

# **FOOD RECOMENDATION**

Pemrosesan Text Kelompok 9 2023A

---

Shafira Nabila Noer Poerwanto (011)

Ratu Arum Rahma Gati (142)

Chiara Dewi Chatlina (170)



# Latar Belakang

---

Pemborosan makanan adalah masalah besar yang merugikan ekonomi dan lingkungan. Banyak bahan makanan terbuang karena kurangnya pemahaman masyarakat dalam mengolahnya, terutama saat bahan dianggap tidak cukup untuk dijadikan masakan.

Limbah makanan berkontribusi pada emisi gas rumah kaca dan memperburuk krisis lingkungan. Solusi berbasis teknologi, seperti sistem rekomendasi resep berdasarkan bahan yang tersedia, dapat membantu masyarakat memanfaatkan bahan makanan dengan lebih bijak, mengurangi limbah, menghemat biaya, dan mendukung keberlanjutan.



```
● ● ●

data = data[['Title', 'Ingredients', 'Instructions', 'Image_Name']]
data = data.dropna()

lemmatizer = WordNetLemmatizer()
stop_words = set(stopwords.words('english'))

def clean_text(text):
    """
    Membersihkan teks dengan lowercasing, lemmatization, dan menghapus stopwords.
    """
    words = text.lower().split()
    words = [lemmatizer.lemmatize(word) for word in words if word not in stop_words and
wordretakphà()]join(words)

data['Ingredients'] = data['Ingredients'].apply(clean_text)
data['Instructions'] = data['Instructions'].apply(clean_text)
```

# Proses Pre-Processing

---

Pada awal tahap ini kami melakukan drop column yang tidak diperlukan selain Tittle, Ingredients, Instructions dan Image\_Name. Lalu setelah itu kami melakukan tahap pre-processing pada column ingredients dan instructions. Kami melakukan lowercasing, lemmatization dan menghapus stopwords

Pada tahap ini kami melakukan tahap feature engineering pada column ingredients. Kami menggunakan tf-idf untuk melakukan tahap feature engineering. Kami menggunakan tf-idf untuk mengukur seberapa penting sebuah kata dalam resep tertentu dibandingkan dengan keseluruhan data set. TF digunakan untuk menghitung frekuensi sebuah kata muncul dalam sebuah resep. IDF digunakan untuk mengurangi bobot kata-kata yang sering muncul di banyak rsep



```
tfidf = TfidfVectorizer()  
tfidf_matrix = tfidf.fit_transform(data['Ingredients'])
```

# Proses Feature Engineering

---

# Proses Modeling

Pada tahap ini, kami membangun sistem rekomendasi resep menggunakan algoritma K-Nearest Neighbors (KNN) dengan jarak kosinus untuk mengukur kesamaan antar bahan makanan. Model ini dilatih menggunakan matriks TF-IDF, yang merepresentasikan hubungan antar kata dalam data resep.

Proses dimulai dengan membersihkan input pengguna, seperti mengubahnya menjadi huruf kecil, memisahkannya berdasarkan koma, melakukan lemmatization, dan menyaring kata-kata yang hanya berupa huruf. Input yang telah diproses kemudian digabung kembali menjadi string dan diubah menjadi vektor TF-IDF.

Vektor ini digunakan untuk mencari resep yang paling relevan dengan input pengguna melalui model KNN. Hasil akhirnya adalah rekomendasi resep berupa judul, bahan-bahan, instruksi memasak, dan nama gambar yang sesuai dengan bahan yang dimasukkan. Tahapan ini memastikan pengguna mendapatkan rekomendasi yang optimal sesuai bahan yang dimiliki.

```
knn = NearestNeighbors(n_neighbors=5, metric='cosine')
knn.fit(tfidf_matrix)

def preprocess_input(user_input):
    """
    Membersihkan input dari user dengan proses yang sama seperti data training.
    """
    words = user_input.lower().split(',')
    words = [lemmatizer.lemmatize(word.strip()) for word in words if word.isalpha()]
    return words

def knn_recommend(user_input):
    """
    Mencari rekomendasi resep menggunakan model KNN.
    """
    user_input = preprocess_input(user_input)
    user_input_str = ' '.join(user_input)
    input_vec = tfidf.transform([user_input_str])
    distances, indices = knn.kneighbors(input_vec)
    return data.iloc[indices[0]][['Title', 'Ingredients', 'Instructions', 'Image_Name']]
```

# Hasil dan Analisis

Hasil output rekomendasi memunculkan beberapa rekomendasi sesuai dengan bahan yang di input user. Kami menggunakan algoritma KNN, algoritma ini dipilih karena memiliki runtime yang cepat (0.0010 detik) dan akurasi tinggi (1.0000), menunjukkan efisiensi dan kemampuan klasifikasi yang sangat baik. Validasi tambahan, seperti cross-validation, disarankan untuk memastikan model tidak overfit, menjadikan KNN pilihan yang tepat untuk analisis berbasis kemiripan data.



# Kontribusi Anggota

80%

80%

80%

**Shafira Nabila N P**

Melakukan pre processing dan  
modeling

**Ratu Arum Rahma G**

Melakukan feature engineering  
dan modeling

**Chiara Dewi C**

Melakukan modeling dan design  
antar muka

# THANK YOU!

And See You Next Time

---

