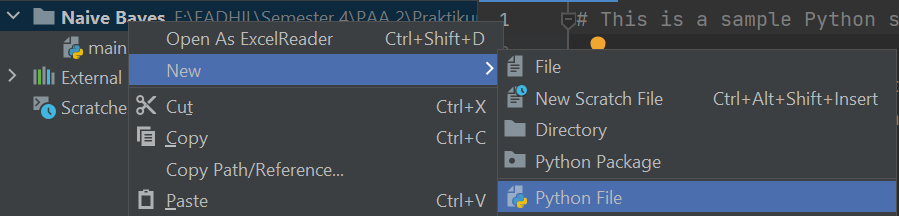
1. Klik *file, new project.*



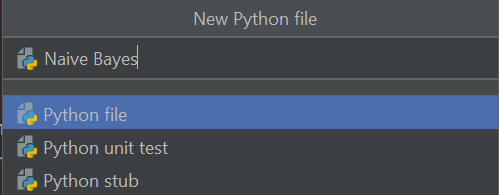
1. Tentukanlah letak *file project* dan beri nama *project* lalu klik *create*.



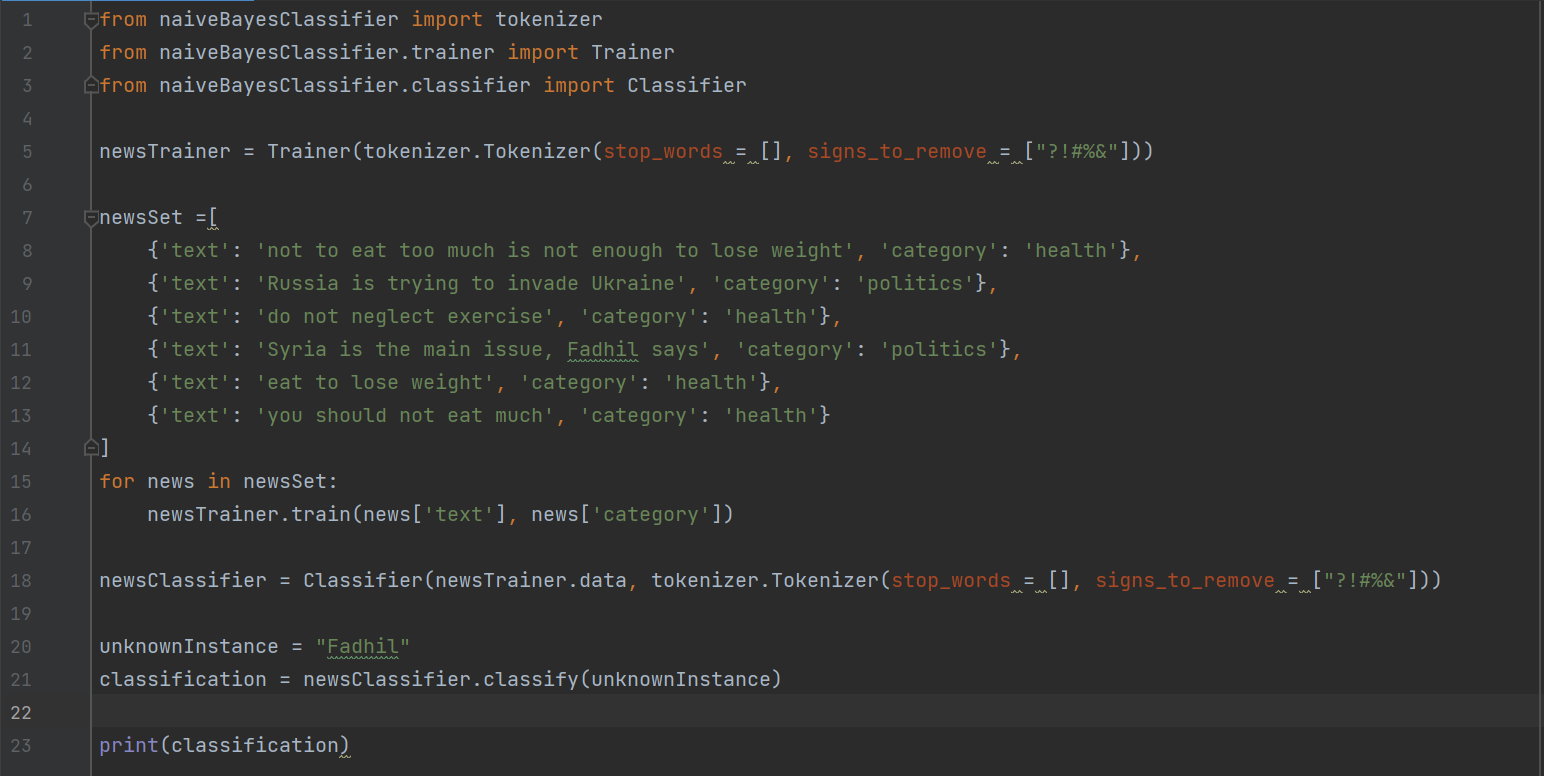
1. Klik kanan pada nama *file* yang telah di buat, kemudian pilih *new file*.



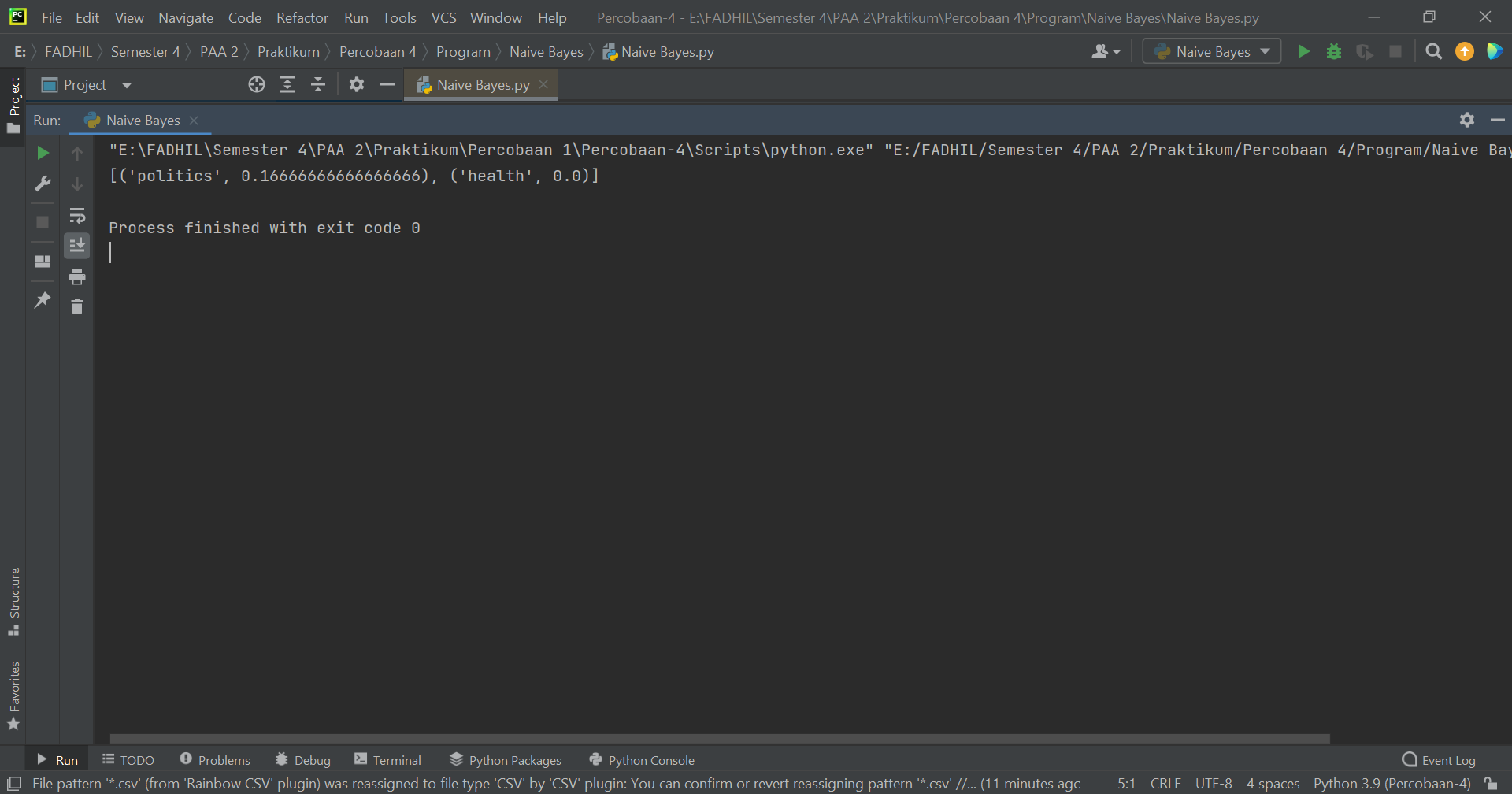
1. Tentukan nama *file project* yang di buat.

****

1. Masukkan kode program berikut.



V. HASIL PERCOBAAN



VI. ANALISIS

Pada percobaan diatas dapat dianalisis bahwa untuk mengimplementasi algoritma *naive bayes*, pertama kita mengimport *library naiveBayesClassifier*, yang berfungsi memprediksi probabilitas pada masa depan berdasarkan pengalaman pada masa sebelumnya,Pada *library* tersebut kita mengimport lagi fungsi *classifier dan trainer.* masukkan kode program untuk fungsi *trainer “newsTrainer = Trainer(tokenizer.Tokenizer(stop\_words = [], signs\_to\_remove = ["?!#%&"])),* pada *trainer* terdapat variabel(*Stop\_words = [], signs\_to\_remove = [“?!#%&”]*) yang dimana akan dilakukan penyaringan data sebelum atau sesudah pemrosesan bahasa alami(teks). Terdapat juga variabel “*newsSet =[{'text': 'Syria is the main issue, Fadhil says', 'category': 'politics'}]*”, sebagai data pelatihan yang terdiri atas *text* dan *category.*

Variabel “*for news in newsSet*:” untuk melakukan perulangan pada *newsSet*, “*newsTrainer.train[‘text’], news[‘category’])*” merupakan kondisi untuk memenuhi perulangan yang dilakukan pada *newsSet. “newsClassifier = Classifier(newsTrainer.data, tokenizer.Tokenizer(stop\_words = [], signs\_to\_remove = ["?!#%&"]))”* merupakan fungsi untuk mengelompokkan data pada data *trainer*. “*unknownInstance = “fadhil*” merupakan data yang ditentukan sebagai data yang kondisi atau letaknya tidak diketahui (*unknown*), “*classification = newsClassifier.classify(unknownInstance)*”, merupakan fungsi untuk melakukan pengklasifikasian data pada *unknownInstance. “Print(classification)”* fungsi ini untuk menampilkan data pada *classification* yang akan ditampilkan pada layar program.

VII. KESIMPULAN

Pada percobaan ini kita dapat mengetahui bahwa algoritma *naive bayes* sebuahpengklasifikasianprobabilistiksederhana yang menghitung sekumpulanprobabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari *dataset* yang diberikan. Algoritma ini akan melakukan pengklasifikasian data pada data *trainer.* Dari program yang telah kita lakukan, pengklasifikasian yang dilakukan hanya terdiri dari dua *category* yaitu *category health* dan *category politics.*

Data yang dilakukan pengklasifikasian adalah “(*unknownInstance = “fadhil”)”* yang mana fadhil berada pada data *trainer* dengan *category politics* yang mana hal tersebut akan menunjukkan nilai yang pada *politics* yaitu 0.16666.