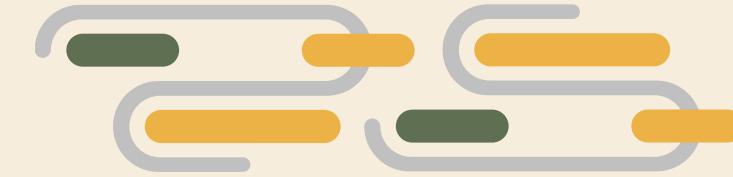


PENERAPAN ALGORITMA DECISION TREE UNTUK REKOMENDASI JURUSAN KULIAH DI FMIPA UNESA BERDASARKAN MINAT

Struktur Data dan Algoritma

Kelompok 8



KELOMPOK 8

Anggota Kelompok

- Tara Tabriza Rachman (24031554107)
- Bilqis Fadiyah Nisrina (24031554216)



DAFTAR ISI

 Universitas Negeri Surabaya

01

Judul Proyek

02

Decision Tree

03

Kode

04

Tampilan UI

MENGAPA JUDUL INI DIPILIH?

Judul “Menentukan Jurusan di FMIPA UNESA Berdasarkan Minat” dipilih karena proyek ini bertujuan membangun sistem berbasis Decision Tree yang dapat membantu calon mahasiswa dalam menentukan jurusan yang sesuai berdasarkan data minat dan akademik. Judul ini mencerminkan metode yang digunakan sekaligus konteks penerapannya secara spesifik di lingkungan FMIPA UNESA, sehingga lebih relevan, aplikatif, dan mudah dipahami.



APA ITU DECISION TREE?

Decision Tree adalah metode pengambilan keputusan yang berbentuk seperti pohon bercabang. Dalam konteks proyek ini, setiap node pada pohon merepresentasikan pertanyaan terkait minat siswa terhadap bidang-bidang yang ada di FMIPA, seperti Matematika, Biologi, Fisika, Kimia, Sains Data, Sains Aktuaria, atau Kecerdasan Artificial. Jawaban dari setiap pertanyaan akan menentukan cabang yang diikuti, hingga akhirnya mencapai node akhir yang berisi rekomendasi jurusan di lingkungan FMIPA. Metode ini dipilih karena mampu menyusun alur pengambilan keputusan secara sistematis dan mudah dipahami, terutama ketika hanya menggunakan satu jenis data, yaitu minat siswa.

KODE

```
import tkinter as tk
from PIL import Image, ImageTk
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder

# Baca dataset dari Excel
df = pd.read_excel("jurusan.xlsx")

# Label encoding untuk kolom kategori
le_minat = LabelEncoder()
le_subminat = LabelEncoder()
le_jurusan = LabelEncoder()

# Encode kolom
df['Minat_enc'] = le_minat.fit_transform(df['Minat'])
df['MinatSpesifik_enc'] = le_subminat.fit_transform(df['Minat Spesifik'])
df['Jurusan_enc'] = le_jurusan.fit_transform(df['Jurusan'])

# Model training
X = df[['Minat_enc', 'MinatSpesifik_enc']]
y = df['Jurusan_enc']
model = DecisionTreeClassifier()
model.fit(X, y)

# Detail jurusan
grouped = df.groupby("Jurusan").agg({
    "Deskripsi": lambda x: "\n\n".join(x.unique()),
    "Prospek Kerja": lambda x: "\n\n".join(x.unique()),
    "Akreditasi": lambda x: ", ".join(x.unique())})
jurusan_detail = grouped.to_dict(orient="index")
```

tkinter : Library untuk membuat antarmuka grafis (GUI)

PIL : untuk memproses gambar

pandas : untuk membaca dan memproses data dari file excel

sklearn : Sebagai library Machine Learning (ML) untuk Decision Tree

LabelEncoder : Untuk mengubah teks (kategori) menjadi angka agar bisa di proses ML

df : untuk membaca data “jurusan” dari file excel

le : Membuat objek encoder baru untuk merubah teks

fit_transform : Mempelajari dan mengubah data teks menjadi angka

_enc : Sebagai tempat menyimpan hasil angka

x : Data input, **y** : data target

model.fit : Model belajar untuk menebak jurusan berdasarkan minat yang dipilih

grouped : Menggabungkan deskripsi, prospek dan akreditas tiap jurusan

KODE

```
class JurusanDecisionTreeApp:  
    def __init__(self, root):  
        self.root = root  
        self.root.title("Rekomendasi Jurusan Berdasarkan Minat")  
        self.root.state('zoomed')  
        self.screen_width = self.root.winfo_screenwidth()  
        self.screen_height = self.root.winfo_screenheight()  
  
        self.minat = None  
        self.minat_spesifik = None  
        self.pred_jurusan = None  
  
        # ===== Halaman Awal =====  
        self.frame_awal = tk.Frame(root, width=self.screen_width, height=self.screen_height)  
        self.frame_awal.pack_propagate(False)  
  
        bg_image = Image.open("Merah Kuning Profesional Zoom Latar Belakang.png")  
        bg_image = bg_image.resize((self.screen_width, self.screen_height), Image.Resampling.LANCZOS)  
        self.bg_photo = ImageTk.PhotoImage(bg_image)  
  
        self.bg_label = tk.Label(self.frame_awal, image=self.bg_photo)  
        self.bg_label.place(x=0, y=0, relwidth=1, relheight=1)  
  
        self.start_button = tk.Button(  
            self.frame_awal, text="Mencari Jurusan", font=("Arial", 22, "bold"),  
            bg="white", fg="darkred", padx=30, pady=10, command=self.start_questions  
        )  
        self.start_button.place(relx=0.5, rely=0.5, anchor="center")  
  
        self.frame_awal.pack()  
        self.label = tk.Label(root, text="", font=("Arial", 16), wraplength=800)  
        self.options_frame = tk.Frame(root)  
  
    def start_questions(self):  
        self.frame_awal.pack_forget()  
        self.frame_pertanyaan = tk.Frame(self.root, width=self.screen_width, height=self.screen_height)  
        self.frame_pertanyaan.pack_propagate(False)
```

Terdapat kelas utama **JurusanDecisionTreeApp** yang akan mengelola seluruh aplikasi

self.root : window utama dari aplikasi, lalu menyimpan ukuran layar dan memulai aplikasi dalam fullscreen

Menyimpan pilihan ataupun hasil di **self minat, minat_spesifik, pred_jurusan** dan **jurusan_detail**

Membuat Frame untuk tampilan awal dengan background gambar yang di-resize sesuai layar

start_button : tombol untuk memulai proses pencarian jurusan dan ketika di klik akan pindah ke halaman **frame_pertanyaan**

start_questions : Frame pertanyaan yang berisi tombol pilihan minat (options_frame) dan memuat background baru

pack_forget untuk menghilangkan frame awal dan menampilkan **frame_pertanyaan**

KODE

```
self.frame_pertanyaan.pack()

bg_image2 = Image.open("Merah Kuning Profesional Zoom Latar Belakang (1).png")
bg_image2 = bg_image2.resize((self.screen_width, self.screen_height), Image.Resampling.LANCZOS)
self.bg_photo2 = ImageTk.PhotoImage(bg_image2)

self.bg_label2 = tk.Label(self.frame_pertanyaan, image=self.bg_photo2)
self.bg_label2.place(x=0, y=0, relwidth=1, relheight=1)

self.options_frame = tk.Frame(self.frame_pertanyaan)
self.options_frame.place(relx=0.5, rely=0.35, anchor="n")

self.show_minat_options()

def clear_options(self):
    for widget in self.options_frame.winfo_children():
        widget.destroy()

def show_minat_options(self):
    self.clear_options()
    minat_list = sorted(df["Minat"].unique())
    for m in minat_list:
        btn = tk.Button(self.options_frame, text=m, width=50, font=("Arial", 16), command=lambda m=m: self.select_minat(m))
        btn.pack(pady=5)

def select_minat(self, minat):
    self.minat = minat
    self.show_minat_spesifik_options()

def show_minat_spesifik_options(self):
    self.frame_pertanyaan.pack_forget()
    self.frame_minat_spesifik = tk.Frame(self.root, width=self.screen_width, height=self.screen_height)
    self.frame_minat_spesifik.pack_propagate(False)
    self.frame_minat_spesifik.pack()
```

clear_options : Menghapus seluruh tombol didalam **options_frame**

show_minat_options : menampilkan semua minat yang ada di data excel sebagai tombol (button) dan ketika tombol di klik maka akan memanggil **self.select_minat(minat)**

select_minat : Akan menyimpan minat yang dipilih pengguna dan menampilkan ke pertanyaan subminat selanjutnya yaitu frame

show_minat_spesifik_options

show_minat_spesifik_options: Memberikan frame pertanyaan baru yang menampilkan Minat Spesifik (subminat) berdasarkan pilihan minat sebelumnya sesuai dengan data yang ada di Excel

pack_propagate(False) : Mencegah ukuran frame menyesuaikan otomatis dengan isi didalamnya

KODE

```
bg_image_sub = Image.open("Merah Kuning Profesional Zoom Latar Belakang (6).png")
bg_image_sub = bg_image_sub.resize((self.screen_width, self.screen_height), Image.Resampling.LANCZOS)
self.bg_photo_sub = ImageTk.PhotoImage(bg_image_sub)

self.bg_label_sub = tk.Label(self.frame_minat_spesifik, image=self.bg_photo_sub)
self.bg_label_sub.place(x=0, y=0, relwidth=1, relheight=1)

self.options_frame = tk.Frame(self.frame_minat_spesifik)
self.options_frame.place(relx=0.5, rely=0.4, anchor="n")

sub_list = sorted(df[df["Minat"] == self.minat]["Minat Spesifik"].unique())
for s in sub_list:
    btn = tk.Button(self.options_frame, text=s, width=50, font=("Arial", 16),
                    command=lambda s=s: self.select_minat_spesifik(s))
    btn.pack(pady=5)

def select_minat_spesifik(self, minat_spesifik):
    self.minat_spesifik = minat_spesifik
    self.predict_jurusan()

def predict_jurusan(self):
    minat_enc = le_minat.transform([self.minat])[0]
    subminat_enc = le_subminat.transform([self.minat_spesifik])[0]
    X_input = pd.DataFrame([[minat_enc, subminat_enc]], columns=['Minat_enc', 'MinatSpesifik_enc'])
    pred_encoded = model.predict(X_input)[0]
    self.pred_jurusan = le_jurusan.inverse_transform([pred_encoded])[0]
    self.show_result()

def show_result(self):
    self.frame_minat_spesifik.pack_forget()
    self.frame_result = tk.Frame(self.root, width=self.screen_width, height=self.screen_height)
    self.frame_result.pack_propagate(False)
    self.frame_result.pack()
```

sub_list = sorted : Mengambil daftar minat ataupun minat spesifik sesuai dengan yang dipilih lalu diurutkan daftarnya

select_minat_spesifik : Menyimpan subminat yang telah dipilih dan menampilkan frame **predict_jurusan** berdasarkan pilihan sebelumnya yang telah di cocokkan dengan data “Jurusan” Excel

predict_jurusan : Data yang dipilih pengguna dikirim ke model ML lalu diprediksi jurusan yang cocok berdasarkan minat yang telah dipilih

show_result : Menampilkan jurusan yang direkomendasikan dan klik nama jurusan untuk melihat detailnya

KODE

```
bg_image3 = Image.open("Merah Kuning Profesional Zoom Latar Belakang (5).png")
bg_image3 = bg_image3.resize((self.screen_width, self.screen_height), Image.Resampling.LANCZOS)
self.bg_photo_result = ImageTk.PhotoImage(bg_image3)
bg_label = tk.Label(self.frame_result, image=self.bg_photo_result)
bg_label.place(x=0, y=0, relwidth=1, relheight=1)

btn = tk.Button(self.frame_result, text=self.pred_jurusan, width=50, font=("Arial", 18),
                command=lambda: self.show_detail(self.pred_jurusan))
btn.place(relx=0.5, rely=0.4, anchor="center")

self.detail_label = tk.Label(self.frame_result, text="", font=("Arial", 14), bg="white", justify="left",
                             wraplength=1000, anchor="nw", relief="groove", bd=2, padx=10, pady=10)
self.detail_label.place_forget()

def show_detail(self, jurusan):
    row = jurusan_detail[jurusan]
    detail = (
        f"Jurusan: {jurusan}\n\n"
        f"Deskripsi: {row['Deskripsi']}\n\n"
        f"Prospek Kerja: {row['Prospek Kerja']}\n\n"
        f"Akkreditasi: {row['Akreditasi']}"
    )
    self.detail_label.config(text=detail)
    self.detail_label.place(relx=0.5, rely=0.5, anchor="n")

def print_tree_structure():
    print("Struktur Pohon Keputusan Berdasarkan Excel:\n")
    grouped = df.groupby(["Minat", "Minat Spesifik"])
    for (minat, minat_spesifik), group in grouped:
        print(f"Minat: {minat}")
        print(f"Minat Spesifik: {minat_spesifik}")
        for jurusan in group["Jurusan"]:
            print(f"    - {jurusan}")
        print()
```

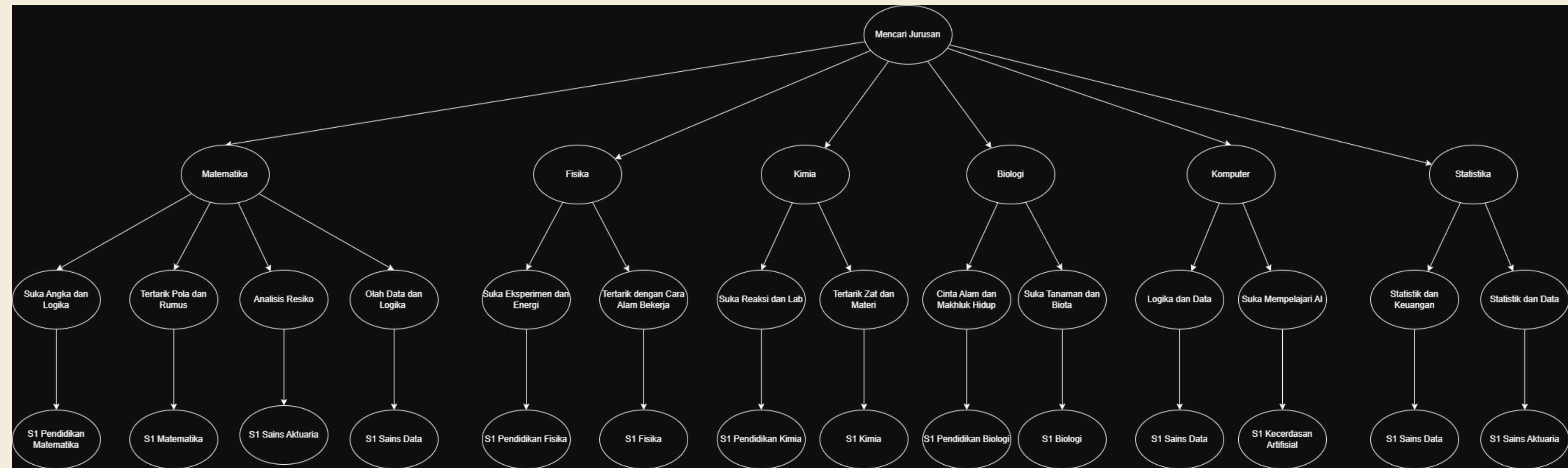
```
if __name__ == "__main__":
    print_tree_structure()
    root = tk.Tk()
    app = JurusanDecisionTreeApp(root)
    root.mainloop()
```

show_detail : Menampilkan deskripsi lengkap dari jurusan yang didapatkan, mulai dari deskripsi jurusan, prospek kerja dan akreditasinya

print_tree_structure : Mencetak struktur data *Decision Tree* sesuai dengan data yang ada di Excel

if __name__ : Untuk membuat GUI dan menjalankan aplikasi

TAMPILAN DECISION TREE



EXCEL

Minat	Minat Spesifik	Jurusan	Deskripsi	Prospek Kerja	Akreditasi
Matematika	Suka Angka dan Logika	S1 Pendidikan Matematika	Mempelajari metode pengajaran dan pemahaman konsep matematika Guru, Dosen, Pengembang Kurikulum	Guru, Dosen, Pengembang Kurikulum	A
Matematika	Tertarik Pola dan Rumus	S1 Matematika	Mendalami teori matematika serta penerapannya dalam berbagai bidang Peneliti, Analis Data, Statistisi	Peneliti, Analis Data, Statistisi	A
Matematika	Analisis Resiko	S1 Sains Aktuaris	Mempelajari pengelolaan risiko finansial menggunakan teori probabilitas Aktuaris, Analis Risiko	Aktuaris, Analis Risiko	B
Matematika	Olah Data dan Logika	S1 Sains Data	Fokus pada teknik pengumpulan, pengolahan, dan analisis data bersifat Data Scientist, Data Analyst	Data Scientist, Data Analyst	A
Fisika	Suka Eksperimen dan Energi	S1 Pendidikan Fisika	Menekankan pada strategi pengajaran dan penguasaan konsep dasar fisika Guru, Dosen	Guru, Dosen	B
Fisika	Tertarik dengan Cara Alam Bekerja	S1 Fisika	Mempelajari hukum-hukum fisika serta aplikasinya dalam riset dan penelitian Peneliti, Engineer, Akademisi	Peneliti, Engineer, Akademisi	A
Kimia	Suka Reaksi dan Lab	S1 Pendidikan Kimia	Mengkaji pendekatan dalam mengajarkan konsep kimia secara efektif Guru Kimia, Dosen	Guru Kimia, Dosen	B
Kimia	Tertarik Zat dan Materi	S1 Kimia	Fokus pada struktur dan reaksi zat kimia serta penerapannya di berbagai Peneliti, Industri Kimia	Peneliti, Industri Kimia	A
Biologi	Cinta Alam dan Makhluk Hidup	S1 Pendidikan Biologi	Mempersiapkan tenaga pendidik yang menguasai konsep dan pengajaran Guru, Dosen	Guru, Dosen	B
Biologi	Suka Tanaman dan Biota	S1 Biologi	Mempelajari makhluk hidup dan ekosistem serta penerapannya dalam Peneliti, Analis Lingkungan	Peneliti, Analis Lingkungan	A
Komputer	Logika dan Data	S1 Sains Data	Menyediakan pemahaman dan keterampilan dalam analisis dan interpretasi Data Scientist, Data Engineer	Data Scientist, Data Engineer	A
Komputer	Suka Mempelajari AI	S1 Kecerdasan Artificial	Mengembangkan kemampuan dalam merancang dan menerapkan sistem AI Engineer, Machine Learning Engineer	Engineer, Machine Learning Engineer	A
Statistika	Statistik dan Keuangan	S1 Sains Data	Mempelajari teknik statistik modern untuk mengolah dan menganalisis Data Analyst, Statistisi	Data Analyst, Statistisi	A
Statistika	Statistik dan Data	S1 Sains Aktuaris	Menggabungkan ilmu statistik dan finansial untuk analisis risiko dan penelitian Aktuaris, Analis Risiko	Aktuaris, Analis Risiko	B

TAMPILAN UI





THANK YOU

Kelompok 8