

Perancangan Sistem Rekomendasi Konten Video *Youtube* Berdasarkan Minat Pengguna Menggunakan Metode *Content-Based Filtering*

Neville Venerdi^{1*}, Ahmad Fitriansyah², Jamah Sari³

1. Program Studi Teknik Informatika, Institut Teknologi dan Bisnis Swadharma; nevillevenerdi@gmail.com
2. Program Studi Teknik Informatika, Institut Teknologi dan Bisnis Swadharma; ahmad.fitriansyah@swadharma.ac.id
3. Program Studi Teknik Informatika, Institut Teknologi dan Bisnis Swadharma; jamah@swadharma.ac.id

* Penulis Korespondensi: Neville Venerdi

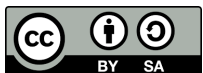
Abstract: YouTube provides a large number of videos with diverse topics, but users still often face difficulties in finding content that matches their current interests. Based on a questionnaire involving 60 respondents, 83.3% of respondents stated that they often receive repetitive YouTube video recommendations, 86.7% stated that excessive search results make the search process less directed, and 88.3% were interested in using a recommendation system that presents videos based on specific interests. This study aims to design a YouTube video content recommendation system based on user interests using the Content-Based Filtering method. The proposed system uses YouTube video metadata, including title, description, hashtag, channel name, duration, view count, likes, publication date, thumbnail, and video URL. The dataset consists of 1,225 videos grouped into 9 categories and 49 subcategories. Sentence-BERT (SBERT) is applied to represent metadata and user-selected interests as semantic embedding vectors, while Cosine Similarity is used to calculate the similarity between user interest queries and video metadata. The system generates five top recommendations for each selected subcategory, combines the results, and ranks them based on the highest similarity score. The implementation includes category and subcategory selection, recommendation display, result filtering, and access to videos on YouTube. Black Box Testing shows that the main system functions run according to user needs. Therefore, the proposed system can help users explore YouTube videos more directly, specifically, and relevantly based on selected interests.

Keywords: YouTube recommendation system; Content-Based Filtering; Sentence-BERT; Cosine Similarity; video metadata; user interest; Black Box Testing

Abstrak: YouTube menyediakan banyak konten video dengan berbagai topik, tetapi pengguna masih sering mengalami kesulitan dalam menemukan video yang sesuai dengan minat yang ingin dieksplorasi. Berdasarkan kuesioner terhadap 60 responden, sebanyak 83,3% responden menyatakan sering mendapatkan rekomendasi video YouTube yang serupa secara berulang, 86,7% menyatakan jumlah hasil pencarian yang terlalu banyak membuat proses pencarian menjadi kurang terarah, dan 88,3% tertarik menggunakan sistem rekomendasi yang dapat menampilkan video berdasarkan minat tertentu secara lebih spesifik. Penelitian ini bertujuan untuk merancang sistem rekomendasi konten video YouTube berdasarkan minat pengguna menggunakan metode *Content-Based Filtering*. Sistem yang diusulkan memanfaatkan metadata video, seperti judul, deskripsi, hashtag, nama channel, durasi, jumlah penonton, jumlah suka, tanggal publikasi, thumbnail, dan tautan video. Dataset yang digunakan terdiri dari 1.225 video yang dikelompokkan ke dalam 9 kategori dan 49 subkategori. Model *Sentence-BERT* (SBERT) digunakan untuk merepresentasikan metadata dan pilihan minat pengguna ke dalam vektor *embedding* semantik, sedangkan *Cosine Similarity* digunakan untuk menghitung kemiripan antara *query* minat pengguna dan metadata video. Sistem menghasilkan lima rekomendasi terbaik pada setiap subkategori yang dipilih, kemudian menggabungkan dan mengurutkan hasil berdasarkan nilai similarity tertinggi. Hasil implementasi menunjukkan bahwa sistem menyediakan fitur pemilihan kategori dan subkategori, tampilan rekomendasi, *filter* hasil, serta akses video ke YouTube. Pengujian *Black Box* menunjukkan bahwa fungsi utama sistem berjalan sesuai kebutuhan pengguna. Dengan demikian, sistem ini dapat membantu pengguna mengeksplorasi konten YouTube secara lebih terarah dan relevan berdasarkan minat yang dipilih.

Kata kunci: sistem rekomendasi YouTube; Content-Based Filtering; Sentence-BERT; Cosine Similarity; metadata video; minat pengguna; Black Box Testing

Diterima: June 30, 2026
Direvisi: July 21, 2026
Diterima: July 24, 2026
Diterbitkan: July 27, 2026
Versi sekarang: July 30, 2026



Hak cipta: © 2026 oleh penulis.
Diserahkan untuk kemungkinan
publikasi akses terbuka
berdasarkan syarat dan ketentuan
lisensi Creative Commons
Attribution (CC BY SA) (
<https://creativecommons.org/licenses/by-sa/4.0/>)

1. Pendahuluan

YouTube merupakan salah satu *platform* berbagi video yang banyak digunakan untuk mencari informasi, hiburan, pembelajaran, bisnis, kesehatan, teknologi, hingga pengembangan karier [1]. Banyaknya konten yang tersedia memberikan kemudahan bagi pengguna dalam memperoleh informasi, tetapi juga menimbulkan tantangan dalam proses penemuan konten atau *content discovery* [2]. Pengguna sering kali perlu mencoba beberapa kata kunci, membandingkan banyak video, dan menyeleksi sendiri konten yang dianggap sesuai dengan minatnya. Untuk memperkuat permasalahan tersebut, penelitian ini menggunakan kuesioner terhadap 60 responden yang terdiri dari mahasiswa Teknik Informatika dan Sistem Informasi. Kuesioner disusun menggunakan skala Likert dengan lima pilihan jawaban, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS), dan Sangat Tidak Setuju (STS). Kolom SS+S menunjukkan jumlah responden yang memberikan jawaban Sangat Setuju dan Setuju. Hasil kuesioner ditunjukkan pada Tabel 1.

Tabel 1. Hasil Kuesioner

No	Pernyataan	SS	S	N	TS	STS	SS+S
1	Saya sering mendapatkan rekomendasi video YouTube yang serupa secara berulang.	24	26	6	3	1	50 responden / 83,3%
2	Saya merasa hasil rekomendasi video YouTube belum selalu sesuai dengan minat yang ingin saya eksplorasi saat itu.	24	27	4	3	2	51 responden / 85,0%
3	Saya terkadang kesulitan menemukan konten video yang sesuai dengan minat tertentu di YouTube.	23	29	4	2	2	52 responden / 86,7%
4	Saya tidak selalu mengetahui kata kunci (<i>keyword</i>) yang tepat untuk mencari video sesuai minat saya.	23	27	4	4	2	50 responden / 83,3%
5	Saya lebih tertarik menggunakan sistem rekomendasi yang dapat menampilkan video berdasarkan minat tertentu secara lebih spesifik.	27	26	2	5	0	53 responden / 88,3%
6	Jumlah hasil pencarian video di YouTube terkadang terlalu banyak sehingga membuat proses pencarian menjadi kurang terarah.	27	25	6	2	0	52 responden / 86,7%

Sistem rekomendasi digunakan untuk membantu pengguna menemukan item yang relevan di tengah banyaknya pilihan informasi [3]. Pada *platform* video, rekomendasi dapat dibangun melalui histori interaksi pengguna, pencarian *keyword*, maupun kemiripan

7	Rekomendasi video yang muncul di YouTube sering dipengaruhi oleh histori tontonan sebelumnya.	28	24	5	3	0	52 responden / 86,7%
8	Saya ingin mendapatkan rekomendasi video yang lebih beragam sesuai minat yang ingin saya eksplorasi.	25	25	5	4	1	50 responden / 83,3%

karakteristik konten [4]. Pendekatan berbasis histori memiliki kelebihan karena dapat menyesuaikan rekomendasi dengan kebiasaan pengguna, tetapi berisiko membuat hasil yang muncul terlalu mengikuti aktivitas sebelumnya [5]. Sementara itu, pendekatan berbasis *keyword* lebih mudah digunakan karena pengguna cukup memasukkan kata kunci tertentu, tetapi metode ini menuntut pengguna mengetahui istilah pencarian yang tepat agar hasilnya sesuai dengan kebutuhan [6].

Permasalahan tersebut berkaitan dengan fenomena *filter bubble* dan *narrow exposure*. Thompson (2020) menjelaskan bahwa rekomendasi YouTube memiliki peran besar dalam membentuk pola konsumsi video pengguna [7]. Sukiennik dkk. (2024) juga membahas bahwa mekanisme rekomendasi pada *platform* video digital dapat membuat pengguna berada dalam lingkup rekomendasi yang relatif homogen [8]. *Filter bubble* terjadi ketika pengguna terus menerima konten serupa berdasarkan histori interaksi sebelumnya, sedangkan *narrow exposure* terjadi ketika pengguna hanya terpapar pada jenis konten tertentu secara berulang. Kondisi ini dapat membatasi eksplorasi pengguna ketika ingin mencari minat baru yang berbeda dari kebiasaan menonton sebelumnya.

Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan sistem rekomendasi konten video YouTube berdasarkan minat pengguna menggunakan metode *Content-Based Filtering*. Metode ini dipilih karena mampu memberikan rekomendasi berdasarkan karakteristik konten, seperti judul, deskripsi, *hashtag*, dan metadata video, sehingga tidak sepenuhnya bergantung pada histori pengguna lain [9]. Dataset yang digunakan terdiri dari 1.225 video yang dikelompokkan ke dalam 9 kategori dan 49 subkategori. Struktur kategori dan subkategori digunakan agar pengguna dapat menentukan minat secara lebih spesifik tanpa harus selalu menebak *keyword* pencarian yang tepat.

Pada proses rekomendasi, sistem menggunakan *Sentence-BERT* atau SBERT untuk membentuk representasi semantik dari metadata video dan pilihan minat pengguna. SBERT mengubah teks menjadi vektor *embedding*, sehingga hubungan makna antar teks dapat dibandingkan secara komputasional [10]. Selanjutnya, *Cosine Similarity* digunakan untuk menghitung tingkat kemiripan antara *query* minat pengguna dan metadata video [11]. Setiap subkategori yang dipilih menghasilkan lima video dengan nilai kemiripan tertinggi, kemudian seluruh hasil rekomendasi digabungkan dan diurutkan berdasarkan nilai *similarity* tertinggi agar video yang paling relevan tampil pada urutan teratas.

Berdasarkan paparan tersebut, penelitian mengenai sistem rekomendasi konten video YouTube berdasarkan minat pengguna sangat penting, menarik, dan perlu dilakukan. Penelitian ini diharapkan dapat menghasilkan sistem rekomendasi yang mampu membantu pengguna menemukan video secara lebih terarah melalui pemilihan kategori dan subkategori minat, dengan memanfaatkan *Content-Based Filtering*, SBERT, dan *Cosine Similarity*. Melalui pendekatan tersebut, sistem dapat menampilkan rekomendasi video yang lebih relevan, mengurutkan hasil berdasarkan nilai *similarity* tertinggi, serta menyediakan fitur *filter* kategori, *filter* subkategori, mode semua, dan akses menuju video asli pada *platform* YouTube.

2. Tinjauan Literatur

Ananta dan Supriana (2023) meneliti sistem rekomendasi konten YouTube menggunakan metode *Content-Based Filtering* dengan durasi menonton sebagai indikator minat pengguna [12]. Penelitian tersebut sudah menerapkan perhitungan kemiripan menggunakan *Cosine Similarity*, tetapi masih berfokus pada histori aktivitas pengguna. Witjaksana (2023) juga mengembangkan sistem rekomendasi video YouTube pada topik *Information Technology* dengan memanfaatkan metadata judul dan deskripsi melalui YouTube API [13]. Namun, penelitian

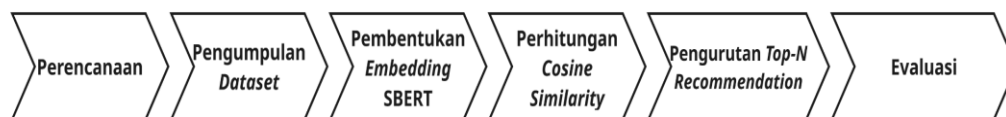
tersebut masih bergantung pada input *keyword*, terbatas pada satu topik, dan belum menggunakan struktur kategori serta subkategori minat.

Sistem rekomendasi pada dasarnya digunakan untuk membantu pengguna menemukan item yang relevan di tengah banyaknya pilihan informasi [14]. Salah satu pendekatan yang sering digunakan adalah *Content-Based Filtering*, yaitu metode yang memberikan rekomendasi berdasarkan karakteristik konten seperti judul, deskripsi, *hashtag*, dan metadata lainnya [15]. Dalam pengolahan teks, *Sentence-BERT* atau SBERT dapat digunakan untuk membentuk representasi semantik dalam bentuk vektor *embedding*, sehingga kemiripan makna antar teks dapat dihitung secara lebih baik menggunakan *Cosine Similarity* [16].

Berdasarkan penelitian terdahulu, masih terdapat kesenjangan pada sistem rekomendasi video YouTube yang lebih menekankan pilihan minat pengguna secara langsung. Penelitian ini berbeda karena menggunakan struktur 9 kategori dan 49 subkategori sebagai dasar pemilihan minat pengguna. Selain itu, sistem yang diusulkan memanfaatkan SBERT dan *Cosine Similarity* untuk menghitung kemiripan antara *query* minat pengguna dan metadata video, sehingga rekomendasi yang dihasilkan tidak hanya bergantung pada histori tontonan atau pencarian *keyword*.

3. Metode

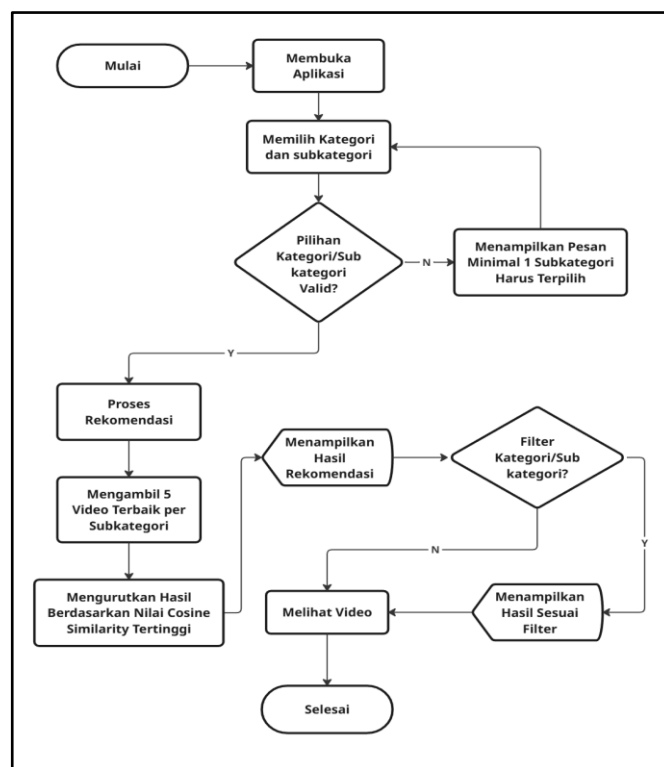
Penelitian ini dilakukan melalui enam tahap, yaitu *perencanaan*, pengumpulan *dataset*, pembentukan *embedding* SBERT, perhitungan *Cosine Similarity*, pengurutan *Top-N Recommendation*, dan evaluasi. Keenam tahap tersebut disusun agar proses rekomendasi berjalan secara terarah, mulai dari penentuan alur sistem sampai pengujian hasil akhir.



Gambar 1. Alur Metode Penelitian

3.1. Perencanaan

Tahap perencanaan dilakukan untuk menentukan alur kerja sistem rekomendasi yang akan dibangun. Pada tahap ini, sistem dirancang agar pengguna dapat memilih satu atau lebih kategori, kemudian memilih subkategori minat yang muncul melalui *pop-up*. Setelah itu, sistem memproses pilihan pengguna untuk menghasilkan rekomendasi video yang relevan.



Gambar 2. Alur Proses Pengguna

Proses dimulai ketika pengguna membuka aplikasi, memilih kategori dan subkategori, lalu menekan tombol lihat rekomendasi. Sistem kemudian memeriksa apakah minimal satu subkategori telah dipilih. Jika valid, sistem melanjutkan proses rekomendasi dan menampilkan hasil. Setelah hasil muncul, pengguna dapat menggunakan *filter* kategori, *filter* subkategori, mode semua, membuka detail video ke YouTube, atau mengakhiri proses..

3.2. Pengumpulan *Dataset*

Tahap ini dilakukan dengan menelusuri video-video yang muncul pada halaman beranda dan menu eksplorasi YouTube sebagai kandidat data awal. Tautan video yang relevan kemudian dicatat, disaring, dan dikelompokkan ke dalam kategori serta subkategori sesuai ruang lingkup penelitian. Setiap tautan video selanjutnya diproses menggunakan program autoFile untuk mengambil metadata melalui YouTube API. Data yang diperoleh disimpan dalam format JSON agar dapat digunakan pada tahap pembentukan *embedding* dan proses rekomendasi.

```

def get_video_metadata(video_url):
    video_id = extract_video_id(video_url)

    if not video_id:
        print(f"Video ID tidak ditemukan: {video_url}")
        return None

    endpoint = "https://www.googleapis.com/youtube/v3/videos"

    params = {
        "part": "snippet,contentDetails,statistics",
        "id": video_id,
        "key": API_KEY
    }

    response = requests.get(endpoint, params=params)
    response.raise_for_status()

    data = response.json()
  
```

Gambar 3. Fungsi Pengambilan *Metadata* Video

Fungsi pengambilan metadata video berdasarkan URL YouTube. Proses diawali dengan mengambil video_id dari tautan video, karena YouTube API membaca data berdasarkan ID video. Setelah itu, sistem mengakses *endpoint* API dengan parameter *snippet*, *contentDetails*, dan *statistics* [17] untuk memperoleh informasi utama dari video.

```

snippet = item.get("snippet", {})
content_details = item.get("contentDetails", {})
statistics = item.get("statistics", {})

description = snippet.get("description", "")

result = {
    "channelName": snippet.get("channelTitle"),
    "duration": format_duration(content_details.get("duration", "PT0S")),
    "date": format_date_to_000z(snippet.get("publishedAt")),
    "hashtag": extract_hashtags(description),
    "likes": int(statistics.get("likeCount", 0)) if "likeCount" in statistics else None,
    "thumbnailUrl": snippet.get("thumbnails", {}).get("high", {}).get("url"),
    "title": snippet.get("title"),
    "viewCount": int(statistics.get("viewCount", 0)) if "viewCount" in statistics else 0,
    "url": f"https://www.youtube.com/watch?v={video_id}",
    "text": description.replace("\n", " ")
}

```

Gambar 4. Penyusunan *Dictionary Metadata*

Proses penyusunan hasil metadata ke dalam bentuk *dictionary*. Data yang disusun meliputi nama *channel*, durasi video, tanggal publikasi, *hashtag*, jumlah suka, *thumbnail*, judul video, jumlah penayangan, tautan video, dan deskripsi. Struktur ini digunakan agar setiap data video memiliki format yang seragam sebelum disimpan ke dalam file JSON [18].

```

def build_combined_text(item):
    title = item.get("title", "")
    description = item.get("text", "")
    hashtags = item.get("hashtag", [])

    if isinstance(hashtags, list):
        hashtags = " ".join(hashtags)

    combined = f"{title} {description} {hashtags}"

    return clean_text(combined)

```

Gambar 5. Fungsi *build_combined_text*

Fungsi *build_combined_text(item)* yang digunakan untuk menggabungkan beberapa *metadata* utama video, yaitu judul, deskripsi, dan *hashtag*, menjadi satu teks utama bernama *combined_text*. Teks gabungan ini kemudian dibersihkan menggunakan fungsi *clean_text()* agar lebih siap diproses pada tahap pembentukan *embedding* SBERT. Dengan cara ini, representasi video tidak hanya bergantung pada judul, tetapi juga memanfaatkan konteks dari deskripsi dan *hashtag*.

3.3. Pembentukan *Embedding* SBERT

Tahap pembentukan *embedding* dilakukan untuk mengubah teks metadata video menjadi representasi numerik. Teks yang digunakan berasal dari *combined_text* yaitu gabungan judul, deskripsi, dan *hashtag* video. Representasi ini digunakan agar sistem dapat memahami hubungan makna antar video sebelum dilakukan perhitungan kemiripan menggunakan *Cosine Similarity*.

```
texts = [item.get("combined_text", "") for item in dataset]
```

```
print("Membuat embedding dataset...")
embeddings = model.encode(
    texts,
    show_progress_bar=True,
    convert_to_numpy=True,
    normalize_embeddings=True
)
```

Gambar 6. Pembentukan *Embedding* SBERT

Bagian pertama mengambil field `combined_text` dari setiap data video sebagai teks utama yang akan diproses. Bagian kedua menggunakan fungsi `model_encode()` pada SBERT untuk mengubah kumpulan teks tersebut menjadi vektor numerik. Parameter `convert_to_numpy=True` digunakan agar hasil *embedding* tersimpan dalam bentuk *array* NumPy, sedangkan `normalize_embeddings=True` digunakan untuk menormalkan vektor agar lebih siap digunakan pada perhitungan *similarity*.

3.4. Perhitungan *Cosine Similarity*

Tahap perhitungan *Cosine Similarity* dilakukan untuk mengukur tingkat kemiripan antara *embedding query* pengguna dan *embedding combined_text* video [19]. Nilai kemiripan ini digunakan sebagai dasar untuk menentukan video yang paling relevan terhadap subkategori minat yang dipilih pengguna.

$$\text{Cos}(\theta) = \frac{A \cdot B}{|A| |B|} \quad (1)$$

Keterangan:

Cos(θ) = nilai kemiripan antara dua vektor

A = *embedding* vektor *query*

B = *embedding* vektor *combined_text* video

|A| = panjang atau *magnitude* vektor A

|B| = panjang atau *magnitude* vektor B

Pada persamaan (1) digunakan untuk membandingkan vektor *query* minat pengguna dengan vektor *combined_text* video. Vektor *query* diperoleh dari subkategori yang dipilih pengguna, sedangkan vektor *combined_text* diperoleh dari teks video yang telah diproses menjadi *embedding* SBERT. Hasil perhitungan berupa nilai *similarity* antara 0 sampai 1, di mana nilai yang semakin mendekati 1 menunjukkan bahwa video tersebut semakin relevan dengan minat pengguna. Nilai inilah yang kemudian digunakan sistem sebagai dasar untuk memilih dan mengurutkan video rekomendasi.

```
scores = cosine_similarity(
    query_embedding,
    filtered_embeddings
)[0]
```

Gambar 7. Perhitungan *Cosine Similarity*

Proses pada sistem menunjukkan perhitungan kemiripan antara `query_embedding` dan `filtered_embeddings` menggunakan fungsi `cosine_similarity()`, selanjutnya `query_embedding` merupakan vektor dari pilihan minat pengguna, sedangkan `filtered_embeddings` berisi

kumpulan vektor video yang telah disesuaikan dengan kategori dan subkategori pilihan, lalu '[0]' digunakan untuk mengambil baris pertama yang berisi seluruh nilai *similarity* antara *query* dan video pada dataset. Hasil fungsi tersebut berupa nilai *similarity* yang kemudian digunakan pada tahap pengurutan rekomendasi.

3.5. Pengurutan *Top-N Recommendation*

Tahap pengurutan *Top-N Recommendation* dilakukan setelah seluruh nilai *similarity* antara *query* pengguna dan *metadata* video diperoleh. Pada tahap ini, sistem mengambil lima video dengan nilai *similarity* tertinggi dari setiap subkategori yang dipilih, kemudian menggabungkan seluruh hasil dan mengurutkannya kembali secara global.

```
ranked_indices = np.argsort(scores)[::-1]

for rank_idx in ranked_indices[:LIMIT_PER_SUBCATEGORY]:
    final_results.sort(
        key=lambda item: item.get("similarity_score_float", 0),
        reverse=True
    )
```

Gambar 8. Pengurutan *Top-N Recommendation*

Proses pengurutan hasil rekomendasi menggunakan `np.argsort(scores)[::-1]`. Fungsi `np.argsort(scores)` digunakan untuk mengurutkan indeks berdasarkan nilai *similarity*, sedangkan `[::-1]` membalik urutan agar nilai tertinggi berada di posisi awal. Setelah itu, sistem mengambil lima video teratas sesuai nilai `LIMIT_PER_SUBCATEGORY`, lalu seluruh hasil dari beberapa subkategori digabungkan dan diurutkan kembali berdasarkan nilai *similarity* tertinggi.

3.6. Evaluasi

Evaluasi sistem dilakukan menggunakan metode *Black Box Testing* [20]. Metode ini digunakan untuk menguji fungsi sistem dari sisi pengguna tanpa melihat struktur kode program. Pengujian difokuskan pada kesesuaian antara masukan pengguna dan keluaran yang dihasilkan oleh sistem.

Skenario pengujian meliputi pemilihan kategori, pemilihan subkategori, fitur pilih semua, tampilan hasil rekomendasi, *filter* kategori, *filter* subkategori, mode semua, tombol kembali, dan akses video menuju YouTube. Evaluasi ini bertujuan untuk memastikan bahwa seluruh fitur utama sistem dapat berjalan sesuai kebutuhan sebelum hasil pengujiannya dibahas pada bagian hasil dan pembahasan.

4. Hasil dan Pembahasan

4.1. Lingkungan Implementasi

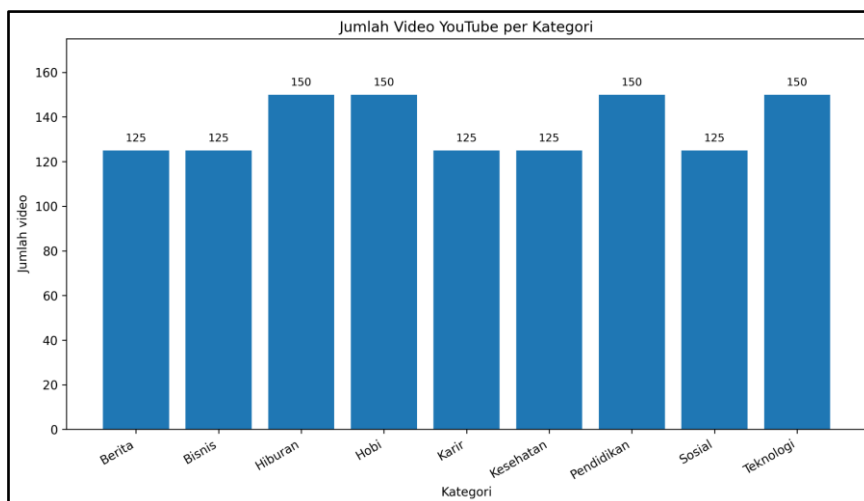
Lingkungan implementasi digunakan untuk menjelaskan perangkat yang dipakai dalam proses pengembangan dan pengujian sistem rekomendasi. Sistem dikembangkan berbasis web menggunakan Python dan *framework* Flask, sedangkan proses rekomendasi memanfaatkan model SBERT untuk pembentukan *embedding* dan *Cosine Similarity* untuk menghitung kemiripan antara minat pengguna dan *metadata* video.

Tabel 2. Lingkungan Implementasi Sistem

No	Komponen	Spesifikasi
1	Sistem Operasi	Windows 11
2	Bahasa Pemrograman	Python
3	<i>Framework</i>	Flask
4	Model <i>Embedding</i>	paraphrase-multilingual-MiniLM-L12-v2
5	Format Dataset	JSON
6	Editor Kode	Visual Studio Code
7	Browser Pengujian	Google Chrome
8	Prosesor	AMD Ryzen 5 5500U with Radeon Graphics
9	RAM	8 GB
10	Penyimpanan	256 GB SSD
11	Resolusi Layar	1920 × 1080 piksel

Berdasarkan Tabel 2, perangkat lunak utama yang digunakan mendukung proses pengolahan data, pembentukan *embedding*, perhitungan *similarity*, dan implementasi sistem berbasis web. Spesifikasi perangkat keras yang digunakan juga sudah mencukupi untuk menjalankan proses pengembangan, penyimpanan *dataset*, serta pengujian sistem rekomendasi secara lokal.

4.2. Dataset Penelitian

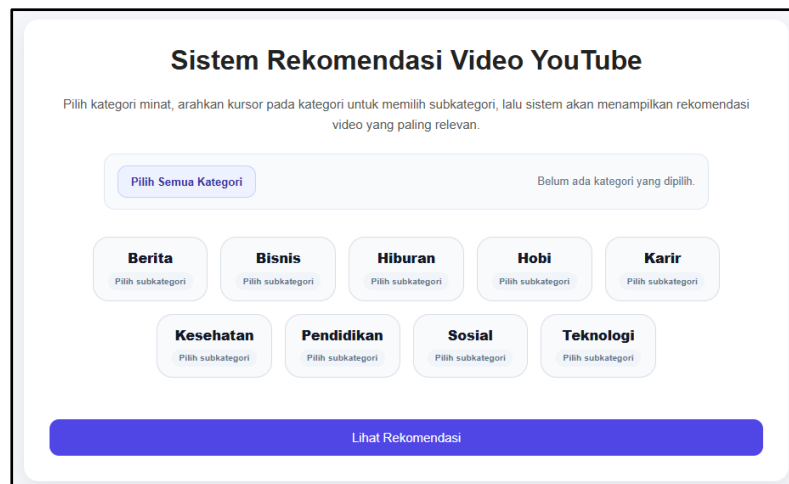


Gambar 9. Jumlah Video YouTube per Kategori

Dataset pada penelitian ini terdiri dari 1.225 video YouTube yang dikelompokkan ke dalam 9 kategori dan 49 subkategori. Berdasarkan Gambar 8, kategori Hiburan, Hobi, Pendidikan, dan Teknologi memiliki masing-masing 150 video karena terdiri dari 6 subkategori. Sementara itu, kategori Berita, Bisnis, Karir, Kesehatan, dan Sosial memiliki masing-masing 125 video karena terdiri dari 5 subkategori. Komposisi ini menunjukkan bahwa dataset disusun secara terstruktur, dengan jumlah video yang seimbang pada setiap subkategori, yaitu 25 video.

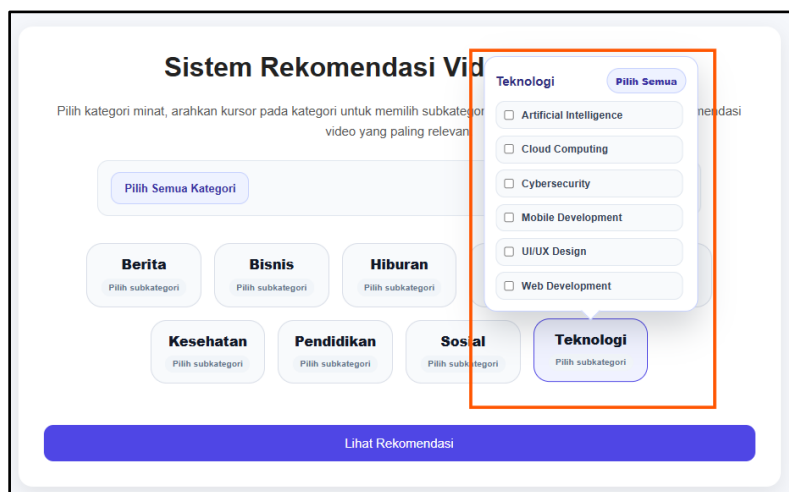
4.3. Implementasi Tampilan

Implementasi tampilan sistem dibuat agar pengguna dapat memilih minat, melihat hasil rekomendasi, dan memfilter video secara sederhana. Interaksi utama pengguna terdiri dari pemilihan kategori, pemilihan subkategori melalui *pop-up*, tampilan hasil rekomendasi, serta penggunaan *filter* pada halaman hasil.



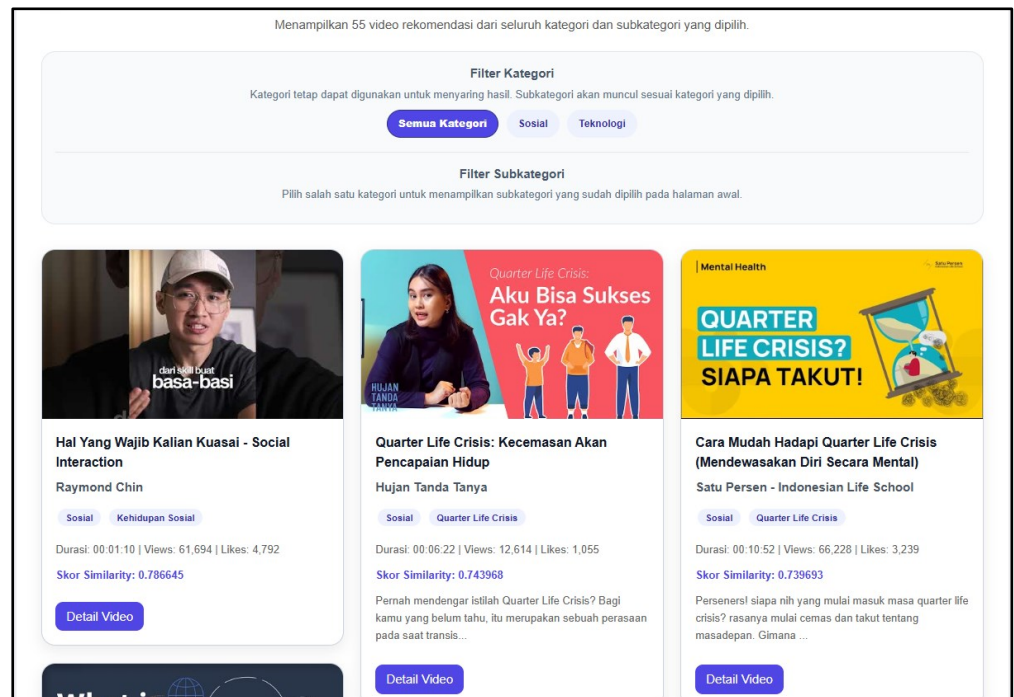
Gambar 10. Tampilan Pemilihan Kategori

Pada halaman ini, pengguna dapat memilih satu atau lebih kategori minat, seperti Berita, Bisnis, Hiburan, Hobi, Karir, Kesehatan, Pendidikan, Sosial, dan Teknologi sebelum menekan tombol lihat rekomendasi.



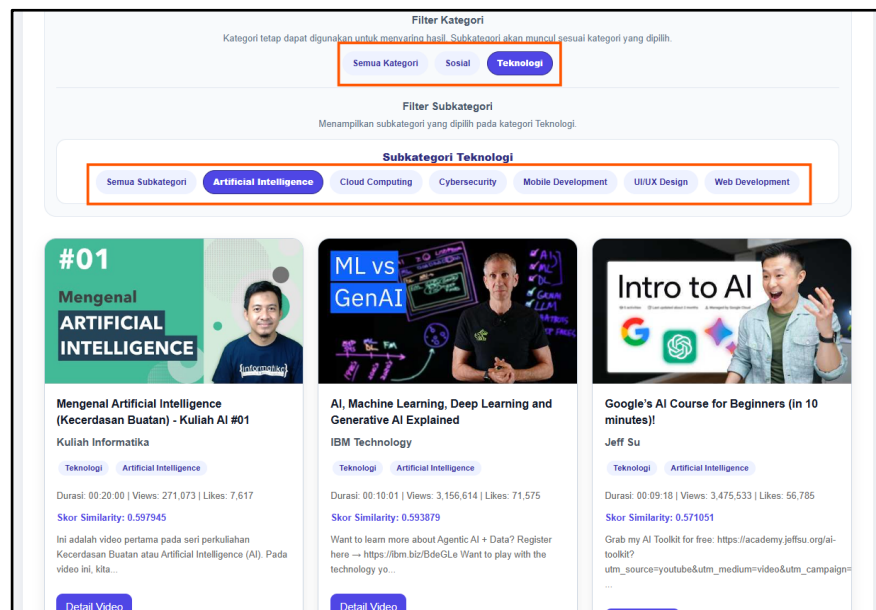
Gambar 11. Tampilan *Pop-Up* Subkategori

Tampilan *pop-up* subkategori yang muncul ketika pengguna mengarahkan mouse ke salah satu kategori misal pada kategori teknologi maka munculnya seperti *artificial intelligence*, *cloud computing*, dll. Melalui tampilan ini, pengguna dapat memilih minat yang lebih spesifik, baik satu subkategori maupun beberapa subkategori dari kategori yang sama atau berbeda.



Gambar 12. Tampilan Hasil Rekomendasi

Halaman hasil rekomendasi setelah pengguna memilih kategori atau subkategori. Sistem menampilkan video dalam bentuk *card* yang memuat *thumbnail*, judul video, nama *channel*, durasi, jumlah penonton, kategori, subkategori, nilai *similarity*, serta tombol untuk membuka video di YouTube.

Gambar 13. Tampilan *Filter* Kategori dan Subkategori

Pengguna dapat menyaring hasil berdasarkan kategori, subkategori, atau mode semua. Fitur ini membantu pengguna melihat rekomendasi secara lebih terarah tanpa perlu kembali ke halaman pemilihan awal.

4.4. Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan seluruh fungsi pada sistem rekomendasi konten video YouTube berjalan sesuai kebutuhan pengguna. Pengujian menggunakan metode *Black Box Testing*, yaitu pengujian yang berfokus pada masukan dan keluaran sistem tanpa melihat struktur kode program. Skenario pengujian mencakup proses pemilihan kategori, subkategori, tampilan rekomendasi, filter, detail video, dan tombol kembali.

4.4.1. *Black Box Testing*

Tabel 3. *Black Box Testing* Sistem Rekomendasi

No	Pengujian	Skenario Pengujian	Hasil yang Diharapkan	Status
1	Pilih satu kategori atau subkategori	Pengguna memilih satu kategori atau satu subkategori pada halaman utama	Sistem menyimpan pilihan pengguna dan menandai pilihan tersebut sebagai aktif	Berhasil
2	Pilih lebih dari satu kategori atau subkategori	Pengguna memilih lebih dari satu kategori atau subkategori pada halaman utama	Sistem menyimpan seluruh pilihan kategori dan subkategori yang dipilih pengguna	Berhasil
3	Pilih semua kategori atau subkategori	Pengguna menekan tombol pilih semua kategori atau pilih semua subkategori	Sistem memilih seluruh kategori atau subkategori sesuai tombol yang digunakan	Berhasil
4	Lihat rekomendasi	Pengguna menekan tombol lihat rekomendasi setelah memilih kategori atau subkategori	Sistem menampilkan halaman hasil rekomendasi sesuai pilihan pengguna	Berhasil
5	Filter kategori	Pengguna memilih salah satu filter kategori pada halaman hasil	Sistem menampilkan hasil rekomendasi sesuai kategori yang dipilih	Berhasil
6	Filter subkategori	Pengguna memilih salah satu filter subkategori pada halaman hasil	Sistem menampilkan hasil rekomendasi sesuai subkategori yang dipilih	Berhasil
7	Filter semua kategori atau subkategori	Pengguna memilih tombol semua kategori atau semua subkategori pada halaman hasil	Sistem menampilkan seluruh hasil rekomendasi sesuai cakupan filter yang dipilih	Berhasil
8	Detail video	Pengguna menekan tombol detail video pada salah satu hasil rekomendasi	Sistem mengarahkan pengguna ke halaman video YouTube	Berhasil
9	Tombol kembali	Pengguna menekan tombol kembali pada halaman hasil rekomendasi	Sistem mengarahkan pengguna kembali ke halaman pemilihan kategori dan subkategori	Berhasil

4.4.2. Hasil Pengujian

Berdasarkan hasil pengujian menggunakan metode *Black Box Testing*, seluruh fitur utama pada sistem rekomendasi berjalan sesuai kebutuhan pengguna. Pengguna dapat memilih satu atau lebih kategori maupun subkategori, menggunakan fitur pilih semua, serta menampilkan hasil rekomendasi berdasarkan pilihan yang telah ditentukan.

Selain itu, fitur *filter* kategori, *filter* subkategori, mode semua, detail video, dan tombol kembali juga berjalan dengan baik. Dengan demikian, sistem rekomendasi yang dikembangkan telah memenuhi kebutuhan fungsional dari sisi pengguna.

4.5. Perbandingan

Perbandingan dilakukan untuk melihat kontribusi penelitian ini terhadap penelitian serupa yang telah dilakukan sebelumnya. Aspek yang dibandingkan meliputi objek penelitian, metode, dasar rekomendasi, cakupan data, representasi teks, dan bentuk hasil rekomendasi.

Tabel 4. Perbandingan dengan Penelitian Terdahulu

Aspek	Ananta dan Supriana (2023) [12]	Witjaksana (2023) [13]	Penelitian Ini
Objek penelitian	Konten video YouTube	Video YouTube topik <i>Information Technology</i>	Konten video YouTube berbagai minat
Metode utama	<i>Content-Based Filtering</i> dan <i>Cosine Similarity</i>	<i>Content-Based Filtering</i> , <i>YouTube API</i> , dan <i>Cosine Similarity</i>	<i>Content-Based Filtering</i> , SBERT, dan <i>Cosine Similarity</i>
Dasar rekomendasi	Durasi menonton pengguna	Input <i>keyword</i> dan <i>metadata</i> video	Pilihan kategori, subkategori, dan <i>metadata</i> video
Representasi teks	Belum menggunakan representasi semantik SBERT	Belum menggunakan representasi semantik SBERT	Menggunakan SBERT untuk membentuk <i>embedding</i> semantik
Cakupan data	Berdasarkan histori aktivitas pengguna	Terbatas pada topik teknologi informasi	1.225 video, 9 kategori, dan 49 subkategori
Bentuk hasil	Rekomendasi berdasarkan kemiripan dan histori tontonan	Rekomendasi video sesuai topik pencarian	<i>Top-N Recommendation</i> berdasarkan nilai <i>similarity</i> tertinggi
Kontribusi utama	Memanfaatkan durasi menonton sebagai indikator minat	Memanfaatkan <i>metadata</i> YouTube melalui <i>API</i>	Menyediakan rekomendasi berbasis minat langsung dengan kategori, subkategori, SBERT, dan <i>filter</i> hasil

Berdasarkan Tabel 3, penelitian ini memiliki kontribusi pada penggunaan struktur minat yang lebih terarah melalui kategori dan subkategori. Selain itu, penggunaan SBERT membuat proses rekomendasi tidak hanya bergantung pada *keyword* atau histori tontonan, tetapi juga mempertimbangkan kedekatan makna antara minat pengguna dan *metadata* video. Dengan demikian, sistem yang dikembangkan memberikan pendekatan yang lebih semantik, terstruktur, dan sesuai dengan pilihan minat pengguna.

5. Kesimpulan

Penelitian ini menghasilkan sistem rekomendasi konten video YouTube berdasarkan minat pengguna menggunakan *Content-Based Filtering*, SBERT, dan *Cosine Similarity*. Sistem memanfaatkan 1.225 video yang dikelompokkan ke dalam 9 kategori dan 49 subkategori, kemudian menampilkan rekomendasi berdasarkan pilihan minat pengguna dengan urutan nilai *similarity* tertinggi. Hasil pengujian *Black Box Testing* menunjukkan bahwa fitur utama, seperti pemilihan kategori dan subkategori, tampilan rekomendasi, *filter* hasil, detail video, dan tombol kembali berjalan sesuai kebutuhan. Dengan demikian, sistem ini dapat membantu pengguna menemukan konten YouTube secara lebih terarah dan relevan. Keterbatasan penelitian ini terdapat pada jumlah dataset yang masih terbatas dan pembaruan data yang belum otomatis, sehingga penelitian selanjutnya dapat memperluas dataset, menambahkan pembaruan melalui *YouTube API*, serta melakukan evaluasi relevansi rekomendasi dari sisi pengguna.

6. Saran

Berdasarkan hasil penelitian dan keterbatasan sistem, pengembangan selanjutnya dapat dilakukan dengan memperluas jumlah video, kategori, dan subkategori agar rekomendasi yang dihasilkan lebih beragam. Dataset juga dapat dibuat lebih dinamis melalui pembaruan *metadata* secara berkala menggunakan *YouTube API*, sehingga sistem tidak hanya bergantung pada dataset statis. Selain itu, sistem dapat dikembangkan ke tahap *deployment* atau *hosting* agar dapat

diakses secara online melalui *browser* oleh pengguna. Setelah sistem stabil, promosi dapat dilakukan melalui iklan digital seperti Meta Ads, TikTok Ads, dan X Ads untuk menjangkau pengguna yang memiliki minat pada edukasi, teknologi, hiburan, bisnis, dan topik digital lainnya.

NV: Neville Venerdi
AF: Ahmad Fitriansyah
JS: Jamah Sari

Kontribusi Penulis: Konseptualisasi: NV, AF, dan JS; Metodologi: NV; Perangkat Lunak: NV; Validasi: NV, AF, dan JS; Analisis formal: NV; Investigasi: NV; Sumber daya: NV; Kurasi data: NV; Penulisan—persiapan draf asli: NV; Penulisan—peninjauan dan penyuntingan: AF dan JS; Visualisasi: NV; Supervisi: AF dan JS; Administrasi proyek: NV; Akuisisi pendanaan: tidak berlaku.

Pendanaan: Penelitian ini tidak menerima pendanaan eksternal.

Pernyataan Ketersediaan Data: Data yang digunakan dalam penelitian ini berupa *metadata* video YouTube dan hasil kuesioner responden. Data pendukung penelitian tersedia dari penulis korespondensi atas permintaan yang wajar, karena sebagian data berkaitan dengan hasil kuesioner responden dan pengelolaan *dataset* internal. Ketersediaan data tetap memperhatikan batasan privasi responden dan kepentingan akademik penelitian.

Ucapan Terima Kasih: Penulis mengucapkan terima kasih kepada Program Studi Teknik Informatika, Institut Teknologi dan Bisnis Swadharma, serta dosen pembimbing yang telah memberikan arahan dan masukan selama proses penelitian. Penulis juga menyampaikan terima kasih kepada responden yang telah berpartisipasi dalam pengisian kuesioner. Dalam proses penyusunan naskah, penulis menggunakan bantuan perangkat AI untuk mendukung perapian bahasa dan struktur penulisan, dengan seluruh isi, analisis, dan tanggung jawab akhir tetap berada pada penulis.

Konflik Kepentingan: Penulis menyatakan tidak ada konflik kepentingan. Karena penelitian ini tidak menerima pendanaan eksternal maka dari itu pendana tidak memiliki peran dalam desain studi; dalam pengumpulan, analisis, atau interpretasi data; dalam penulisan naskah; atau dalam keputusan untuk menerbitkan hasil.

Referensi

- [1] F. Faiqah, M. Nadjib, dan A. S. Amir, "Youtube sebagai sarana komunikasi bagi komunitas Makassarvidgram," *Jurnal Komunikasi KAREBA*, vol. 5, no. 2, 2016. doi: <https://journal.unhas.ac.id/index.php/kareba/article/view/1905>.
- [2] P. Covington, J. Adams, dan E. Sargin, "Deep Neural Networks for YouTube Recommendations," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 191–198, doi: 10.1145/2959100.2959190. doi: <https://doi.org/10.1145/2959100.2959190>.
- [3] C. C. Aggarwal, *Recommender Systems: The Textbook*. Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-29659-3. doi: <https://link.springer.com/book/10.1007/978-3-319-29659-3>.
- [4] F. Ricci, L. Rokach, dan B. Shapira, Eds., *Recommender Systems Handbook*, 3rd ed. Cham, Switzerland: Springer, 2022, doi: 10.1007/978-1-0716-2197-4. doi: <https://link.springer.com/book/10.1007/978-1-0716-2197-4>.
- [5] Y. Wang, W. Ma, M. Zhang, Y. Liu, dan S. Ma, "A Survey on the Fairness of Recommender Systems," *arXiv preprint arXiv:2206.03761*, 2022. doi: <https://arxiv.org/abs/2206.03761>.
- [6] B. Bakiyev, "Method for Determining the Similarity of Text Documents for the Kazakh Language, Taking Into Account Synonyms: Extension to TF-IDF," *arXiv preprint arXiv:2211.12364*, 2022. doi: <https://arxiv.org/abs/2211.12364>.
- [7] C. Thompson, "YouTube's Plot to Silence Conspiracy Theories," *WIRED*, 2020. doi: <https://www.wired.com/story/youtube-algorithm-silence-conspiracy-theories/>.
- [8] N. Sukiennik, C. Gao, dan N. Li, "Uncovering the Deep Filter Bubble: Narrow Exposure in Short-Video Recommendation," *arXiv preprint arXiv:2403.04511*, 2024. doi: <https://arxiv.org/abs/2403.04511>.
- [9] S. Puglisi, J. Parra-Arnau, J. Forné, dan D. Rebollo-Monedero, "On Content-Based Recommendation and User Privacy in Social-Tagging Systems," *arXiv preprint arXiv:1605.06538*, 2016. doi: <https://arxiv.org/abs/1605.06538>.
- [10] N. Reimers dan I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, pp. 3982–3992, doi: 10.18653/v1/D19-1410. doi: <https://aclanthology.org/D19-1410/>.
- [11] H. Steck, C. Ekanadham, dan N. Kallus, "Is Cosine-Similarity of Embeddings Really About Similarity?," *arXiv preprint arXiv:2403.05440*, 2024. doi: <https://arxiv.org/abs/2403.05440>.

- [12] I. G. N. W. Ananta dan I. W. Supriana, "Analisa Sistem Rekomendasi Konten Youtube Berdasarkan Durasi Menonton Menggunakan Content-Based Filtering," *Jurnal Nasional Teknologi Informasi dan Aplikasinya*, vol. 1, no. 3, pp. 901–908, 2023. doi: https://www.researchgate.net/publication/389823778_Analisa_Sistem_Rekomendasi_Konten_Youtube_Berdasarkan_Durasi_Menonton_Menggunakan_Content-Based_Filtering.
- [13] A. B. Witjaksana, "Sistem Rekomendasi Video YouTube Topik Information Technology dengan Metode Content-Based Filtering," Skripsi/Tugas Akhir, Politeknik Negeri Jakarta, 2023. doi: <https://repository.pnj.ac.id/id/eprint/14310/>.
- [14] P. Castells dan D. Jannach, "Recommender Systems: A Primer," *arXiv preprint arXiv:2302.02579*, 2023. doi: <https://arxiv.org/abs/2302.02579>.
- [15] X. Thomas, "Content-Based Personalized Recommender System Using Entity Embeddings," *arXiv preprint arXiv:2010.12798*, 2020. doi: <https://arxiv.org/abs/2010.12798>.
- [16] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, dan W. Wang, "Language-agnostic BERT Sentence Embedding," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022, pp. 878–891, doi: 10.18653/v1/2022.acl-long.62. doi: <https://aclanthology.org/2022.acl-long.62/>.
- [17] Google Developers, "Videos: list," *YouTube Data API v3 Documentation*, 2026. doi: <https://developers.google.com/youtube/v3/docs/videos/list>.
- [18] T. Bray, "The JavaScript Object Notation (JSON) Data Interchange Format," *RFC 8259*, Internet Engineering Task Force, 2017, doi: 10.17487/RFC8259. doi: <https://www.rfc-editor.org/rfc/rfc8259>.
- [19] H. Steck, L. Baltrunas, E. Elahi, D. Liang, Y. Raimond, dan J. Basilico, "Deep Learning for Recommender Systems: A Netflix Case Study," *AI Magazine*, vol. 42, no. 3, pp. 7–18, 2021. doi: <https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/18140>.
- [20] M. S. Mustaqbal, R. F. Firdaus, dan H. Rahmadi, "Pengujian aplikasi menggunakan black box testing," *Jurnal Ilmiah Teknologi Informasi Terapan*, vol. 2, no. 3, 2016. doi: <https://journal.widyatama.ac.id/index.php/jitter/article/view/70>.