PERANCANGAN PENGEMBANGAN MODEL MACHINE LEARNING DETEKSI BERITA HOAX

LAPORAN UJIAN TENGAH SEMESTER.

Diajukan sebagai syarat kelulusan Mata Kuliah pengembangan aplikasi mobile.

Disusun Oleh:

VANNY MUSTAQIMAH

10222116

PROGRAM STUDI INFORMATIKA

SEKOLAH TINGGI TEKNOLOGI CIPASUNG

TASIKMALAYA

2025

1. Latar Belakang

Di era digital ini, kemajuan pesat dalam teknologi informasi dan komunikasi telah mengubah secara mendasar bagaimana informasi diakses dan disebarluaskan oleh masyarakat.

Kini, masyarakat mayoritas mengandalkan media sosial, portal berita daring, dan berbagai platform pesan sebagai sumber utama untuk mendapatkan kabar dan isu-isu terbaru.

Akan tetapi,kemudahan penyaluran informasi ini juga turut memfasilitasi penyebaran berita palsu (hoax) yang berpotensi menyesatkan masyarakat luas.Melihat kondisi ini, pengembangan sistem Deteksi Berita Hoax yang otomatis dan objektif untuk memverifikasi kebenaran informasi menjadi sangat penting dan strategis

2. Tujuan Pengembangan

Pembangunan model *machine learning* deteksi berita *hoax* ini memiliki tujuan utama untuk mendukung proses identifikasi dan klasifikasi berita secara otomatis. Tujuan ini didasari oleh kebutuhan untuk menyediakan informasi yang akurat, terverifikasi, dan bebas dari konten menyesatkan bagi masyarakat.

Tujuan spesifik dari pengembangan ini adalah:

a. Membangun sebuah model kecerdasan buatan berbasis *machine learning* yang mampu menganalisis teks berita.
b. Mengklasifikasikan teks berita ke dalam dua kategori utama, yaitu:
    - ➔ **Hoax:** Berita yang terbukti tidak benar, menyesatkan, atau bersifat provokatif tanpa dasar fakta.
    - ➔ **Faktual (Valid):** Berita yang bersumber dari informasi terpercaya dan memiliki dasar bukti yang jelas.
c. Menghadirkan sistem pendukung keputusan berbasis data dalam menanggulangi penyebaran berita palsu.

3. Deskripsi Dataset

Dataset yang digunakan dalam pengembangan model ini merupakan kumpulan berita daring (*online news*) yang telah melalui proses pelabelan berdasarkan kategori

kebenaran informasinya, yaitu hoax dan faktual (valid). File dataset disimpan dalam format *Comma-Separated Values* (CSV) dengan nama file berita_HOAX_indonesia.csv. File ini menggunakan 499 baris data yang dalam pelatihan.

Dataset ini berfungsi sebagai data latih (*training data*) dan data uji (*testing data*) untuk membangun dan mengevaluasi model klasifikasi teks. Setiap baris dalam dataset mewakili satu entri berita lengkap, terdiri dari kolom kategori sebagai label dan kolom berita yang memuat teks lengkap yang akan dianalisis.

Tujuan utama penggunaan dataset ini adalah untuk melatih model agar mampu mempelajari pola linguistik, struktur kalimat, dan pemilihan kata yang membedakan secara otomatis antara berita *hoax* dan berita faktual.

Konten dalam dataset ini seluruhnya disajikan dalam Bahasa Indonesia dan mencakup berbagai topik yang pernah menjadi isu publik, baik berupa fakta maupun misinformasi. Topik berita yang dimuat sangat beragam dan dinamis, meliputi:

a. Isu Bencana Alam: Meliputi laporan terkait peristiwa alam seperti erupsi gunung berapi.
b. Isu Sosial dan Politik: Berisi klaim-klaim sensitif terkait kebijakan publik, aktivitas aparat keamanan, atau isu internasional yang memicu reaksi di Indonesia.
c. Informasi Misleading Umum: Mencakup klaim-klaim yang menyebar cepat, seperti informasi terkait kesehatan, *sweeping* wilayah, hingga klaim-klaim ilmiah yang tidak terverifikasi (contoh: planet emas).

4. Struktur Dataset

Tabel 4.1 struktur dataset

No.	Nama kolom	Type data	Deskripsi
1.	Id	*String* / Integer	Nomor unik untuk setiap berita (diasumsikan ada untuk identifikasi baris data).
2.	Title	String	Judul berita (dikombinasikan di dalam kolom berita dataset).
3.	Text	String	Isi atau konten utama dari berita, berisi paragraf teks dalam format bebas (diwakili oleh kolom berita dataset).
4.	Label	String	Kategori kebenaran berita, terdiri dari nilai: hoax atau valid (diwakili oleh kolom kategori dataset).

Struktur dataset yang disajikan pada Tabel 3.1 merupakan pemetaan logis dari data yang ada di file berita_HOAX_indonesia.csv ke dalam format standar *Natural Language Processing* (NLP) untuk keperluan perancangan model. Meskipun file sumber secara fisik hanya memiliki dua kolom (kategori dan berita), pemetaan ini diperlukan untuk mengaitkan setiap elemen data dengan peran fungsionalnya dalam proses pelatihan model.Berikut adalah penjelasan rincinya:

a. Id (Nomor Urut Identifikasi): Kolom ini diasumsikan ada secara logis untuk memberikan identitas unik pada setiap baris data (berita). Dalam konteks implementasi, kolom ini berfungsi untuk memudahkan pelacakan data selama proses *data loading* dan *debugging* model.
b. Title dan Text (Konten Berita): Kedua kolom ini diwakili oleh kolom tunggal berita dalam dataset yang digunakan. Hal ini mengasumsikan bahwa kolom berita memuat gabungan dari judul dan isi utama berita. Konten pada kolom ini adalah fitur utama yang akan diolah melalui teknik *text tokenization* dan *feature extraction* (seperti *N-Grams* atau *Word Embedding*) untuk diubah menjadi representasi numerik yang dapat dipelajari oleh model.
c. Label (Kategori Kebenaran): Kolom ini secara aktual diwakili oleh kolom kategori dalam dataset. Kolom ini adalah variabel target (*target variable*) yang akan diprediksi oleh model. Label ini memiliki nilai diskrit (hoax atau valid), yang menandakan tugas klasifikasi multikelas yang harus diselesaikan oleh model.

Secara keseluruhan, struktur ini memastikan bahwa data mentah dapat diserap dan dikonversi menjadi representasi yang siap untuk dilatih, dengan kolom berita sebagai *input* dan kolom kategori sebagai *output* yang diharapkan.

5. Contoh Dataset

Tabel 5.1 Contoh dataset

No.	Id	Title	Teks	Label
1.	1	Erupsi Gunung Agung di Bali	Gunung Agung erupsi untuk pertama kali pada 21 November 2017. Letusan terjadi pada pukul 17.05 Wita. Asap teramati bertekanan sedang dengan warna kelabu tebal dan dengan ketinggian maksimum sekitar 700 m di atas puncak... (Teks lengkap dari kolom berita Anda)	**valid**
2.	2	Ekspor Mobil CBU Indonesia Naik	Berdasarkan data Badan Pusat Statistik (BPS), ekspor mobil CBU (Completely Built Up) pada 2016 mencapai 211.777 unit. Angka ini naik 5,9 persen dibandingkan 2015 yang sebanyak 200.000 unit. Mobil yang diekspor didominasi merek-merek ternama	**valid**
3.	3	Suu Kyi Minta Indonesia Tutup Mulut	Kami hanya ingin orang-orang di Indonesia tutup mulut dan diam. Stop pembahasan mengenai Muslim Rohingya. Urus saja negeri kalian..." (Klaim yang dimuat di kolom berita Anda)	**hoax**
4.	4	Sweeping Brimob Terhadap OJOL Malam Hari	Hindari wilayah selatan, Blok M, Mampang, Buncit, Ampera, Kemang, Cipete, Fatmawati raya sampai Lotte Mart. pemberitahuan langsung keseluruh BC/grup/komunitas. Situasi selatan TIDAK KONDUSIF MALAM INI.	**hoax**
5.	5	Petisi Rahasia Referendum Papua Barat	Petisi Rahasia yang Menuntut Referendum Kemerdekaan Baru untuk Papua Barat Telah Dipresentasikan ke Perserikatan Bangsa-Bangsa.	**hoax**

6. Sumber dan Pengumpulan data

Dataset yang digunakan dalam perancangan model ini bersumber dari Kanggle, yaitu berita HOAX indonesia.csv. Dataset ini adalah kumpulan berita Indonesia yang telah melalui proses pelabelan validitas.

a. Nama File: berita_HOAX_indonesia.csv
b. Jumlah Data Total: Data awal memiliki jumlah entri yang substansial, namun data yang diolah dan dilatih dalam proyek ini berjumlah 499 baris.
c. Proses Pengolahan: Data ini telah disaring dan dibersihkan (*pre-processed*) untuk memastikan setiap entri memiliki kolom label (kategori) dan kolom teks berita (berita) yang lengkap dan siap untuk proses *machine learning*.

7. Alasan pemilihan dataset

Dataset ini dipilih sebagai *input* utama dalam proses pelatihan model deteksi berita *hoax* berdasarkan pertimbangan sebagai berikut: Bahasa dan Konteks Lokal: Dataset sepenuhnya menggunakan Bahasa Indonesia, mencerminkan kondisi berita daring aktual yang sering ditemui masyarakat Indonesia di media sosial dan portal berita. b. Keseimbangan Kategori: Dataset mencakup dua kategori utama yang jelas (hoax dan valid), menjadikannya ideal untuk melatih model klasifikasi multi kelas (*Multiclass Classification*) menggunakan *framework* ML.NET. c. Kebutuhan untuk Pola Linguistik: Meskipun jumlah data yang digunakan untuk pelatihan (499 baris) masih terbatas, data ini penting untuk menguji kemampuan model dalam mempelajari pola linguistik yang beragam, termasuk upaya *hoax* yang disamarkan dengan gaya bahasa formal. d. Efisiensi Komputasi: Jumlah data yang digunakan (499 entri) sangat efisien untuk diolah dan dilatih secara cepat di lingkungan komputasi lokal, memungkinkan iterasi dan evaluasi model dilakukan tanpa memerlukan sumber daya komputasi tinggi (GPU).

8. Arsitektur dan Alur Model

Model deteksi berita *hoax* dirancang menggunakan *framework* ML.NET pada *platform* .NET 9.0 (berdasarkan konfigurasi proyek yang digunakan), dengan pendekatan *supervised learning* (pembelajaran terawasi). Model ini berfungsi untuk menganalisis teks berita (yang diwakili oleh kolom berita pada dataset) serta memprediksi kategori kebenaran berdasarkan pola linguistik dan karakteristik teks.

Secara umum, sistem *pipeline* ini diorganisir menjadi tiga komponen utama yang saling terkait:

a. Data Preparation (Persiapan Data)

Tahap ini berfokus pada pemuatan, pembersihan, dan transformasi data mentah dari file berita_HOAX_indonesia.csv menjadi format yang dapat diproses secara numerik oleh algoritma *machine learning*.

Proses Utama:

- ➔ Data Loading: Memuat data dari file CSV dengan mengidentifikasi kolom kategori sebagai label dan kolom berita sebagai fitur teks.
- ➔ Data Splitting: Membagi data yang telah dimuat menjadi *Training Set* (untuk pelatihan) dan *Testing Set* (untuk evaluasi).
- ➔ Text Transformation: Menerapkan proses *Tokenization* dan *Key Mapping* untuk mengkonversi *string* (teks) menjadi representasi numerik.
- ➔ Feature Engineering: Ekstraksi fitur teks menggunakan teknik Hashed N-Grams untuk menghasilkan vektor fitur.

b. Model Building (Pembangunan dan Pelatihan Model)

Tahap ini melibatkan pemilihan dan pelatihan algoritma klasifikasi.

- ➔ Algoritma: Model dilatih menggunakan algoritma SDCA Maximum Entropy (*Stochastic Dual Coordinate Ascent*) karena efisiensinya yang baik dalam menangani tugas klasifikasi teks multikelas dengan fitur *sparse* (jarang).
- ➔ Proses Utama: Model diajarkan untuk memetakan vektor fitur yang dihasilkan dari kolom berita ke nilai *target* di kolom kategori (hoax atau valid).

c. Model Evaluation & Deployment (Evaluasi dan Implementasi Model)

Tahap terakhir adalah mengukur kinerja model dan menyimpannya untuk penggunaan di masa mendatang.

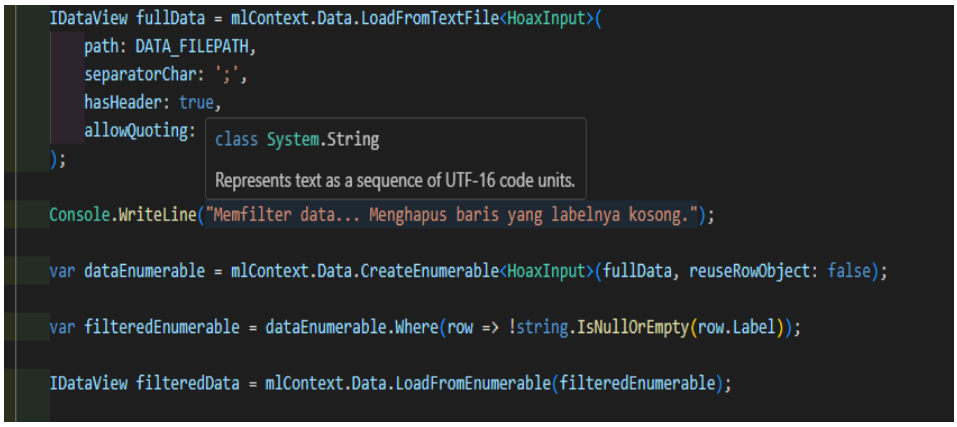
- ➔ Evaluasi: Model diuji menggunakan *Testing Set*. Metrik utama yang digunakan adalah MicroAccuracy (yang menghasilkan nilai 83.17%), MacroAccuracy, dan LogLoss untuk mengukur kualitas prediksi.

➔ Penyimpanan: Setelah evaluasi, model disimpan dalam format .zip (hoax_detector_model.zip) agar siap diimplementasikan untuk melakukan prediksi tunggal atau *batch prediction* di lingkungan *production*.

8.1 Komponen Utama *Pipeline* ML.NET

Pipeline (alur pemrosesan) dalam ML.NET adalah serangkaian *transformer* yang dikonfigurasi untuk memproses data mentah menjadi format yang dapat dipelajari oleh model, melatih model, dan mengevaluasinya. Setiap langkah dalam *pipeline* dirancang khusus untuk menangani data teks dan mengoptimalkan ekstraksi fitur.

a. Data Loading & Filtering



Gambar 8.1.1 Data Loading & Filtering

Data dimuat dari file berita_HOAX_indonesia.csv dengan pemisah titik koma (;). Kemudian, data dibersihkan (*filtered*) untuk menghilangkan baris yang nilai labelnya kosong (null atau string kosong), memastikan kualitas data latih.

b. Data Splitting (Pembagian Data)

```
var splitData = mlContext.Data.TrainTestSplit(filteredData, testFraction: 0.2);

Console.WriteLine("========== Mulai Pelatihan Model ==========");
ITransformer model = pipeline.Fit(splitData.TrainSet);
Console.WriteLine("========== Pelatihan Selesai ==========\n");
```

Gambar 8.1.2 Data Splitting (Pembagian Data)

Data yang sudah bersih dibagi menjadi dua set: Training Set (80%) dan Testing Set (20%). *Training Set* digunakan untuk melatih model, sedangkan *Testing Set* digunakan untuk menguji akurasi model.

c. Map Value To Key (Transformasi Label)

```
var pipeline = mlContext.Transforms.Conversion.MapValueToKey(inputColumnName: "Label", outputColumnName: "LabelKey")
    .Append(mlContext.Transforms.Text.FeaturizeText(outputColumnName: "Features", inputColumnName: "Text"))
    .Append(mlContext.MulticlassClassification.Trainers.SdcaMaximumEntropy(labelColumnName: "LabelKey", featureColumnName: "Features"))
    .Append(mlContext.Transforms.Conversion.MapKeyToValue("PredictedLabel"));
```

Gambar 8.1.3 Map Value To Key (Transformasi Label)

Langkah pertama dalam *core pipeline* adalah Transformasi Label menggunakan fungsi MapValueToKey(). Berdasarkan *screenshot* kode di atas, fungsi ini diaplikasikan pada kolom *input* Label yang berisi nilai *string* ("hoax" atau "valid"). Tujuannya adalah untuk mengkonversi nilai *string* tersebut menjadi representasi numerik (key index) yang dapat diproses oleh algoritma SDCA Maximum Entropy. Hasil konversi ini disimpan dalam kolom baru bernama LabelKey

d. Feature Extraction (*FeaturizeText*)

Gambar 8.1.4 Feature Extraction (*FeaturizeText*)

Fungsi FeaturizeText() adalah langkah kunci dalam *Feature Engineering*. Berdasarkan *screenshot* kode di atas, fungsi ini diaplikasikan pada kolom *input* Text (yang berisi isi berita). FeaturizeText secara otomatis melakukan serangkaian transformasi teks, termasuk *Tokenization* (pemecahan kata) dan *Vectorization* (pengubahan kata menjadi vektor numerik). Hasil dari proses ini adalah kolom baru bernama Features yang berisi vektor fitur padat, siap digunakan oleh algoritma pelatihan

e. Model Training (*SdcaMaximumEntropy*)

Gambar 8.1.5 Model Training (*SdcaMaximumEntropy*)

Sdca Maximum Entropy() adalah algoritma pelatihan utama yang dipilih untuk model klasifikasi teks ini. Berdasarkan *screenshot* kode di atas, fungsi ini menerima dua *input* krusial: kolom LabelKey (target numerik) dan kolom Features (vektor fitur teks). SDCA Maximum Entropy adalah pilihan yang sangat efisien untuk data teks dengan fitur *sparse* (jarang) dan efektif untuk masalah klasifikasi multikelas. Algoritma ini akan mempelajari hubungan antara pola vektor fitur dengan label hoax atau valid

f. Map Key To Value (Label Decoding)

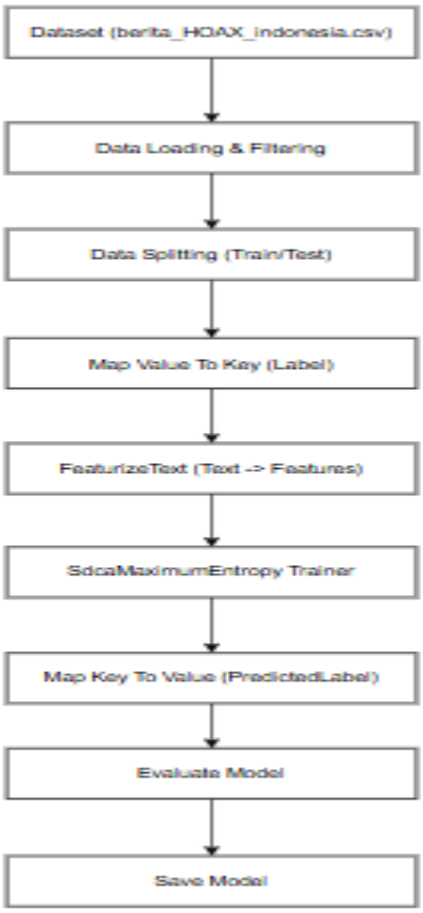
Gambar 8.1.6 Map Key To Value (Label Decoding)

Langkah MapKeyToValue() berfungsi sebagai *label decoding*. Setelah model dilatih, hasil prediksi dari algoritma SDCA Maximum Entropy masih berbentuk *key index* numerik. Fungsi ini akan mengkonversi kembali *key index* numerik tersebut menjadi label *string* aslinya, yaitu 'hoax' atau 'valid'. Hasil akhir prediksi yang mudah dibaca ini kemudian disimpan di kolom Predicted Label, siap untuk disajikan kepada pengguna.

8.2 Diagram Alur Proses Model

Berikut diagram konseptual dari pipeline model :

Tabel 8.2 Diagram alur proses model



9. Hasil dan Evaluasi

Model yang telah dilatih kemudian diuji menggunakan Testing Set (20% dari total 499 baris data). Evaluasi ini bertujuan untuk mengukur kinerja dan mengidentifikasi kelemahan model sebelum diimplementasikan**.**

a. Pelatihan Model (*Fitting*)

```
ITransformer model = pipeline.Fit(splitData.TrainSet);
Console.WriteLine("========== Pelatihan Selesai ==========\n");
```

Gambar 8.1 Pelatihan Model (*Fitting*)

Pelatihan Model: Model dilatih dengan memasukkan *Training Set* ke dalam *pipeline* yang sudah didefinisikan sebelumnya. Hasil pelatihan disimpan di variabel model.

b. Evaluasi Model

```
EvaluateModel(model, splitData.TestSet);

mlContext.Model.Save(model, filteredData.Schema, MODEL_FILEPATH);
Console.WriteLine($"\nModel berhasil disimpan di: {MODEL_FILEPATH}");
Console.WriteLine("File .zip ini yang akan di-upload ke GitHub.\n");
```

Gambar 8.2 evaluasi model

Evaluasi Model: Memanggil fungsi pembantu untuk menguji performa model (model) menggunakan *Testing Set*. Hasil metrik (MicroAccuracy, dll.) akan dicetak ke konsol.

c. Penyimpanan Model

```
mlContext.Model.Save(model, filteredData.Schema, MODEL_FILEPATH);
Console.WriteLine($"\nModel berhasil disimpan di: {MODEL_FILEPATH}");
Console.WriteLine("File .zip ini yang akan di-upload ke GitHub.\n");
```

Gambar 8.3 Penyimpanan Modell

Penyimpanan Model: Menyimpan model yang sudah dilatih (model) ke dalam format file .zip (hoax_detector_model.zip), menjadikannya siap untuk digunakan (*deployment*).

d. Pengujian Prediksi Tunggal

```
private static void TestSinglePrediction(ITransformer model)
{
    var predictionEngine = mlContext.Model.CreatePredictionEngine<HoaxInput, HoaxPrediction>(model);

    var sampleData = new HoaxInput
    {
        Text = "Ditemukan planet baru di Tasikmalaya yang terbuat dari emas murni kata peneliti ITB."
    };

    var prediction = predictionEngine.Predict(sampleData);

    Console.WriteLine(                                                    ======");
    Console.WriteLine(
    Console.WriteLine(                                                    }");
    Console.WriteLine("============================================");
}
```

class System.String

Represents text as a sequence of UTF-16 code units.

Gambar 8.4 Pengujian Prediksi Tunggal

Pengujian Prediksi Tunggal: Memanggil fungsi pembantu untuk menguji model dengan satu contoh teks buatan tangan, yang hasilnya digunakan untuk menganalisis kelemahan model.

10. Kesimpulan

Proyek pengembangan model *machine learning* deteksi berita *hoax* telah berhasil diimplementasikan menggunakan *framework* ML.NET dengan algoritma SDCA Maximum Entropy yang dilatih menggunakan 499 baris data berita Bahasa Indonesia, menghasilkan Micro Accuracy awal sebesar 83.17%. Kinerja ini menunjukkan model memiliki kemampuan dasar yang baik dalam klasifikasi; namun, kelemahan krusial teridentifikasi ketika model gagal mendeteksi *hoax* yang disamarkan dengan gaya bahasa formal, sebuah keterbatasan yang diakibatkan oleh kuantitas data latih yang sangat minim, yang pada akhirnya membatasi pemahaman kontekstual model dan menegaskan bahwa pengembangan lanjutan sangat bergantung pada peningkatan skala data secara signifikan.

Refleksi Pembelajaran Machine Learning

Pada mulanya, saya memandang *machine learning* sebagai disiplin ilmu yang kaku dan didominasi oleh rangkaian rumus yang rumit.Namun, proyek implementasi ini membuka pandangan saya bahwa konsep intinya sebetulnya lebih intuitif dan sederhana, yakni melatih sistem komputasi agar mampu mengenali pola dan membuat keputusan berdasarkan data.

Dalam proses ini, saya belajar bahwa keberhasilan model tidak hanya ditentukan oleh kemampuan *coding*, tetapi juga oleh logika berpikir, pemecahan masalah, dan pemahaman mendalam tentang siklus hidup data. Saya mendapatkan pemahaman praktik tentang bagaimana data mentah harus diolah, diubah menjadi fitur numerik (*vectorization*), dan digunakan untuk melatih model (*fitting*). Meskipun demikian, saya menyadari bahwa masih banyak area yang perlu dikuasai, terutama pemahaman mendalam tentang optimasi *hyperparameter* dan arsitektur *deep learning*. Refleksi ini menegaskan bahwa *machine learning* adalah bidang yang dinamis, menuntut adaptasi berkelanjutan, dan memiliki potensi besar sebagai solusi berbasis data di berbagai sektor, jauh melampaui sebatas istilah akademik.