

Machine Learning

Penerapan Algoritma *Naive Bayes* dan *Selection Sort* pada Penilaian Kuis di Aplikasi Pembelajaran Pemrograman Java dan PHP

Cindy Yovita, Khairuddin Nasution, Tasliyah Haramaini

Fakultas Teknik, Program Studi Teknik Informatika, Universitas Islam Sumatera Utara, Medan, Indonesia

INFORMASI ARTIKEL

Diterima Redaksi: 09 Juni 2023
Revisi Akhir: 31 Juli 2023
Diterbitkan Online: 01 Agustus 2023

KATA KUNCI

Teknologi Informasi; Pembelajaran Online;
Naive Bayes; *Selection Sort*

KORESPONDENSI

Phone: +6282267739198
E-mail: sindiyofita2000@gmail.com

A B S T R A K

Pada era digital saat ini, aplikasi pembelajaran berbasis komputer telah menjadi salah satu alat yang efektif dalam memfasilitasi proses pembelajaran. Dalam konteks pembelajaran pemrograman, aplikasi pembelajaran dapat memberikan pengalaman belajar yang interaktif dan mendalam bagi para pengguna.

Penelitian ini bertujuan untuk menerapkan algoritma *Naive Bayes* dan *Selection Sort* pada penilaian kuis di aplikasi pembelajaran pemrograman Java dan PHP. Algoritma *Naive Bayes* digunakan untuk melakukan analisis sentimen terhadap jawaban kuis yang diberikan oleh pengguna, sementara algoritma *Selection Sort* digunakan untuk mengurutkan dan menampilkan peringkat hasil kuis pengguna.

Metode penelitian yang digunakan adalah metode pengembangan perangkat lunak, yang meliputi analisis kebutuhan, perancangan sistem, implementasi, dan evaluasi. Pengembangan aplikasi menggunakan bahasa pemrograman Java dan PHP sebagai bahasa utama, dengan dukungan pengembangan menggunakan framework yang sesuai.

Pada tahap implementasi, aplikasi pembelajaran pemrograman Java dan PHP dikembangkan dengan memperhatikan desain yang telah dirancang sebelumnya. Algoritma *Naive Bayes* diterapkan untuk menganalisis sentimen jawaban kuis pengguna, sementara algoritma *Selection Sort* digunakan untuk mengurutkan dan menampilkan peringkat hasil kuis.

Hasil evaluasi menunjukkan bahwa penerapan algoritma *Naive Bayes* dan *Selection Sort* pada penilaian kuis di aplikasi pembelajaran pemrograman Java dan PHP memberikan hasil yang sesuai dengan harapan. Analisis sentimen dapat membantu mengidentifikasi jawaban kuis yang benar atau salah, sementara penggunaan *Selection Sort* membantu dalam menyajikan peringkat hasil kuis secara efisien.

Penelitian ini memberikan kontribusi pada pengembangan aplikasi pembelajaran pemrograman dengan memanfaatkan algoritma *Naive Bayes* dan *Selection Sort*. Hasil penelitian ini diharapkan dapat meningkatkan kualitas pengalaman belajar pengguna dalam memahami materi pemrograman Java dan PHP melalui kuis interaktif.

PENDAHULUAN

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (intruccion) atau pernyataan (statement) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output. Menurut [1] Aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi suatu pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu [2]. Seperti yang diketahui, pembelajaran pemrograman merupakan salah satu pembelajaran penting di Teknik Informatika. Universitas Islam Sumatera Utara adalah salah satu Universitas yang memiliki jurusan Teknik Informatika. Mahasiswa UISU diharuskan

untuk mengetahui bahasa pemrograman dari yang low level hingga high level. Mempelajari bahasa pemrograman diimbangi agar dapat mempermudah mahasiswa yang akan memilih skripsi pada peminatan pemrograman.

Bahasa pemrograman adalah perangkat lunak atau *software* yang dapat digunakan dalam proses pembuatan program yang melalui beberapa tahapan-tahapan penyelesaian masalah. Proses pemrograman komputer bukan saja sekedar menulis suatu urutan instruksi yang harus dikerjakan oleh komputer akan tetapi bertujuan untuk memecahkan suatu masalah serta membuat mudah pekerjaan pengguna komputer (*user*). Didalam membuat sebuah program komputer, tentu tidak terlepas dari sifat individu pemrogram (*Programmer*) [3]. Kurangnya pemahaman mahasiswa dalam mempelajari bahasa pemrograman dapat mempersulit mahasiswa untuk membuat suatu aplikasi. Banyak mahasiswa yang hanya berpatok kepada pembelajaran tatap muka, sehingga membuat pemahaman mereka kurang dalam pemrograman. Memperhitungkan persentase kemampuan mahasiswa dapat dilakukan dengan membuat statistik kemampuan.

Pendidikan dan pembelajaran merupakan faktor penting dalam pengembangan sumber daya manusia. Dalam era digital yang semakin berkembang, aplikasi pembelajaran berbasis komputer telah menjadi alat yang efektif dalam memfasilitasi proses pembelajaran di berbagai bidang, termasuk pembelajaran pemrograman. Pemrograman merupakan salah satu bidang yang membutuhkan pemahaman mendalam dan penerapan konsep-konsep yang tepat. Oleh karena itu, penggunaan aplikasi pembelajaran pemrograman dapat membantu para pembelajar untuk memperoleh pemahaman yang baik melalui metode pembelajaran yang interaktif dan terstruktur. Dalam konteks pembelajaran pemrograman, kuis sering digunakan sebagai alat evaluasi untuk mengukur pemahaman dan kemampuan para pembelajar. Namun, penilaian kuis yang dilakukan secara manual dapat memakan waktu dan menghasilkan kesalahan yang tidak diinginkan. Oleh karena itu, penerapan algoritma komputer dalam penilaian kuis dapat memberikan solusi yang lebih efisien dan akurat.

Dalam penelitian ini, penulis berfokus pada penerapan dua algoritma yaitu Algoritma *Naive Bayes* dan *Selection Sort* dalam penilaian kuis di aplikasi pembelajaran pemrograman Java dan PHP. Algoritma *Naive Bayes* digunakan untuk melakukan analisis sentimen terhadap jawaban kuis yang diberikan oleh pengguna, sementara algoritma *Selection Sort* digunakan untuk mengurutkan dan menampilkan peringkat hasil kuis pengguna. Algoritma *Naive Bayes* dipilih karena kemampuannya dalam melakukan klasifikasi berdasarkan kemungkinan terbaik. Dengan menganalisis sentimen jawaban kuis, algoritma ini dapat membantu mengidentifikasi jawaban yang benar atau salah. Sedangkan algoritma *Selection Sort* dipilih karena kemampuannya dalam mengurutkan data dengan efisien, yang akan digunakan untuk menampilkan peringkat hasil kuis pengguna.

Melalui penerapan algoritma *Naive Bayes* dan *Selection Sort* dalam penilaian kuis, diharapkan aplikasi pembelajaran pemrograman Java dan PHP dapat memberikan umpan balik yang lebih akurat dan efisien kepada pengguna. Selain itu, penggunaan algoritma komputer juga diharapkan dapat meningkatkan interaktivitas dan pengalaman belajar pengguna dalam memahami materi pemrograman. penulis akan menggunakan metode pengembangan perangkat lunak untuk mengembangkan aplikasi pembelajaran pemrograman Java dan PHP yang memanfaatkan algoritma *Naive Bayes* dan *Selection Sort*. Metode ini meliputi tahap analisis kebutuhan, perancangan sistem, implementasi, dan evaluasi. Hasil penelitian ini diharapkan dapat memberikan kontribusi positif dalam pengembangan aplikasi pembelajaran pemrograman yang lebih efektif dan efisien.

TINJAUAN PUSTAKA

Algoritma

Algoritma adalah urutan langkah-langkah logis penyelesaian yang disusun secara sistematis. Pertimbangan dalam memilih algoritma adalah algoritma memiliki nilai benar, memiliki efisiensi yang artinya algoritma dipakai karena dapat memberikan nilai benar dengan cepat. Pertimbangan dalam memilih algoritma adalah algoritma memiliki nilai benar, memiliki efisiensi yang artinya algoritma dipakai karena dapat memberikan nilai benar dengan cepat. [4]

Pengertian algoritma adalah prosedur komputasi yang mengambil beberapa nilai atau kumpulan nilai sebagai input kemudian di proses sebagai output sehingga algoritma merupakan urutan langkah komputasi yang mengubah input menjadi output [5].

Algoritma Naive Bayes

Algoritma *Naive Bayes* (NB) dapat digunakan untuk masalah klasifikasi biner dan multikelas. NB membuat dan menilai model dengan sangat cepat dan skala secara linear dalam jumlah prediksi dan baris. NB juga merupakan klasifikasi yang mempresetasikan setiap kelas objek berdasarkan kesimpulan atau rekapitulasi probabilistik dan menemukan kemungkinan besar kelas yang sesuai dengan tiap objek yang akan ditentukan kelasnya dari objek-objek uji yang ada berdasarkan atribut-atribut atau variabel yang telah ditentukan nilainya. Klasifikasi NB juga tergolong ke dalam pembelajaran yang terawasi (*supervised learning*) dimana untuk menentukan kelas objek uji berdasarkan pada data latihan dan setiap objek didata latihan telah diketahui kelasnya masing masing. NB mengasumsikan bahwa fitur fiturnya atau variabel atau atributnya adalah independen sesuai kelas yang ditentukan, artinya bahwa semua fitur hanya mempunyai kelas sebagai node asal atau node orang tua (Kononenko, 1990; Langley, 1992; Domingos, 1997; Mitchell, 1997). Penggunaan NB untuk klasifikasi dianggap penting karena beberapa hal, seperti (Wu. X., dll., 2007); Sangat mudah untuk dibangun karena tidak memerlukan skema untuk estimasi parameter iterative yang rumit dan metode ini dapat langsung diimplementasikan ke dalam jumlah data dalam skala yang sangat besar. Mudah untuk ditafsirkan sehingga pengguna yang kurang terampil dalam teknik klasifikasi dapat dengan mudah memahami hasil akhir yang diperoleh.

Namun demikian, walaupun klasifikasi NB mudah dibangun dan bekerja dengan baik, tetapi dalam beberapa domain NB bekerja kurang baik, hal ini disebabkan oleh performa NB yang membiarkan adanya korelasi atau tidak relevannya atribut atributnya. Jika dua atau lebih atribut yang memiliki korelasi sangat kuat, mereka menerima terlalu banyak bobot dalam keputusan akhir terhadap kelas mana yang akan ditempati oleh suatu objek. Klasifikasi NB didasarkan pada teorema bayes dengan asumsi adanya sifat independen antara atribut atau variabel yang ada, atau dengan kata lain setiap atribut, fitur atau variabel yang ada tidak saling bergantung antara satu dengan yang lain. Selain itu, dalam NB ini juga diasumsikan bahwa semua fitur atau variabel sama pentingnya, tetapi asumsi tersebut secara umum boleh jadi tidak benar, tetapi secara praktis algoritmanya memberikan hasil yang sangat baik. Sebelum membahas lebih jauh tentang metode NB maka ada beberapa hal yang perlu di-review, seperti tentang probabilitas bersyarat dan teorema bayes [6].

Misalkan X merupakan sampel data dengan label kelasnya yang belum diketahui, dan misalkan H adalah beberapa hipotesis sehingga sampel data X ada dalam kelas C yang spesifik. Probabilitas bersyarat atau probabilitas H didalam X adalah probabilitas interaksi H dan X dari probabilitas X, atau dengan notasi lain dapat ditulis $P(H|X)$ yang merupakan persentase banyaknya H didalam X, atau dapat ditulis:

$$P(H|X) = \left(\frac{P(H \cap X)}{P(X)} \right) \quad (1)$$

Oleh karena itu, untuk menentukan $P(H|X)$ yang merupakan persentase banyaknya H dalam X. $P(H|X)$ adalah peluang posterior yang merepresentasikan tingkat kepercayaan.

Rumus teorema bayes dapat dilihat seperti ini:

$$P(H|X) = \left(\frac{P(X|H) \times P(H)}{P(X)} \right) \quad (2)$$

Dengan:

H = Hipotesisnya

X = Kejadian yang merelasikannya ke hipotesis

$P(H)$ = Peluang hipotesis

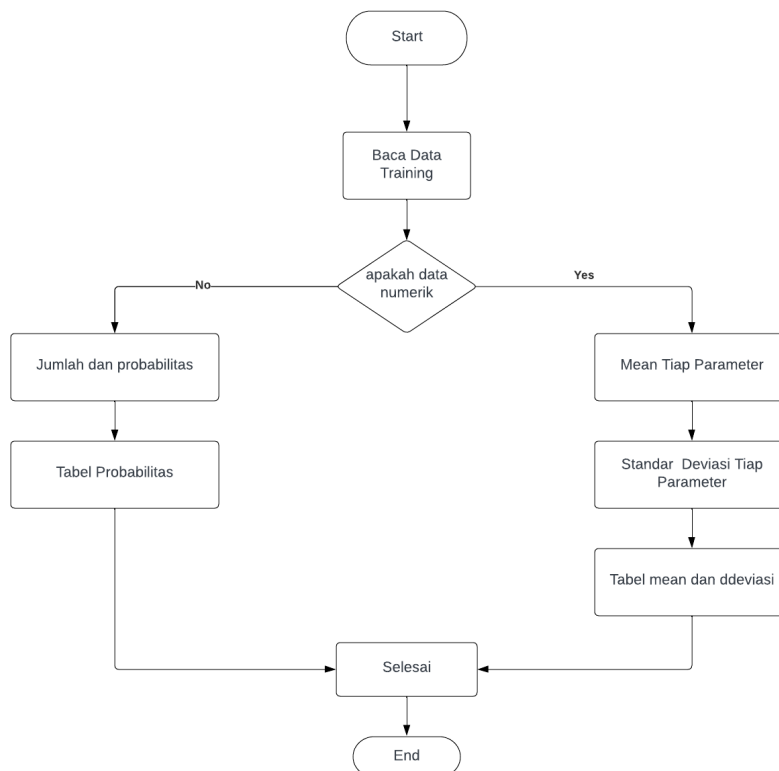
$P(X)$ = Peluang kejadian

$P(X|H)$ = Peluang banyaknya X didalam H

$P(H|X)$ = Peluang banyaknya H didalam X

Data yang akan digunakan dalam penggunaan algoritma ini adalah data nilai mahasiswa yang sudah mengikuti tes kuis.

Flowchart Algoritma Naive Bayes



Gambar 1. Flowchart Algoritma Naive Bayes

Pada gambar 1 merupakan contoh bagaimana cara kerja algoritma *naive bayes*. Algoritma ini dimulai dengan membaca data training yang dimasukkan, kemudian algoritma ini akan melihat apakah data yang dimasukkan merupakan data numerik atau tidak. Jika data yang dimasukkan merupakan bilangan numerik, maka tahap awal yang dilakukan adalah menghitung nilai rata-rata dari setiap parameter, lalu menghitung standar deviasi tiap parameter dan menampilkan tabel rata-rata dan deviasi. Tetapi, jika data yang dimasukkan bukan bilangan numerik, maka algoritma ini akan menghitung jumlah probabilitas dan setelah itu menampilkan tabel probabilitas.

Selection Sort

Selection Sort merupakan salah satu algoritma pengurutan yang sederhana. Ide dasarnya adalah melakukan beberapa kali pass untuk melakukan penyeleksian elemen struktur data. Untuk sorting *ascending* (menaik) elemen yang paling kecil di antara elemen-elemen yang belum urut, disimpan indeksinya, kemudian dilakukan pertukaran nilai elemen dengan indeks yang disimpan tersebut dengan elemen yang paling depan yang belum urut. Sebaliknya, untuk sorting *descending* (menurun) elemen yang paling besar yang disimpan indeksinya kemudian ditukar. *Selection Sort* diakui karena kesederhanaan algoritmanya dan performanya lebih bagus dari pada algoritma lain yang lebih rumit dalam situasi tertentu [7].

Metode sorting dibahas dalam penelitian yang dilakukan oleh Sitepu yang bertujuan untuk membandingkan antara metode Bubble Sort dengan Selection Sort. Algoritma Bubble Sort dan Selection Sort menggunakan Array List multidimensi mampu mengurutkan data multi kriteria yang memiliki lebih dari satu kriteria prioritas. Jumlah iterasi Selection Sort pada setiap pengujian data lebih sedikit dibandingkan dengan Bubble Sort, dan waktu eksekusi Selection Sort pada setiap pengujian data juga lebih cepat jika dibandingkan dengan waktu eksekusi Bubble Sort [7].

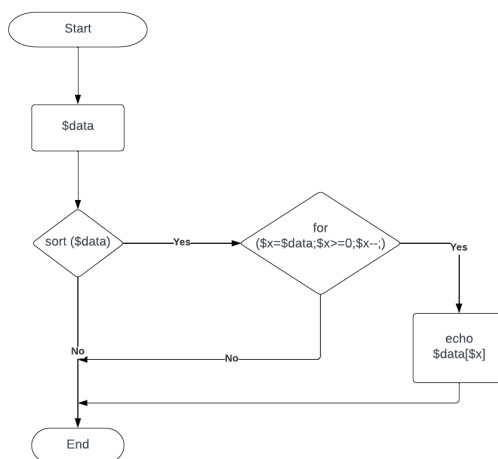
Penelitian oleh Retnoningsih juga melakukan perbandingan metode sorting antara Insertion Sort dan Bubble Sort untuk mengurutkan data berupa angka. Kedua metode tersebut dibandingkan dengan cara mengukur waktu yang dibutuhkan untuk melakukan pengurutan pada data acak. Hasil yang didapat adalah metode insertion sort lebih unggul pada jumlah data yang sedikit yaitu pada sejumlah sepuluh elemen data membutuhkan waktu 0,003 detik, lima puluh elemen data

membutuhkan waktu 0,013 detik, sedangkan metode selection sort lebih unggul pada jumlah data yang lebih banyak yaitu pada sejumlah lima ratus elemen data membutuhkan waktu 1,083 detik [8].

Penelitian ketiga tentang sorting dilakukan oleh Saputro & Khasanah yang membandingkan antara metode Selection Sort dan Bubble Sort untuk pengurutan data angka. Penelitian ini berhasil melakukan proses sorting terhadap tiga kelompok bilangan, kelompok pertama rentang 1 s/d 100, kelompok kedua rentang 100 s/d 1000, dan kelompok ketiga rentang 1000 s/d 1000000. Dari hasil analisa waktu yang telah dilakukan menunjukkan bahwa metode Selection Sort membutuhkan waktu yang lebih singkat dibandingkan dengan metode Bubble Sort dimana Selection Sort membutuhkan rata-rata waktu sekitar 0,031 detik, sedangkan metode Bubble Sort membutuhkan rata-rata waktu sekitar 0,078 detik [9].

Selection Sort adalah suatu metode pengurutan yang membandingkan elemen yang sekarang dengan elemen berikutnya sampai ke elemen yang terakhir. Jika ditemukan elemen lain yang lebih kecil dari elemen sekarang maka dicatat posisinya dan langsung ditukar. Metode selection sort adalah melakukan pemilihan dari suatu nilai yang terkecil dan kemudian menukarnya dengan elemen paling awal, lalu membandingkan dengan elemen yang sekarang dengan elemen berikutnya sampai dengan elemen terakhir, perbandingan dilakukan terus sampai tidak ada lagi pertukaran data [10][11].

Flowchart Selection Sort



Gambar 2. Flowchart Selection Sort

Pada gambar 2 merupakan contoh rangkaian cara kerja *selection sort*, dari data *flowchart* diatas akan menghasilkan sebuah *sorting* dengan keadaan *descending* (menaik) ke *ascending* (menurun). *Flowchart* tersebut dimulai dengan memperkenalkan variabel data yang lalu variabel data tersebut akan dimulai pengurutan dengan fungsi *sort*. Jika keadaan tersebut benar akan diurutkan, maka proses akan otomatis melanjutkan proses sorting dengan keadaan untuk variabel x = variabel data, dimana variabel x tidak sama dengan 0 dan perulangan dilakukan secara menurun. Maka hasil yang akan keluar nantinya adalah penurunan bilangan dari yang terbesar ke yang terkecil.

METODOLOGI

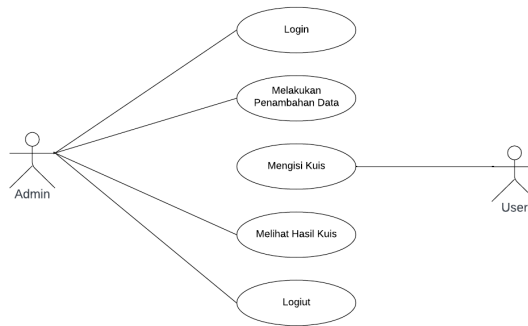
Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan pada penelitian ini, yaitu:

1. Observasi
Penelitian ini dilakukan dimana peneliti meninjau langsung proses pembelajaran dan pengisian quis ditempat.
2. Studi literatur
Studi literatur atau studi kepustakaan dilakukan dengan mengambil data dari buku, jurnal ataupun artikel mengenai teori dan langkah-langkah dalam pembuatan aplikasi. Sehingga dapat menggunakannya sebagai dasar landasan teori dan perancangan pembuatan aplikasi sistem pendukung keputusan

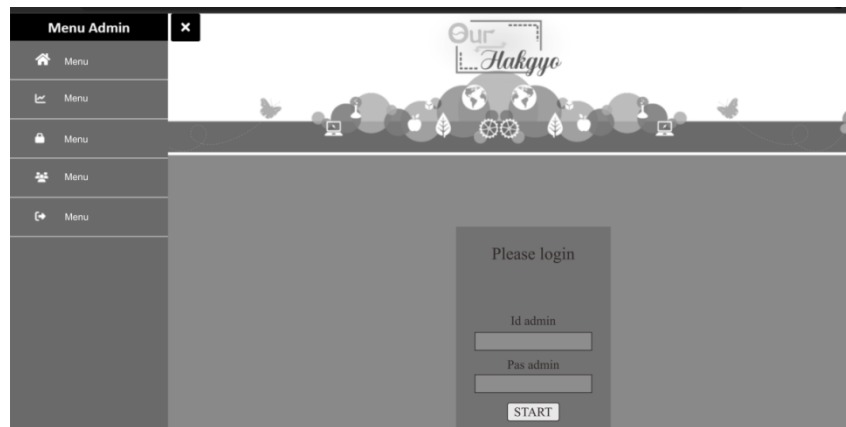
Perancangan Sistem

Use Case Diagram



Gambar 3. Use Case Diagram

Pada gambar 3 merupakan use case diagram, dimana admin akan melakukan *login* terlebih dahulu untuk dapat melakukan aktivitas didalam menu admin. Didalam menu admin, admin dapat melakukan penambahan data admin dan melihat hasil kuis yang sudah diisi oleh *user*. Sedangkan *user* dapat mengisi kuis yang ada.



Gambar 4. Perancangan Menu Login

Pada gambar 4, merupakan perancangan aplikasi pada menu login. Pada menu ini admin akan diminta untuk memasukkan username serta password yang admin sudah miliki.



Gambar 5. Perancangan Menu Admin

Pada gambar 5 adalah sebuah perancangan untuk halaman awal admin. Pada halaman ini akan ada 3 sub menu yang dapat admin kelola, yaitu: sub menu dashboard, data admin, dan data mahasiswa.



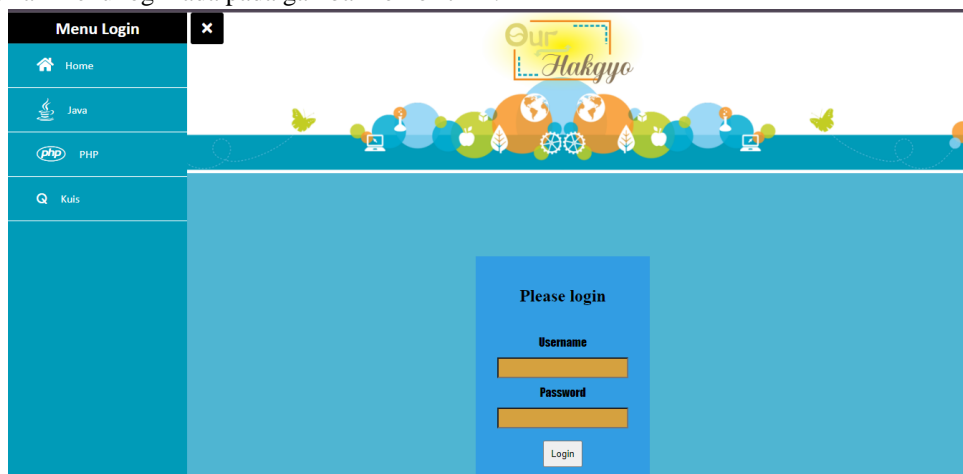
Gambar 6. Perancangan Menu Kuis

Gambar 6 adalah sebuah perancangan untuk menu Kuis. Didalam menu ini nantinya user akan mengerjakan kuis setelah user memahami bahan ajar yang ada.

HASIL DAN PEMBAHASAN

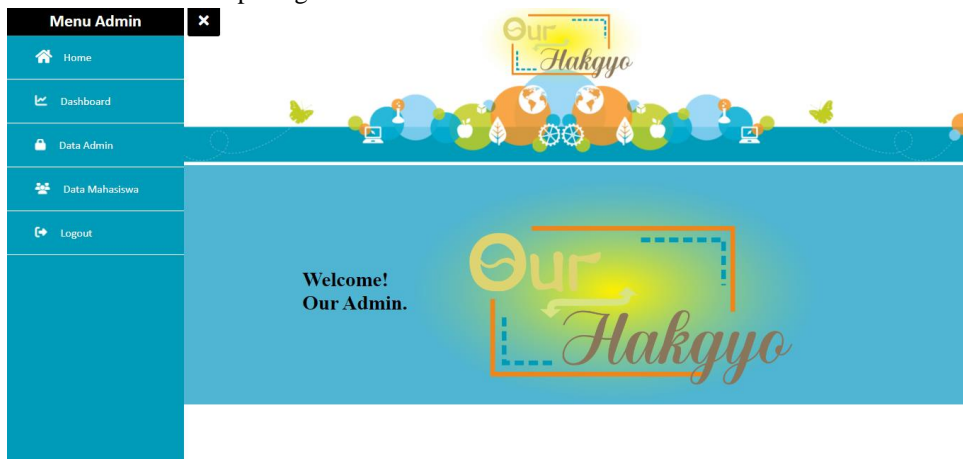
Implementasi

Tampilan halaman menu login ada pada gambar nomor 7 ini.



Gambar 7 Halaman Menu Login

Tampilan halaman menu admin ada pada gambar 8 ini



Gambar 8 Tampilan Halaman Menu Admin

Tampilan halaman menu kuis ada pada gambar 9 dibawah ini.



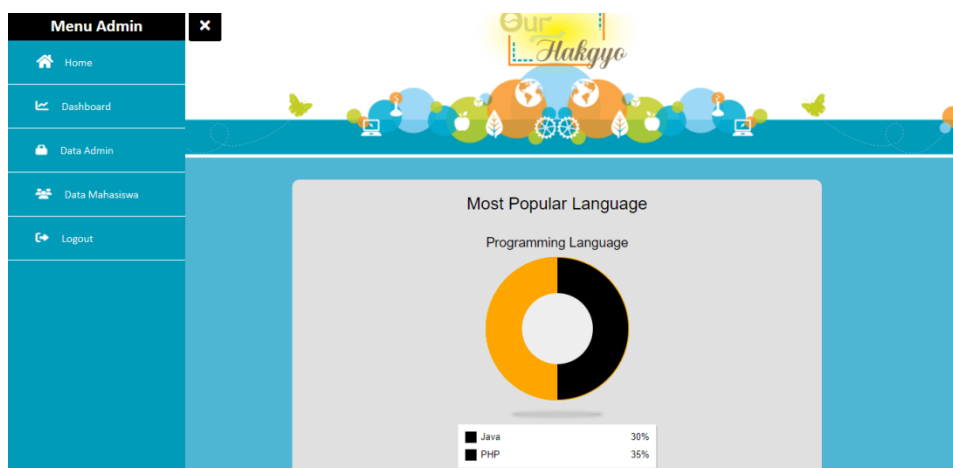
Gambar 9 Tampilan Halaman Menu Kuis

Tampilan hasil *sortig* nilai mahasiswa dengan menggunakan algoritma *Selection Sort* ada pada gambar 10 dibawah ini.

No	Nama	Tahun Akademik	Nilai
1	Aulia Fasya Ardana	2021	40
2	Nadhira Lubis	2020	40
3	Muhammad Fauzan Rifiqi	2021	40
4	Indah Trivati	2020	30
5	Sholly Hafidzon	2020	30
6	Hanafi Batubara	2021	30
7	Amanda Rengganis	2021	30

Gambar 10 Tampilan Mneu Hasil Kuis

Tampilan *chart* kemampuan yang menghitung jumlah persentase hasil kuis java dan PHP mahasiswa dengan algoritma *naïve bayes* ada pada gambar 11 dibawah ini.



Gambar 11 Tampilan Halaman *Chart* Kemampuan

KESIMPULAN DAN SARAN

Hasil dari *chart* kemampuan ini adalah mahasiswa lebih memahami bahasa pemrograman PHP dibandingkan dengan bahasa pemrograman Java. Penggunaan algoritma *Selection Sort* pada aplikasi ini adalah untuk mengurutkan nilai kuis mahasiswa dari yang terbesar ke yang terkecil. Hasil dari algoritma *naïve bayes* kurang efektif untuk aplikasi bahan ajar ini dikarenakan data yang ada lebih sedikit. Maka dari itu, untuk pengembangan aplikasi ini diharapkan dapat menggunakan algoritma lain yang lebih baik agar hasil dari aplikasi ini dapat maksimal.

DAFTAR PUSTAKA

- [1] Jogiyanto, H. M. (1999), Analisis dan Desain Sistem Informasi, Yogyakarta: Andi.
- [2] Noviansyah. Eka, "Aplikasi Website Museum Nasional Menggunakan Macromedia Dreamweaver MX", Jakarta : STIK, 2008.
- [3] Indrajani, 2007, Pemrograman Berbasis Objek dengan Bahasa Java, PT. Alex Media Komputindo, Jakarta.
- [4] Isroqmi, Asnurul. 2017. Kemampuan Mahasiswa Memahami Logika Pemrograman Komputer Melalui Algoritma. Jurnal Pendidikan Matematika. Vol. 2, No.2.
- [5] Cormen, Thomas H., 2009, Introduction to Algorithms, The MIT Press, Cambridge.
- [6] Arhami, M., & Nasir, M. (2020). Data Mining Algoritma dan Implementasi. Yogyakarta: ANDI.
- [7] Sitepu, R. R., Yusman, M., & Febriansyah, F. E. (2017). Implementasi Algoritma Bubble Sort Dan Selection Sort Menggunakan Arraylist Multidimensi Pada Pengurutan Data Multi Prioritas. Jurnal Komputasi, 5(1).
- [8] Retnoningsih, E. (2018). Algoritma Pengurutan Data (Sorting) Dengan Metode Insertion Sort dan Selection Sort. Information Management For Educators And Professionals: Journal of Information Management, 3(1), 95–106.
- [9] Saputro, F. E., & Khasanah, F. N. (2018). Teknik Selection Sort dan Bubble Sort Menggunakan Borland C++. Jurnal Mahasiswa Bina Insani, 2(2), 136–145.
- [10] Lasriana, Lasriana, and Aris Gunaryati. "SISTEM INFORMASI APOTEK BERBASIS WEB MENGGUNAKAN ALGORITMA SEQUENTIAL SEARCH DAN SELECTION SORT." JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika) 7.2 (2022): 392-401.
- [11] Pelayananpublik," pengertian php menurut para ahli, <https://pelayananpublik.id/2022/09/09/pengertian-phpmenurut-para-ahli-terlengkap>, 9 sep 2022.