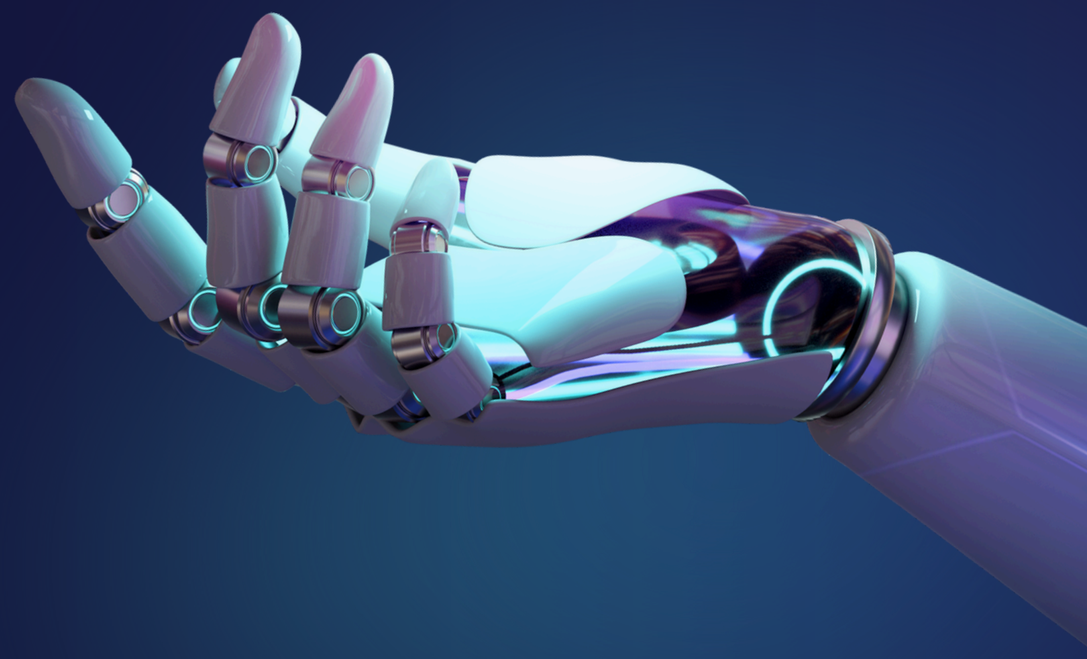
MBKM

BOT REGRESSION

Prediksi Harga Mobil Bekas

Directed by:
I Gede Sudiartika
Ni Kadek Aprianti
Komang Wahyu Kusuma Wardana
Anak Agung Gede Andre Kusuma
Cindy Hapsari
Komang Adi Satya Mahagangga



MBKM

PENJELASAN UMUM

Bot ini berfungsi sebagai alat untuk calon pembeli dalam proses pembelian mobil bekas untuk memberikan prediksi harga yang sesuai dengan fungsi atau fitur yang diinginkan, seperti merk, jenis bahan bakar dan masih banyak lainnya. Media bot ini menggunakan Telegram agar dapat diakses oleh semua orang.

MBKM

FITUR BOT

Beberapa fitur bot ini didukung oleh random forest regression untuk menghasilkan harga mobil sesuai hasil prediksi berdasarkan fitur-fitur dalam mobil.

01 User dapat menentukan merk mobil bekas yang diinginkan.

02 User mendapatkan rekomendasi harga berdasarkan merk.

03 User dapat menentukan fitur-fitur mobil bekas sesuai yang diinginkan

MBKM

PANDUAN PEMAKAIAN

Pemakaian bot dapat dilakukan berulang untuk mendapatkan harga prediksi yang sesuai dengan *budget User* dengan fitur mobil yang sudah ditentukan.

01 **Silahkan akses @car_predict_bot secara private atau panggil /mulai@car_predict_bot untuk memanggil bot**

02 **Ketika bot dimulai, silahkan pilih fitur mobil yang diinginkan untuk menentukan harga prediksi dan hasil prediksi akan muncul di akhir**

03 **Jika ingin memulai perbincangan dengan bot, maka panggil kembali command / yang ada di awal perbincangan**

MBKM

MEKANISME INTEGRASI

Pada bot menggunakan library ‘telepot’ untuk menjalankan proses bot dari awal hingga akhir.

MBKM

MEKANISME INTEGRASI PADA CLASS TELEBOT

```python
def handle_message(self, msg: Dict) -> None:
    """
    Menangani pesan pada bot telegram
    """
    chat_id = msg['chat']['id']
    username = msg['from']['username']
    if (msg['text'] == '/mulai') or (msg['text'] == '/mulai@Teamtwo_bot'):
        buttons = InlineKeyboardMarkup(inline_keyboard=[
            [InlineKeyboardButton(text=x, callback_data=x.lower()) for x in list_of_merk[i:i+2]]
            for i in range(0, len(list_of_merk), 2)
        ])

        self.bot.sendMessage(chat_id, 'Selamat datang di Bot Prediksi Harga Mobil...\n\nSilakan pilih merk:
        return
```

Fungsi handle_message menangani pesan yang diterima oleh bot. Ketika pengguna mengirimkan perintah /mulai, bot akan mengirimkan pesan dengan tombol inline yang memungkinkan pengguna untuk memilih merk mobil.

MBKM

MEKANISME INTEGRASI PADA CLASS TELEBOT

```python
def handle_input_text(self, text: str, user_id: int):
    """
    Mencari prediksi yang belum diinputkan oleh user
    """
    for message_id, prediksi in self.all_prediksi.items():
        if prediksi.user_id == user_id:
            if prediksi.year is None:
                prediksi.set_year(text)
                return (False, prediksi)
            elif prediksi.km_driven is None:
                prediksi.set_km_driven(text)
                return (True, prediksi)
    return (False, None)
```

Fungsi handle_input_text, mencari prediksi yang belum selesai untuk pengguna yang memberikan input dan memperbarui nilai yang belum diisi.

MBKM

MEKANISME INTEGRASI PADA CLASS TELEBOT

```python
def add_new_prediction(self, prediksi: PrediksiHarga) -> None:
    """
    Menambahkan objek prediksi ke dictionary
    """
    prediksi.save()
    self.all_prediksi[prediksi.message_id] = prediksi
```

Fungsi add_new_prediction menambahkan objek PrediksiHarga baru ke dictionary all_prediksi

```python
def delete_prediction(self, message_id: int) -> None:
    """
    Menghapus objek prediksi berdasarkan message_id
    """
    if message_id in self.all_prediksi:
        self.all_prediksi[message_id].delete()
        del self.all_prediksi[message_id]
```

Fungsi delete_prediction menghapus objek prediksi yang telah selesai.

MBKM

MEKANISME INTEGRASI PADA CLASS TELEBOT

```python
def run(self) -> None:
    """
    Menjalankan bot dan menangani pesan dengan menggunakan `MessageLoop`
    """
    MessageLoop(self.bot, {'chat': self.handle_message,
                           'callback_query': self.on_callback_query}).run_as_thread()
    print('Bot sedang berjalan...')
    while True:
        pass
```

Fungsi ‘run’ menjalankan bot menggunakan ‘MessageLoop’ dari library telepot untuk menangani pesan dan callback query secara asinkron.

MBKM

MEKANISME INTEGRASI PADA CLASS TELEBOT

```python
def on_callback_query(self, msg: Dict) -> None:
    """
    Menangani callback query (ketika tombol ditekan)
    """
    query_id, _, query_data = telepot.glance(msg, flavor='callback_query')
    chat_id = msg['message']['chat']['id']
    message_id = msg['message']['message_id']
    username = msg['from']['username']
    # print(msg)

    chat_type = msg['message']['chat']['type']

    print(f"Callback query ditekan secara {chat_type}")
    # print(f"Isi msg: {msg}")

    self.handlers[query_data](self, query_id, chat_id, message_id, username, chat_type)
```

Fungsi on_callback_query menangani callback ketika pengguna menekan tombol inline yang dikirimkan oleh bot.

MBKM

PENJELASAN ALGORITMA

Algoritma yang digunakan untuk memprediksi harga mobil yaitu **Random Forest.**

Algoritma ini merupakan metode ensambel yang menggabungkan beberapa pohon keputusan (**decision trees**) untuk menghasilkan hasil yang lebih akurat dan stabil.

MBKM

DESIGN CODE

Konsep struktur data

Dictionary (Dict):

```
self.handlers: Dict[str, function]
```

List (list)

```
buttons = InlineKeyboardMarkup(inline_keyboard=[
    [InlineKeyboardButton(text=x, callback_data=x.lower()) for x in list_of_merk[i:i+2]]
    for i in range(0, len(list_of_merk), 2)
```

Tuple

```
query_id, _, query_data = telepot.glance(msg, flavor='callback_query')
```

MBKM

CLASS

```python
class TeleBot:
    def get_prediction(self, message_id: int) -> PrediksiHarga:
        """
        Mengembalikan objek prediksi berdasarkan message_id
        """
        if message_id not in self.all_prediksi:
            prediction = PrediksiHarga.load(message_id)
            if prediction is not None:
                self.add_new_prediction(prediction)
                return prediction
            return None
        return self.all_prediksi[message_id]
```

Class TeleBot

MBKM

CLASS

```python
class PrediksiHarga:
    def __init__(self, message_id: int, user_id: int):
        self.merk: str | None = None
        self.year: int | None = None
        self.km_driven: int | None = None
        self.fuel: str | None = None
        self.seller_type: str | None = None
        self.transmission: str | None = None
        self.owner: str | None = None
        self.message_id = message_id
        self.user_id = user_id
```

Class Prediksi Harga

MBKM

FUNGSI

```python
def buttons_merk(teleBot: TeleBot, query_id: int, chat_id: int, message_id: int, user_id: int, chat_type:str):
    teleBot.bot.answerCallbackQuery(query_id, text='Pilih merk:')
    msg_id = (chat_id, message_id)

    buttons = InlineKeyboardMarkup(inline_keyboard=[
        [InlineKeyboardButton(text=x, callback_data=x.lower()) for x in list_of_merk]
        ])

    teleBot.bot.editMessageText(msg_id, 'Selamat datang di Bot Prediksi Harga Mobil...\n\nSilakan pilih merk:')
    teleBot.bot.editMessageReplyMarkup(msg_id, reply_markup=buttons)
```

Fungsi untuk menentukan Merk Mobil

MBKM

FUNGSI

```python
def fuel_buttons(teleBot: TeleBot, query_id: int, chat_id: int, message_id: int, user_id: int, chat_type:str):
    teleBot.bot.answerCallbackQuery(query_id, text='Pilih merk:')
    msg_id = (chat_id, message_id)

    buttons = InlineKeyboardMarkup(inline_keyboard=[
        [InlineKeyboardButton(text=x, callback_data=x.lower()) for x in list_of_fuel]
        ])

    teleBot.bot.editMessageText(msg_id, 'Silahkan pilih jenis bahan bakar:')
    teleBot.bot.editMessageReplyMarkup(msg_id, reply_markup=buttons)
```

Fungsi untuk menentukan Bahan Bakar Mobil

MBKM

FUNGSI

```python
def seller_type_buttons(teleBot: TeleBot, query_id: int, chat_id: int, message_id: int, user_id: int, chat_type:str):
    teleBot.bot.answerCallbackQuery(query_id, text='Pilih jenis seller:')
    msg_id = (chat_id, message_id)

    buttons = InlineKeyboardMarkup(inline_keyboard=[
        [
            InlineKeyboardButton(text=x, callback_data=x.lower()) for x in list_of_seller_type[:2]
        ],
        [
            InlineKeyboardButton(text=x, callback_data=x.lower()) for x in list_of_seller_type[2:]
        ]
    ])

    teleBot.bot.editMessageText(msg_id, 'Silahkan pilih jenis seller:')
    teleBot.bot.editMessageReplyMarkup(msg_id, reply_markup=buttons)
```

Fungsi untuk menentukan Seller Mobil

MBKM

FUNGSI

```python
def transmission_buttons(teleBot: TeleBot, query_id: int, chat_id: int, message_id: int, user_id: int, chat_type:str):
    teleBot.bot.answerCallbackQuery(query_id, text='Pilih jenis transmisi:')
    msg_id = (chat_id, message_id)

    buttons = InlineKeyboardMarkup(inline_keyboard=[
        [InlineKeyboardButton(text=x, callback_data=x.lower()) for x in list_of_transmission]
    ])

    teleBot.bot.editMessageText(msg_id, 'Silahkan pilih jenis transmisi:')
    teleBot.bot.editMessageReplyMarkup(msg_id, reply_markup=buttons)
```

Fungsi untuk menentukan Transmisi Mobil

MBKM

FUNGSI

```python
def owner_buttons(teleBot: TeleBot, query_id: int, chat_id: int, message_id: int, user_id: int, chat_type:str):
    teleBot.bot.answerCallbackQuery(query_id, text='Pilih owner:')
    msg_id = (chat_id, message_id)

    buttons = InlineKeyboardMarkup(inline_keyboard=[
        [
            InlineKeyboardButton(text=x, callback_data=x.lower()) for x in list_of_owner[:len(list_of_owner)//2]
        ],
        [
            InlineKeyboardButton(text=x, callback_data=x.lower()) for x in list_of_owner[len(list_of_owner)//2:]
        ],
    ])

    teleBot.bot.editMessageText(msg_id, 'Silahkan pilih owner:')
    teleBot.bot.editMessageReplyMarkup(msg_id, reply_markup=buttons)
```

Fungsi untuk menentukan Owner Mobil

MBKM

THANK YOU

ARTIFICIAL INTELLIGENCE PRESENTATION