

Perancangan Sistem Rekomendasi Laptop dengan Model *Prototyping* dan Penerapan *Content-Based Filtering Approach* pada ELS Computer Shop Semarang

Riandy Oktavian^{1✉}, Fatkhul Amin²

^{1,2} Teknik Informatika, Teknologi Informasi, Universitas Stikubank Semarang, Indonesia

Informasi Artikel

Riwayat Artikel

Diserahkan : 12-11-2023

Direvisi : 21-11-2023

Diterima : 23-11-2023

Kata Kunci:

Sistem rekomendasi,
Content-based filtering, *One hot encoding*, *Ordinal encoding*, *Cosine similarity*

Keywords :

Recommendation system,
Content-based filtering, *One hot encoding*, *Ordinal encoding*, *Cosine similarity*

ABSTRAK

Beragamnya pilihan laptop yang ditawarkan oleh Els Computer Shop Semarang menyebabkan pelanggan kesulitan dalam memilih laptop yang sesuai dengan kebutuhan. Masalah tersebut dapat diatasi dengan sistem rekomendasi yang memberikan saran atau rekomendasi laptop berdasarkan preferensi pengguna. Penelitian ini bertujuan untuk membangun sistem rekomendasi untuk laptop di Els Computer Shop Semarang, menggunakan pendekatan *content-based filtering* yang memudahkan pelanggan dalam memilih laptop sesuai dengan atribut spesifikasi laptop yang dicari. Metode prototipe dipilih sebagai teknik pengembangan sistem karena yang dapat mengatasi permasalahan antara pengguna dan analis. Sistem rekomendasi ini memanfaatkan *one hot encoding* dan *ordinal encoding* untuk mengubah data kategorikal spesifikasi laptop. *One hot encoding* mengubah data menjadi angka biner (0 atau 1), sedangkan *ordinal encoding* mengubah data menjadi angka numerik (0, 1, 2, 3) berdasarkan urutan tertentu. Hasil yang diperoleh dari penelitian ini adalah rekomendasi tiga laptop yang paling mirip dengan indeks yang dicari oleh pengguna berdasarkan perhitungan *cosine similarity* spesifikasi laptop.

ABSTRACT

The variety of laptop choices offered by Els Computer Shop Semarang makes it difficult for customers to choose a laptop that suits their needs. This problem can be overcome with a recommendation system that provides suggestions or laptop recommendations based on user preferences. This research aims to build a recommendation system for laptops at Els Computer Shop Semarang, using a *content-based filtering approach* that makes it easier for customers to choose laptops according to the laptop specification attributes they are looking for. The prototype method was chosen as a system development technique because it can overcome problems between users and analysts. This recommendation system utilizes *one hot encoding* and *ordinal encoding* to change categorical data on laptop specifications. *One hot encoding* changes data into binary numbers (0 or 1), while *ordinal encoding* changes data into numeric numbers (0, 1, 2, 3) based on a certain order. The results obtained from this research are recommendations for three laptops that are most similar to the index sought by the user based on the *cosine similarity* calculation of the laptop specifications.

Corresponding Author :

Riandy Oktavian

Teknik Informatika, Teknologi Informasi, Universitas Stikubank Semarang, Indonesia

Jl. Tri Lomba Juang, Mugassari, Kec. Semarang Sel., Kota Semarang, Jawa Tengah 50241

Email: riandyokta09@gmail.com

PENDAHULUAN

ELS Computer Shop didirikan pada 1997 di Yogyakarta merupakan toko laptop dan komputer terpercaya dengan cabang di Solo, Purwokerto, dan Semarang. Toko ini menawarkan beragam spesifikasi laptop, seperti, merek, tipe prosessor, ukuran layar, dan harga dan sebagainya. Banyaknya pilihan yang tersedia membuat orang kesulitan menentukan laptop yang sesuai dengan kebutuhan mereka. Melihat masalah tersebut maka dibutuhkan solusi berupa sistem rekomendasi yang dapat membantu memberikan saran dan rekomendasi laptop yang sesuai dengan preferensi dan kebutuhan mereka.

Sistem rekomendasi memberikan rekomendasi dengan mempertimbangkan preferensi pengguna dan mengintegrasikan informasi kontekstual. Jumlah informasi yang tersedia menunjukkan kebutuhan untuk memproses informasi yang tidak relevan (Kanmani & Surendiran, 2020). Rekomendasi yang diberikan dapat beragam seperti rekomendasi musik, trek, berita, dan sebagainya. Pendekatan yang paling umum untuk menerapkan sistem rekomendasi adalah *Content-based Filtering*, *Collaborative Filtering* dan *Hybrid Filtering* (Sharma et al., 2021). *Content-based filtering* merupakan salah satu metode dari sistem rekomendasi yang didasarkan pada atribut, karakteristik, atau fitur yang dimiliki oleh sebuah item (Nasiti, 2019). Pembuatan sistem rekomendasi memiliki beberapa tujuan dan salah satunya adalah untuk memberikan informasi sebanyak mungkin untuk pembelajaran dan preferensi yang dipersonalisasi berdasarkan pola interaksi pengguna (Kausar, 2020).

Berdasarkan penelitian yang dilakukan (Pramestia & Santiyasa, 2022) berhasil membuat sistem rekomendasi video game yang menggunakan data sekunder dari kaggle.com yang terdiri dari 16.026 rekaman video game dengan tiga karakteristik utama yaitu, ID video game (numerik), nama video game (string), dan genre video game (string). Sistem ini mencapai tingkat akurasi rata-rata sebesar 87.75% dan beroperasi secara efisien. Penelitian lain dilakukan oleh (Muliawan et al., 2022) berhasil membuat sistem rekomendasi dalam bentuk aplikasi website yang memberikan rekomendasi hotel berdasarkan preferensi pengguna. Rekomendasi hotel disusun berdasarkan kesamaan dengan data pengguna, dengan kemampuan penambahan data secara otomatis. Bobot permintaan memiliki nilai validitas 0,002, dan hasil pengujian menunjukkan akurasi rata-rata kesamaan sebesar 84,50%.

Sedangkan pada penelitian ini tahun 2023, dengan judul Perancangan Sistem Rekomendasi Laptop dengan model Prototyping dan penerapan Content-Based Filtering Approach pada Els Computer Shop Semarang, menggunakan pendekatan *content-based filtering* untuk memberikan rekomendasi laptop berdasarkan atribut atau fitur dari spesifikasi laptop. Sistem rekomendasi ini memanfaatkan dua jenis *encoding data*, yaitu, *one hot encoding* yang merupakan sebuah teknik dalam merubah bentuk data kategorik menjadi bentuk biner, dengan panjang sesuai dengan banyaknya kategori data yang berbeda (Potdar et al., 2017) dan *ordinal encoding* untuk merubah fitur kategorikal yang memiliki hubungan tingkat atau urutan yang berarti (Roy, 2020). Penggunaan kedua *encoding data* ini bertujuan mempermudah perhitungan kesamaan laptop melalui cosine similarity. Tujuan penelitian ini adalah untuk menghasilkan sistem rekomendasi laptop menggunakan *content-based filtering* yang dapat memberikan rekomendasi tiga laptop paling mirip dengan indeks laptop yang dicari, untuk memberikan layanan optimal kepada pelanggan di Els Computer Shop Semarang.

METODE PENELITIAN

Alur Kerja Penelitian

Penelitian sistem rekomendasi laptop Els Computer Shop dengan menggunakan *content-based filtering* ini dilakukan secara bertahap sesuai dengan alur yang ada pada gambar 1 berikut ini



Gambar 1 Alur Penelitian

a. Analisa Masalah

Els Computer Shop Semarang menghadirkan beragam laptop dengan spesifikasi yang berbeda yang membuat pembeli bingung dalam memilih laptop karena kurangnya pengetahuan tentang kriteria pemilihan laptop sesuai kebutuhan menjadi kendala utama. Sebagai contoh, ada seorang gamer perlu memperhatikan ram, rom dan prosessor. Solusi untuk mengatasi masalah ini, dilakukan penelitian dengan mengembangkan sistem rekomendasi laptop menggunakan pendekatan *content-based filtering*.

b. Rumusan Masalah

Berdasarkan analisa masalah yang diuraikan pada poin sebelumnya, maka solusi yang dibutuhkan adalah suatu sistem rekomendasi laptop menggunakan *content-based filtering* yang membantu memberikan rekomendasi laptop berdasarkan kemiripan dengan atribut laptop yang sedang dicari oleh pengguna sistem. Nilai kemiripan didapatkan melalui proses perhitungan *cosine similarity* yang sebelumnya data diubah terlebih dahulu menggunakan *one hot encoding* dan *ordinal encoding*.

c. Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini adalah dengan *web scrapping* data laptop dari website els computer shop dan studi pustaka untuk mengambil informasi melalui jurnal-jurnal tentang sistem rekomendasi. *Web scrapping* adalah teknik pengumpulan informasi dari situs web dengan cara manual atau otomatis, fokus pada pengambilan dan ekstraksi data. Tujuan *web scrapping* adalah untuk menyaring informasi sehingga memudahkan pencarian data dengan berbagai ukuran (Boeing, 2017).

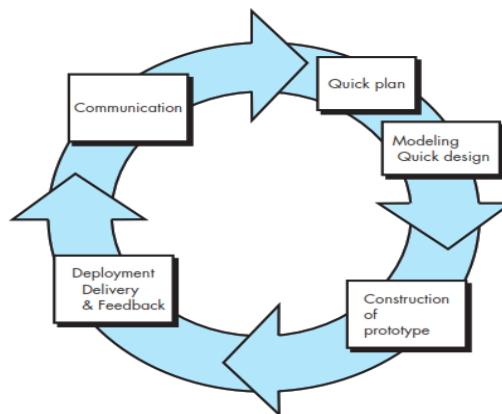
d. Implementasi Content-Based Filtering pada Sistem Rekomendasi

Berikut ini merupakan tahap-tahap implementasi *content-based filtering* untuk sistem rekomendasi laptop di els computer shop semarang :

- 1) Mengumpulkan data laptop dari website els computer shop dengan cara *web scrapping* yang kemudian dibuat dataset untuk digunakan di sistem rekomendasi laptop *content-based filtering*.
- 2) Dataset untuk sistem rekomendasi dengan *content-based filtering* diisi dengan atribut atau fitur penting yang ada pada laptop seperti Layar(inch), RAM(GB), ROM(GB), nama_cpu, dan sebagainya yang dibuat dalam bentuk file excel.
- 3) Membuat sistem rekomendasi menggunakan bahasa pemrograman python dengan *environment* aplikasi *google colab* serta menggunakan dua *encoding data* (*one hot encoding* dan *ordinal encoding*) serta *cosine similarity*.
- 4) Menginput dataset excel kedalam sistem rekomendasi *google colab*.
- 5) Dataset diproses dengan dua *jenis encoding data* (*one hot encoding* dan *ordinal encoding*). *One hot encoding* mengubah data kategorik laptop (Merk, type, dan merk_cpu) menjadi angka biner yaitu 0 dan 1 agar dapat dihitung oleh program. Nilai 1 menunjukkan kategori atau fitur yang benar-benar ada sedangkan nilai 0 menunjukkan kategori lainnya. *Ordinal encoding* mengubah data laptop berdasarkan urutan atau tingkatan tertentu dengan menjadi angka dan diberi nilai mulai dari 0, 1, 2, dan 3. Data yang diubah dengan *ordinal encoding* adalah cpu_gen, RAM (GB), ROM (GB) dan Layar (inch). Hasil dari data yang telah diubah dengan *one hot encoding* dan *ordinal encoding* kemudian dihitung dengan *cosine similarity* untuk menentukan nilai kesamaan antara laptop yang dibandingkan.
- 6) Sistem rekomendasi menampilkan tiga laptop yang paling mirip dengan indeks laptop yang dicari berdasarkan perhitungan *cosine similarity*

Metode Pengembangan Sistem

Metode yang digunakan oleh peneliti dalam pengembangan sistem rekomendasi ini adalah dengan menggunakan metode *prototype*. metode *prototype* memungkinkan adanya interaksi antara pengembang sistem dengan pengguna sistem, sehingga dapat mengatasi resiko ketidakserasian antara pengembang dan pengguna. Metode *prototype* oleh (Pressman, 2015) dijelaskan kedalam beberapa tahapan sebagai berikut.



Gambar 2. Tahapan Metode Prototype (Pressman, 2015)

1) *Communication* (Komunikasi)

Langkah pertama dalam tahap *prototyping* disini adalah komunikasi dimana memerlukan identifikasi permasalahan yang biasa dihadapi oleh pembeli saat menentukan pilihan ketika membeli laptop dan informasi lain yang berkaitan agar dapat membantu pengimplementasian sistem rekomendasi *content-based filtering* laptop di Els Computer Shop Semarang.

2) *Quick Plan* (Perencanaan Cepat)

Langkah berikutnya dalam metode *prototyping* adalah melakukan perencanaan cepat mengenai kebutuhan sumber daya apa saja yang dibutuhkan dalam pembuatan

sistem rekomendasi laptop dengan *content-based filtering* ini. Contoh sumber daya yang digunakan misalnya pada bagian *hardware* yaitu perlunya laptop yang memadai untuk pembuatan sistem sedangkan untuk kebutuhan *software* misalnya perlu penggunaan *google colab* dalam pengkodean dan menjalankan sistem rekomendasi yang dibuat.

3) **Modelling Quick Design (Melakukan Pemodelan)**

Tahap pemodelan disini merancang sistem rekomendasi *content-based filtering* laptop yang disesuaikan dengan kebutuhan pengguna. Contohnya memasukkan atribut atau fitur spesifikasi laptop pilihan pengguna seperti Layar(inch), RAM(GB), ROM(GB), dan sebagainya.

4) **Constuction of prototype (Melakukan Konstruksi)**

Tahap ini mengimplementasikan *content-based filtering* untuk melakukan rekomendasi laptop pada els computer shop dan menggunakan *encoding data* pada data kategorik spesifikasi laptop. *Encoding data* yang pertama adalah *one hot encoding* data kategorik spesifikasi laptop yaitu Merk, type, dan merk_cpu untuk diubah menjadi angka biner yaitu 0 dan 1 agar dapat dihitung oleh program. *Encoding data* yang kedua adalah *ordinal encoding* untuk mengubah data laptop berdasarkan urutan atau tingkatan tertentu dengan mengubah data menjadi angka dan diberi nilai mulai dari 0, 1, 2, dan 3. Data yang diolah dengan *ordinal encoding* adalah cpu_gen, RAM(GB), ROM(GB) dan Layar(inch). Hasil dari *one hot encoding* dan *ordinal encoding* akan dihitung dengan *cosine similarity* untuk memberikan tiga rekomendasi laptop yang mirip dengan yang dicari pengguna.

5) **Penyerahan Hasil**

Langkah terakhir yaitu penyerahan hasil sistem rekomendasi serta menunggu mendapatkan feedback (umpan balik) dari pengguna sistem sebagai hasil dari evaluasi untuk pengimplementasian penggunaan *content-based filtering* pada sistem rekomendasi laptop els computer shop semarang.

Metode Pengumpulan Data

Metode pengumpulan data penelitian ini dengan cara mengambil data laptop dari website resmi Els Computer Shop dengan cara *web scrapping*. *Web scrapping* adalah teknik untuk mendapatkan informasi dari situs tertentu secara manual yang dilakukan dengan cara menyalin informasi secara manual maupun otomatis. *Web scraping* berfokus pada mendapatkan data yang dilakukan dengan cara pengambilan dan ekstraksi. Manfaatnya agar informasi yang telah disalin dapat disaring sehingga mempermudah melakukan pencarian suatu data dengan ukuran yang bervariasi (Boeing, 2017).

HASIL DAN PEMBAHASAN

Sistem rekomendasi *content-based filtering* laptop ini dikembangkan menggunakan bahasa pemrograman python serta memanfaatkan *platform google colab*. Google Colab, singkatnya "Colab", adalah produk dari Google Research yang memungkinkan pengguna untuk menulis dan mengeksekusi kode Python melalui *browser*. Colab sangat cocok untuk keperluan *machine learning*, analisis data, dan pendidikan. Berikut ini merupakan penjelasan beberapa proses pada sistem rekomendasi *content-based filtering* di Els Computer Shop Semarang :

1) Menyiapkan dan menginput dataset ke dalam sistem rekomendasi di *google colab*

Tabel 1. Dataset laptop yang diinput ke sistem rekomendasi

index	no	Merk	Nama_Produk	Harga(IDR)	type	Layar (inch)	merk_cpu	nama_cpu	Ram (GB)	Rom (GB)	Berat (KG)
0	1	Asus	Asus Vivobook 14 VIPS322	6,249,000	notebook	14	Intel	Core i3 1115G4	4	256	1.55
1	2	Asus	Asus VivoBook 14X OLED554	12,999,000	notebook	14	Intel	Core i5 12500H	16	512	1.6
2	3	Asus	Asus VivoBook VIPS553	9,299,000	notebook	16	AMD	Ryzen 5 5600H	16	512	1.6
3	4	Asus	Asus ROG Flow X13 R7R5A6T	21,999,000	gaming	13.4	AMD	Ryzen 7 6800HS	16	1000	1.3
4	5	Lenovo	Lenovo IdeaPad Slim 5FID	6,499,000	notebook	14	AMD	Ryzen 3 7320U	8	512	1.4
5	6	Lenovo	Lenovo Gaming 3 JIID	16,499,000	gaming	15.6	AMD	Ryzen 7 7735HS	8	512	2.32
6	7	Lenovo	Lenovo IdeaPad Slim 3 5MID	9,649,000	notebook	14	Intel	Core i5 1235U	8	512	1.5
7	8	Lenovo	Lenovo Yoga 7 1KID	20,299,000	notebook	14	Intel	Core i7 1360P	16	1000	1.49
8	9	Hp	HP Pavilion 14 EK1005TU	13,449,000	notebook	14	Intel	Core i5 1335U	16	512	1.51
9	10	Hp	HP 14-EM0016AU	8,449,000	notebook	14	AMD	Ryzen 5 7520U	8	512	1.4
10	11	Hp	HP 14S-FQ2038AU	10,199,000	notebook	14	AMD	Ryzen 7 5825U	16	512	1.46
11	12	Hp	HP Victus Gaming 15-FA0008TX	14,999,000	gaming	15.6	Intel	Core i7 12700H	8	512	2.29
12	13	MSI	MSI Modern 14 08IID	6,799,000	notebook	14	Intel	Core i3 1115G4	8	512	1.4
13	14	MSI	MSI Modern 14 033ID	8,999,000	notebook	14	AMD	Ryzen 7 5825U	16	512	1.4
14	15	MSI	MSI Bravo 15 C7VF-026ID	18,999,000	gaming	15.6	AMD	Ryzen 7 7735HS	16	512	1.96
15	16	MSI	MSI Stealth GS 15-A13VF-049ID	29,999,000	gaming	15.6	Intel	Core i7 13620H	16	1000	2.1
16	17	Acer	Acer Aspire 3 R24X	6,799,000	notebook	14	AMD	Ryzen 3 7320U	8	512	1.4
17	18	Acer	Acer Swift GO R72E	11,199,000	notebook	14	AMD	Ryzen 5 7530U	16	512	1.2
18	19	Acer	Acer Aspire 57Q8	8,999,000	notebook	14	Intel	Core i5 1335U	8	512	4.5
19	20	Acer	Acer Gaming Nitro 5 710Q	18,699,000	gaming	15.6	Intel	Core i7 12650H	16	512	2.4

Tabel 1 menunjukkan dataset yang berisikan data spesifikasi laptop seperti Merk, Nama_Produk, Harga, type, Layar(inch), merk_cpu, nama_cpu, RAM(GB), ROM(GB), dan Berat(KG) laptop yang diinput ke dalam sistem rekomendasi laptop dengan *content-based filtering* pada *google colab*.

2) Membuat sebuah list pada data kategorik yang ada dalam dataset

```
# Coding untuk menyimpan data kategorik laptop dalam bentuk daftar/list
cols = ['Merk', 'type', 'merk_cpu']
```

Gambar 3. Perintah cols untuk menyimpan data dalam bentuk list

Perintah **cols** pada Gambar 3 digunakan untuk membuat daftar atau list dari data kategorikal spesifikasi laptop. Terdapat tiga kategorikal data yaitu, Merk, type, dan merk_cpu. Data-data ini disimpan dalam list untuk mempermudah proses *one hot encoding*.

3) Melakukan *one hot encoding* pada data kategorikal yang ada dalam list

Tabel 2. Hasil *one hot encoding*

index	Merk_Acer	Merk_Asus	Merk_Hp	Merk_Lenovo	Merk_MSI	type_gaming	type_notebook	merk_cpu_AMD	merk_cpu_intel
0	0	1	0	0	0	0	1	0	1
1	0	1	0	0	0	0	1	0	1
2	0	1	0	0	0	0	1	1	0
3	0	1	0	0	0	1	0	1	0
4	0	0	0	0	0	0	1	1	0
5	0	0	0	0	0	1	0	1	0
6	0	0	0	1	0	0	1	0	1
7	0	0	0	1	0	0	1	0	1
8	0	0	0	1	0	0	1	0	1
9	0	0	0	1	0	0	1	1	0
10	0	0	1	0	0	0	1	1	0
11	0	0	1	0	0	1	0	0	1
12	0	0	1	0	1	0	1	0	1
13	0	0	1	0	1	0	1	1	0
14	0	0	0	0	1	1	0	1	0
15	0	0	0	0	1	1	0	0	1
16	1	0	0	0	0	0	1	1	0
17	1	0	0	0	0	0	1	1	0
18	1	0	0	0	0	0	1	0	1
19	1	0	0	0	0	1	0	0	1

Tabel 2 menunjukkan pengubahan data dengan *one hot encoding* pada data kategorikal spesifikasi laptop yang ada list yaitu, merk, tipe, dan merk_cpu. Proses ini mengubah data menjadi angka biner (0 atau 1). Angka 0 menandakan keberadaan kategori lainnya dan angka 1 menandakan keberadaan kategori tersebut.

4) Membuat dua kolom baru dari pemisahan kolom nama_cpu

```

8 cols_split = data['nama_cpu'].str.split(" ")
  model_arr = []
  gen_arr = []

  for row in cols_split:
      # Split Core rank by number
      row[1] = ''.join(x for x in row[1] if x.isdigit())

      # Append cpu features to array
      model_arr.append(row[1])
      gen_arr.append(row[2])

  data['cpu_model'] = model_arr
  data['cpu_gen'] = gen_arr

```

Gambar 4. Coding untuk memisahkan kolom nama_cpu

Gambar 4 menunjukkan coding yang berfungsi untuk membersihkan kolom nama_cpu dengan cara memisahkan datanya menjadi dua kolom baru yaitu cpu_model dan cpu gen agar nantinya menjadi lebih mudah diolah. Hasil pemisahannya sebagai berikut.

Tabel 3. Hasil pemisahan kolom nama_cpu dan dibuat menjadi dua kolom baru

index	no	Nama_Produk	nama_cpu	cpu_model	cpu_gen
0	1	Asus Vivobook 14 VIPS322	Core i3 1115G4	3	1115G4
1	2	Asus VivoBook 14X OLED5554	Core i5 12500H	5	12500H
2	3	Asus VivoBook VIPS553	Ryzen 5 5600H	5	5600H
3	4	Asus ROG Flow X13 R7R5A6T	Ryzen 7 6800HS	7	6800HS
4	5	Lenovo IdeaPad Slim 5FID	Ryzen 3 7320U	3	7320U
5	6	Lenovo Gaming 3 JLIID	Ryzen 7 7735HS	7	7735HS
6	7	Lenovo IdeaPad Slim 3 5MID	Core i5 1235U	5	1235U
7	8	Lenovo Yoga 7 1KID	Core i7 1360P	7	1360P
8	9	HP Pavilion 14 EK1005TU	Core i5 1335U	5	1335U
9	10	HP 14-EM0016AU	Ryzen 5 7520U	5	7520U
10	11	HP 14S-FQ2038AU	Ryzen 7 5825U	7	5825U
11	12	HP Victus Gaming 15-FA0008TX	Core i7 12700H	7	12700H
12	13	MSI Modern 14 081ID	Core i3 1115G4	3	1115G4
13	14	MSI Modern 14 033ID	Ryzen 7 5825U	7	5825U
14	15	MSI Bravo 15 C7VF-026ID	Ryzen 7 7735HS	7	7735HS
15	16	MSI Stealth GS 15-A13VF-049ID	Core i7 13620H	7	13620H
16	17	Acer Aspire 3 R24X	Ryzen 3 7320U	3	7320U
17	18	Acer Swift GO R72E	Ryzen 5 7530U	5	7530U
18	19	Acer Aspire 57Q8	Core i5 1335U	5	1335U
19	20	Acer Gaming Nitro 5 710Q	Core i7 12650H	7	12650H

5) Mengubah mengubah kolom cpu_gen dari kode generasi menjadi tahun diproduksinya generasi tersebut

```

for row in data.iterrows():
    data_row = row[1]
    gen = 0

    if data_row['merk_cpu_intel'] == 1:
        if data_row['cpu_gen'][:2] == '13':
            gen = 2023
        elif data_row['cpu_gen'][:2] == '12':
            gen = 2022
        elif data_row['cpu_gen'][:2] == '11':
            gen = 2021

    elif data_row['merk_cpu_AMD'] == 1:
        if data_row['cpu_gen'][:1] == '7':
            gen = 2023
        elif data_row['cpu_gen'][:1] == '6':
            gen = 2022
        elif data_row['cpu_gen'][:1] == '5':
            gen = 2021

    data.loc[row[0], 'cpu_gen'] = gen

```

Gambar 5. Coding untuk mengubah kolom cpu_gen

Gambar 5 merupakan coding untuk mengubah kolom cpu_gen menjadi tahun produksi generasi tersebut. Perubahan untuk merk_cpu_intel maka kolom cpu_gen yang diawali angka 13 diubah menjadi 2023, 12 menjadi 2022, dan 11 menjadi 2021. Sementara untuk merk_cpu_AMD, kolom cpu_gen yang diawali angka 7 diubah menjadi 2023, 6 menjadi 2022, dan 5 menjadi 2021. Pengubahan ini bertujuan mempermudah proses *ordinal encoding*.

6) Hasil perubahan data pada kolom `cpu_gen` sesuai dengan tahun produksinyaTabel 4. Hasil perubahan pada kolom `cpu_gen`

index	no	Nama_Produk	nama_cpu	cpu_model	cpu_gen
0	1	Asus Vivobook 14 VIPS322	Core i3 1115G4	3	2021
1	2	Asus VivoBook 14X OLED554	Core i5 12500H	5	2022
2	3	Asus VivoBook VIPS553	Ryzen 5 5600H	5	2021
3	4	Asus ROG Flow X13 R7R5A6T	Ryzen 7 6800HS	7	2022
4	5	Lenovo IdeaPad Slim 5FID	Ryzen 3 7320U	3	2023
5	6	Lenovo Gaming 3 JLIID	Ryzen 7 7735HS	7	2023
6	7	Lenovo IdeaPad Slim 3 5MID	Core i5 1235U	5	2022
7	8	Lenovo Yoga 7 1KID	Core i7 1360P	7	2023
8	9	HP Pavilion 14 EK1005TU	Core i5 1335U	5	2023
9	10	HP 14-EM0016AU	Ryzen 5 7520U	5	2023
10	11	HP 14S-FQ2038AU	Ryzen 7 5825U	7	2021
11	12	HP Victus Gaming 15-FA0008TX	Core i7 12700H	7	2022
12	13	MSI Modern 14 081ID	Core i3 1115G4	3	2021
13	14	MSI Modern 14 033ID	Ryzen 7 5825U	7	2021
14	15	MSI Bravo 15 C7VF-026ID	Ryzen 7 7735HS	7	2023
15	16	MSI Stealth GS 15-A13VF-049ID	Core i7 13620H	7	2023
16	17	Acer Aspire 3 R24X	Ryzen 3 7320U	3	2023
17	18	Acer Swift GO R72E	Ryzen 5 7530U	5	2023
18	19	Acer Aspire 57Q8	Core i5 1335U	5	2023
19	20	Acer Gaming Nitro 5 710Q	Core i7 12650H	7	2022

Tabel 4 menunjukkan hasil dari perubahan data pada kolom `cpu_gen` yang sudah berubah menjadi tahun produksi cpu generasi tersebut.

7) Melakukan *ordinal encoding* pada kolom `cpu_gen`, RAM(GB), ROM(GB), Layar(inch)

```
# Coding untuk melakukan proses (OrdinalEncoding) pada kolom 'cpu_gen', 'RAM (GB)', 'ROM (GB)', 'Layar(inch)'
# Data diubah menjadi angka 0, 1, 2, 3 berdasarkan tingkat atau urutan datanya
# Contoh pada kolom 'Layar(inch)', 13 inch GB menjadi 0, 14 inch menjadi 1, 15 inch menjadi 2, 16 inch menjadi 3

data['cpu_gen'] = encoder.fit_transform(data[['cpu_gen']])
data['ROM(GB)'] = encoder.fit_transform(data[['ROM(GB)']])
data['RAM(GB)'] = encoder.fit_transform(data[['RAM(GB)']])
data['Layar(inch)'] = encoder.fit_transform(data[['Layar(inch)']])
```

Gambar 6 coding untuk melakukan *ordinal encoding*

Gambar 6 menunjukkan coding untuk melakukan proses *ordinal encoding* pada kolom `cpu_gen`, RAM(GB), ROM(GB), Layar(inch). Data diubah menjadi angka dan diberi nilai mulai dari 0, 1, 2, 3 berdasarkan tingkatan atau urutan tertentu. Perubahan data dengan *ordinal encoding* pada kolom ukuran_layar(inch), 13 inch menjadi 0, 14 inch menjadi 1, 15 inch menjadi 2, 16 inch menjadi 3. Kolom RAM(GB), 4 GB menjadi 0, 8 GB menjadi 1 dan 16 GB menjadi 2. Kolom ROM(GB), 256 GB menjadi 0, 512 GB menjadi 1, dan 1000 GB menjadi 2. Kolom `cpu_gen`, tahun 2021 menjadi 0, 2022 menjadi 1, 2023 menjadi 2. Hasil dari *ordinal encoding* adalah sebagai berikut.

Tabel 5. Hasil ordinal encoding

index	Nama_Produk	Layar(inch)	RAM(GB)	ROM(GB)	cpu_gen
0	Asus Vivobook 14 VIPS322	1	0	0	0
1	Asus VivoBook 14X OLEDS554	1	2	1	1
2	Asus VivoBook VIPS553	3	2	1	0
3	Asus ROG Flow X13 R7R5A6T	0	2	2	1
4	Lenovo IdeaPad Slim 5FID	1	1	1	2
5	Lenovo Gaming 3 JIID	2	1	1	2
6	Lenovo IdeaPad Slim 3 5MID	1	1	1	1
7	Lenovo Yoga 7 1KID	1	2	2	2
8	HP Pavilion 14 EK1005TU	1	2	1	2
9	HP 14-EM0016AU	1	1	1	2
10	HP 14S-FQ2038AU	1	2	1	0
11	HP Victus Gaming 15-FA0008TX	2	1	1	1
12	MSI Modern 14 081ID	1	1	1	0
13	MSI Modern 14 033ID	1	2	1	0
14	MSI Bravo 15 C7VF-026ID	2	2	1	2
15	MSI Stealth GS 15-A13VF-049ID	2	2	2	2
16	Acer Aspire 3 R24X	1	1	1	2
17	Acer Swift GO R72E	1	2	1	2
18	Acer Aspire 57Q8	1	1	1	2
19	Acer Gaming Nitro 5 710Q	2	2	1	1

8) Mengisi nama laptop yang ingin dicari kemiripannya pada bagian laptop_user_like serta menyesuaikan indexnya dengan data pada tabel 4

```
laptop_user_like = "MSI Stealth GS 15-A13VF-049ID"

def get_index_from_title(title):
    return input[input.iloc[:20, 2:] == title].index[15]

laptop_index = get_index_from_title(laptop_user_like)
laptop_index

from sklearn.metrics.pairwise import cosine_similarity
cossim = cosine_similarity(input)

similar_laptop = list(enumerate(cossim[laptop_index]))
sorted_similar_laptop = sorted(similar_laptop, key=lambda i: i[1], reverse=True)

sorted_similar_laptop
```

Gambar 7. Coding untuk menghitung kemiripan laptop yang dicari menggunakan cosine similarity

Gambar 7 menunjukkan coding untuk menginput laptop mana yang akan dicari kesamaannya berdasarkan nilai *cosine similarity*nya. Sebagai contoh dan acuan disini menggunakan index ke-15 yaitu laptop "MSI Stealth GS 15-A13VF-049ID" serta hasilnya akan ditampilkan dalam list dari yang atas ke bawah mulai dari yang paling mirip sampai yang paling tidak mirip berdasarkan nomor index laptopnya seperti berikut.

```
[ (15, 1.0000000000000002),
  (14, 0.9789349735678727),
  (19, 0.9713557884402653),
  (11, 0.9648713100575319),
  (7, 0.9621910808441693),
  (5, 0.9573523347386046),
  (8, 0.9475003092807003),
  (6, 0.9467866713911707),
  (1, 0.9464835831561881),
  (3, 0.9310583061386655),
  (9, 0.926686368629258),
  (17, 0.9266439810085599),
  (13, 0.9248522640555285),
  (12, 0.9114287296555365),
  (10, 0.9102864080301186),
  (2, 0.8865716845205518),
  (18, 0.876669980802134),
  (4, 0.8711981110162718),
  (16, 0.8711981110162718),
  (0, 0.8161148942624818) ]
```

Gambar 8. Hasil *cosine similarity* laptop index 15 dengan laptop lain

- 9) Hasil dari sistem rekomendasi laptop yang menggunakan *cosine similarity* memberikan rekomendasi tiga laptop yang paling mirip dengan laptop index ke-15 yaitu laptop "MSI Stealth GS 15-A13VF-049ID"

```
def get_recs(title, k=3):
    i_d = []

    # Get index from laptop
    indices = pd.Series(data.index, index=data['Nama_Produk']).drop_duplicates()
    laptop_index = indices[title]

    # Get similarity
    similar_laptop = list(enumerate(cossim[laptop_index]))
    similar_laptop = sorted(similar_laptop, key=lambda i: i[1], reverse=True)
    similar_laptop = similar_laptop[1 : k+1]

    # Get index of other similar, recommend the name
    idn = [i[0] for i in similar_laptop]
    final_recs = data_laptop.loc[idn, ['Merk', 'Nama_Produk', 'Harga(IDR)', 'type',
    'Layar(inch)', 'merk_cpu', 'nama_cpu', 'RAM(GB)', 'ROM(GB)', 'Berat(KG)']]
    return final_recs

res=get_recs("MSI Stealth GS 15-A13VF-049ID")
res
```

	Merk	Nama_Produk	Harga(IDR)	type	Layar(inch)	merk_cpu	nama_cpu	RAM(GB)	ROM(GB)	Berat(KG)
14	MSI	MSI Bravo 15 C7VF-026ID	18999000	gaming	15.6	AMD	Ryzen 7 7735HS	16	512	1.96
19	Acer	Acer Gaming Nitro 5 AN515-58-710Q	18699000	gaming	15.6	intel	Core i7 12650H	16	512	2.40
11	Hp	HP Victus Gaming 15-FA0008TX	14999000	gaming	15.6	intel	Core i7 12700H	8	512	2.29

Gambar 9. Hasil dari sistem rekomendasi laptop

Gambar 9 merupakan coding untuk menampilkan hasil rekomendasi tiga laptop yang paling mirip atribut dan fitur spesifikasinya dengan index ke-15 yaitu laptop "MSI Stealth GS 15-A13VF-049ID" berdasarkan hasil *cosine similarity* yang telah dilakukan.

Pembahasan Hasil Cosine Similarity Sistem Rekomendasi

Pembahasan ini dilakukan dengan tujuan untuk menguji apakah hasil *cosine similarity* yang didapatkan oleh sistem rekomendasi sudah *valid* / sesuai dengan hitungan manual rumus *cosine similarity* atau belum. *Cosine similarity* adalah salah satu metode pengukuran kemiripan antar dua dokumen yang berbeda dengan menghitung kosinus sudut yang terbentuk oleh vektor yang merepresentasikan masing-masing dokumen (Fauzi et al., 2017). Pengujian dilakukan menggunakan data index ke-15 yaitu laptop "MSI Stealth GS 15-A13VF-049ID" untuk dibandingkan dengan tiga laptop paling mirip yaitu laptop index 14, 19 dan 11. Rumus *cosine similarity* menurut (Rifai & Anugrah, 2021) sebagai berikut :

$$\text{Sim}(i_1, i_2) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (1)$$

Keterangan :

$\cos(\theta)$ = tingkat kesamaan dokumen

A = vektor A, yang akan dibandingkan kemiripannya

B = vektor B, yang akan dibandingkan kemiripannya

$A \cdot B$ = hasil perkalian (dot product) antara vektor A dan vektor B

$\|A\|$ = panjang vector A

$\|B\|$ = panjang vector B

$\|A\| \|B\|$ = perkalian panjang (magnitude) dari dua vektor A dan B

Cosine similarity berjarak dari -1 ke 1. 1 berarti kedua sampel adalah yang paling mirip dan -1 berarti kedua sampel tersebut paling tidak mirip. Seperti yang kita tahu bahwa a dan b adalah vektor satuan, atau $\|a\| = \|b\|$, 1 berarti kedua sampel adalah identik dan -1 berarti kedua sampel berlawanan.

Data yang digunakan untuk perhitungan *cosine similarity* :

Tabel 6. Data yang digunakan untuk perhitungan *cosine similarity*

index	Layar (inch)	RAM (GB)	RO (GB)	Berat (KG)	Merk_Acer	Merk_Asu	Merk_HP	Merk_Lenovo	Merk_MSI	type_gaming	type_notebook	merk_cpu_AMD	merk_cpu_intel	cpu_model	cpu_generation
0	1	0	0	1.55	0	1	0	0	0	0	1	0	1	3	0
1	1	2	1	1.6	0	1	0	0	0	0	1	0	1	5	1
2	3	2	1	1.6	0	1	0	0	0	0	1	1	0	5	0
3	0	2	2	1.3	0	1	0	0	0	1	0	1	0	7	1
4	1	1	1	1.4	0	0	0	1	0	0	1	1	0	3	2
5	2	1	1	2.32	0	0	0	1	0	1	0	1	0	7	2
6	1	1	1	1.5	0	0	0	1	0	0	1	0	1	5	1
7	1	2	2	1.49	0	0	0	1	0	0	1	0	1	7	2
8	1	2	1	1.51	0	0	1	0	0	0	1	0	1	5	2
9	1	1	1	1.4	0	0	1	0	0	0	1	1	0	5	2
10	1	2	1	1.46	0	0	1	0	0	0	1	1	0	7	0
11	2	1	1	2.29	0	0	1	0	0	1	0	0	1	7	1
12	1	1	1	1.4	0	0	0	0	1	0	1	0	1	3	0
13	1	2	1	1.4	0	0	0	0	1	0	1	1	0	7	0
14	2	2	1	1.96	0	0	0	0	1	1	0	1	0	7	2
15	2	2	2	2.1	0	0	0	0	1	1	0	0	1	7	2
16	1	1	1	1.4	1	0	0	0	0	0	1	1	0	3	2
17	1	2	1	1.2	1	0	0	0	0	0	1	1	0	5	2
18	1	1	1	4.5	1	0	0	0	0	0	1	0	1	5	2
19	2	2	1	2.4	1	0	0	0	0	1	0	0	1	7	1

Tabel 6 menunjukkan data yang telah diolah menggunakan *one hot encoding* dan *ordinal encoding* dan siap untuk dihitung kemiripannya menggunakan algoritma *cosine similarity*. Index laptop yang digunakan untuk acuan adalah index ke-15 yaitu laptop "MSI Stealth GS 15-A13VF-049ID" dan akan dihitung *cosine similarity*nya dengan laptop index ke-14, 19, dan 11. Berikut merupakan perhitungannya.

Index 15 dan Index 14 :

$$[15] = (2, 2, 2, 2.10, 0, 0, 0, 0, 1, 1, 0, 0, 1, 7, 2) = A$$

$$[14] = (2, 2, 1, 1.96, 0, 0, 0, 0, 1, 1, 0, 1, 0, 7, 2) = B$$

$$A.B = 69.116$$

$$\|A\| = \sqrt{72.41}$$

$$= 8.50940656$$

$$\|B\| = \sqrt{68.8416}$$

$$= 8.29708325$$

$$\|A\|\|B\| = 8.50940656 \times 8.29708325$$

$$= 70.60325953$$

$$\frac{A.B}{\|A\|\|B\|} = \frac{69.116}{70.60325953}$$

$$= 0.9789349735$$

$$= 0.98$$

Index 15 dan Index 19 :

$$[15] = (2, 2, 2, 2.10, 0, 0, 0, 0, 1, 1, 0, 0, 1, 7, 2) = A$$

$$[19] = (2, 2, 1, 2.40, 1, 0, 0, 0, 0, 1, 0, 0, 1, 7, 1) = B$$

$$A.B = 68.04$$

$$\|A\| = \sqrt{72.41}$$

$$= 8.50940656$$

$$\|B\| = \sqrt{67.76}$$

$$= 8.231646251$$

$$\|A\|\|B\| = 8.50940656 \times 8.231646251$$

$$= 70.04642461$$

$$\frac{A.B}{\|A\|\|B\|} = \frac{68.04}{70.04642461}$$

$$= 0.9713557884$$

$$= 0.97$$

Index 15 dan Index 11 :

$$[15] = (2, 2, 2, 2.10, 0, 0, 0, 0, 1, 1, 0, 0, 1, 7, 2) = A$$

$$[11] = (2, 1, 1, 2.29, 0, 0, 1, 0, 0, 1, 0, 0, 1, 7, 1) = B$$

$$A.B = 65.809$$

$$\|A\| = \sqrt{72.41}$$

$$= 8.50940656$$

$$\|B\| = \sqrt{64.2441}$$

$$= 8.015241731$$

$$\|A\| \|B\| = 8.50940656 \times 8.015241731$$

$$= 68.20495057$$

$$\frac{A.B}{\|A\| \|B\|} = \frac{65.809}{68.20495057}$$

$$= 0.9648713$$

$$= 0.96$$

Setelah dilakukan perhitungan manual dengan rumus cosine similarity antara laptop index ke-15 yaitu laptop "MSI Stealth GS 15-A13VF-049ID" dengan laptop index ke-14 "MSI Bravo 15 C7VF-026ID", index ke-19 "Acer Gaming Nitro 5 710Q ", dan laptop index ke-11 "HP Victus Gaming 15-FA0008TX" telah terbukti valid dan sama dengan apa yang ada dalam sistem seperti yang ada pada gambar 8 sebelumnya.

KESIMPULAN DAN SARAN

Kesimpulan

Setelah dilakukan pembahasan dan pengujian pada sistem rekomendasi *content-based filtering* untuk laptop di Els Computer Shop Semarang dengan *cosine similarity* dalam penelitian ini, dapat disimpulkan sebagai berikut: 1) Sistem rekomendasi ini memberikan tiga saran atau rekomendasi laptop yang dicari berdasarkan perhitungan *cosine similarity* atribut spesifikasi laptop antara index laptop yang dicari kemiripannya dengan index laptop lainnya; 2) Dua *encoding data* digunakan yaitu *one hot encoding* dan *ordinal encoding* yang bertujuan untuk mengubah data kategorikal spesifikasi laptop dan memudahkan proses perhitungan *cosine similarity*; 3) Dengan adanya sistem rekomendasi ini akan memudahkan pelanggan Els Computer Shop Semarang untuk memilih laptop seperti dengan yang mereka butuhkan berdasarkan atribut spesifikasi laptop yang diinginkan.

Saran

Adapun saran yang diberikan dalam penelitian ini yang dapat diperhatikan oleh peneliti selanjutnya adalah sebagai berikut : 1) Data laptop yang ada pada dataset peneliti hanya sebanyak 20 laptop saja, diharapkan kedepannya agar data laptop terus ditambah dan diperbarui mengikuti data terkini yang ada pada *website* els computer shop; 2) Penelitian kedepannya dapat dikembangkan dalam bentuk aplikasi baik itu aplikasi berbasis *web* maupun *mobile*.

UCAPAN TERIMA KASIH

Penelitian dan penyusunan jurnal ini tidak mungkin terselesaikan tanpa dukungan, arahan, doa, dan motivasi yang diberikan oleh orang tua, dosen, sahabat, teman, dan individu lain yang memberikan bantuan selama periode penelitian. Penulis juga ingin menyampaikan rasa syukur atas berbagai nikmat yang Allah SWT anugerahkan, termasuk pengetahuan, pengalaman, kesehatan, dan rezeki materi.

REFERENSI

- R. S. Kanmani and B. Surendiran, "Context-Based Social Media Recommendation System," Recommender System with Machine Learning and Artificial Intelligence: Practical Tools and Applications in Medical, Agricultural and Other Industries, p. 237, 2020. <https://doi.org/10.1002/9781119711582.ch12>
- Sharma, S., Rana, V. & Malhotra, M. Automatic recommendation system based on hybrid filtering algorithm. *Educ Inf Technol* **27**, 1523–1538 (2022). <https://doi.org/10.1007/s10639-021-10643-8>

- S. Kausar et al., "Mining Smart Learning Analytics Data Using Ensemble Classifiers," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, no. 12, pp. 81- 102, 2020. <https://doi.org/10.3991/ijet.v15i12.13455>
- Nastiti, P. (2019). Penerapan Metode Content Based Filtering Dalam Implementasi Sistem Rekomendasi Tanaman Pangan. *Teknika*, 8(1), 1-10. <https://doi.org/10.34148/teknika.v8i1.139>
- PRAMESTI, Dewa Ayu Putri Diah; SANTIYASA, I Wayan. Penerapan Metode Content-Based Filtering dalam Sistem Rekomendasi Video Game. **Jurnal Nasional Teknologi Informasi dan Aplikasinya**, [S.l.], v. 1, n. 1, p. 229-234, nov. 2022. ISSN 2986-3929. Available at: <<https://ojs.unud.ac.id/index.php/jnatia/article/view/92555>>. Date accessed: 10 nov. 2023
- Muliawan, A., Badriyah, T., & Syarif, I. (2022). Membangun Sistem Rekomendasi Hotel dengan Content Based Filtering Menggunakan K-Nearest Neighbor dan Haversine Formula. *Technomedia Journal*, 7(2 October), 231–247. <https://doi.org/10.33050/tmj.v7i2.1893>
- POTDAR, K., S., T., & D., C., 2017. A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers. *International Journal of Computer Applications*, 175(4), 7–9
- Bajiyanta Roy. (2019, Juli 16). *All about Categorical Variable Encoding*. Diakses dari <https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02>
- Els. Pusat Belanja Komputer. Diakses pada Oktober 15, 2023, dari <https://els.id/>
- Boeing, G., & Waddell, P. (2017). New Insights into Rental Housing Markets across the United States: Web Scraping and Analyzing Craigslist Rental Listings. *Journal of Planning Education and Research*, 37(4), 457-476. <https://doi.org/10.1177/0739456X16664789>
- Pressman, S. R., and Maxim, R.B. 2015. *Software Engineering: A Practioner's Approach*. McGraw-Hill Education: New York.
- Google Colab FAQ. Google Colaboratory. Diakses pada November 10, 2023, dari <https://research.google.com/Colaboratory/intl/id/faq.html>
- Fauzi, M. A., Arifin, A. Z., & Yuniarti, A. (2017). Arabic book retrieval using class and book index based term weighting. *International Journal of Electrical and Computer Engineering*, 7(6), 3705-3710. <https://doi.org/10.11591/ijece.v7i6.pp3705-3711>
- Rifai, M. A., & Anugrah, I. G. (2021). Semantic Search for Scientific Articles by Language Using Cosine Similarity Algorithm and Weighted Tree Similarity. *Journal of Development Research*, 5(2), 106–114. <https://doi.org/10.28926/jdr.v5i2.150>
- Lei Mao. (2021, September 22). *Cosine Similarity VS Pearson Correlation Coefficient*. Diakses dari <https://leimao.github.io/blog/Cosine-Similarity-VS-Pearson-Correlation-Coefficient/>