JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 12 No. 3, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v12i3.5112

IMPLEMENTASI NODE.JS DAN PYTHON UNTUK SISTEM REKOMENDASI LAPTOP

Mochammad Fajrin[1], Munif[2], Danang Bagus Reknadi[3], Mustain[4]

[1,2,3,4]Teknik Informatika, Universitas Islam Lamongan, Lamongan, Jawa Timur, Indonesia 62218

Received: 15 Juli 2024
Accepted: 31 Juli 2024
Published: 7 Agustus 2024

Keywords:
Content Based Filtering; Cosine Similarity; Euclidean Distance; Laptop; Sistem Rekomendasi.

Corespondent Email:
mfajrin@unisla.ac.id,
munif@unisla.ac.id,
danz.0907@unisla.ac.id,
mustain@unisla.ac.id

Abstrak. Laptop adalah komputer pribadi yang kecil dan ringan, sehingga *portabel*. Dengan banyaknya *brand* dan spesifikasi yang ada saat ini, calon pembeli sering kebingungan memilih laptop yang sesuai dengan kebutuhan mereka. Kebanyakan orang lebih familiar dengan aplikasi yang digunakan sehari-hari. Penelitian ini bertujuan untuk mengatasi masalah tersebut dengan mengembangkan sistem rekomendasi laptop. Sistem ini akan membantu calon pembeli memilih laptop yang sesuai dengan kebutuhan spesifikasi minimum aplikasi yang akan digunakan. Metode yang digunakan adalah *Content Based Filtering*, yang memberikan rekomendasi berdasarkan profil preferensi pengguna dan deskripsi item. Algoritma yang diterapkan adalah *Euclidean Distance* dan *Cosine Similarity*. Hasil penelitian menunjukkan bahwa *Euclidean Distance* dapat merekomendasikan laptop berdasarkan spesifikasi yang paling mendekati kebutuhan aplikasi, sementara *Cosine Similarity* dapat memberikan rekomendasi laptop dengan spesifikasi serupa saat melihat detail dari hasil rekomendasi *Euclidean Distance*. Pengujian dengan 100 kali uji data menghasilkan rata-rata precision sebesar 98.4% pada *Cosine Similarity*.

Abstract. *A laptop is a personal computer that is small and lightweight, making it portable. With the plethora of brands and specifications available today, potential buyers often find it challenging to choose a laptop that meets their needs. Most people are more familiar with the applications they use daily. This research aims to address this issue by developing a laptop recommendation system. This system will assist potential buyers in selecting a laptop that meets the minimum system requirements of the applications they intend to use. The method used is Content Based Filtering, which provides recommendations based on user preference profiles and item descriptions. The algorithms applied are Euclidean Distance and Cosine Similarity. The results show that Euclidean Distance can recommend laptops based on the specifications closest to the application's needs, while Cosine Similarity can recommend laptops with similar specifications when viewing the details from the Euclidean Distance recommendation results. Testing with 100 trials resulted in an average precision of 98.4% for Cosine Similarity..*

1. PENDAHULUAN

Laptop adalah komputer pribadi yang berukuran relatif kecil dan ringan sehingga sifatnya ini membuatnya menjadi portabel. Laptop atau yang terkadang disebut juga dengan komputer notebook atau komputer pangku, untuk ukuran yang lebih kecil disebut sebagai netbook. Laptop berfungsi yang sama seperti dengan komputer desktop pada umumnya, yaitu sebagai komputer pribadi. Laptop juga memiliki komponen yang sama persis yang terdapat di dalamnya dengan komponen pada desktop, yang membedakannya

hanyalah pada ukurannya diperkecil, lebih ringan, lebih tidak panas, dan lebih hemat[1].

Berbeda dengan komputer desktop, laptop memiliki komponen pendukung yang didesain secara khusus untuk mengakomodasi sifatnya yang portabel. Komponen penyusun laptop memiliki sifat utama seperti ukuran yang kecil, hemat konsumsi energi, dan efisien yang sangat berbeda dengan komputer desktop. Hal tersebut biasanya membuat harga laptop menjadi lebih mahal, tergantung dari brand dan spesifikasi komponen penyusunnya. Meski demikian harga laptop juga semakin mendekati harga desktop seiring dengan semakin tingginya tingkat permintaan konsumen yang mempengaruhi hal tersebut. Saat ini laptop memiliki berbagai macam desain yang menarik dan kekinian, laptop menjadi lebih ringan dan tipis termasuk dari sisi teknologinya yang kian canggih[1].

Di Indonesia sendiri laptop adalah sebuah alat elektronik yang tidak bisa dipisahkan bagi masyarakat Indonesia. Masyarakat biasanya membeli laptop digunakan untuk kebutuhan kantor, sekolah, bermain games dan kegiatan pengembangan seperti desain grafis dan pengembang. Semakin berkembangnya zaman maka laptop saat ini memiliki berbagai macam brand dan spesifikasi yang dapat membuat Calon pembeli ketika memilih serta mencari sebuah laptop yang sesuai dengan kebutuhan menjadi lebih membingungkan serta menyusahkan. Calon pembeli yang awam dengan spesifikasi sebuah laptop biasanya akan kebingungan untuk menentukan laptop yang sesuai dengan kebutuhan, Akan tetapi kebanyakan masyarakat lebih mengenali aplikasi atau software yang biasanya digunakan untuk melakukan kegiatan untuk sekolah, perkantoran, bermain games dan kegiatan pengembangan sehingga aplikasi yang dibutuhkan tersebut bisa menjadi acuan sebuah rekomendasi laptop. Dengan adanya permasalahan tersebut maka Calon pembeli yang memerlukan laptop akan menjadi lebih terbantu ketika terdapat suatu sistem yang dapat merekomendasikan laptop dengan spesifikasi yang paling mendekati kebutuhan aplikasi yang digunakan. Masyarakat memerlukan rekomendasi laptop, untuk itu diperlukan suatu sistem rekomendasi[2] laptop untuk memberikan saran kepada masyarakat agar dapat membeli laptop yang memiliki spesifikasi yang sesuai dan dapat menunjang kebutuhan sistem minimum aplikasi yang akan digunakan.

Berdasarkan penelitian sebelumnya yang berjudul Sistem Rekomendasi Jurusan SMK Menggunakan Metode Content-Based Filtering Di Kabupaten Sleman menggunakan metode Content Based Filtering [3] dengan menggunakan teknik pembobotan dan perhitungan Euclidean Distance dapat memberikan rekomendasi kepada Pengguna sesuai dengan minat, bakat dan kemampuan Pengguna sendiri yang bisa menghasilkan daftar rekomendasi jurusan di sistem dan sistem rekomendasi pemilihan minat[4], bakat dan kemampuan dapat memberikan rekomendasi jurusan dan sekolah sesuai dengan minat, bakat dan kemampuan yang telah dimasukkan user sesuai dengan jarak euclidean terdekat.

Berdasarkan penelitian lainnya yang berjudul Sistem Rekomendasi Buku Menggunakan Metode Content Based Filtering membahas sistem rekomendasi buku dengan menggunakan teknik pembobotan TF-IDF dan algoritma Cosine Similarity [5] dapat memberikan rekomendasi buku kepada Pengguna sesuai dengan kemiripan data buku yang tersedia pada database. Sistem rekomendasi buku dengan menggunakan metode Content Based Filtering pada penelitian ini memiliki tingkat akurasi precision sebesar 85%. Sistem rekomendasi buku [6] tersebut dapat diimplementasikan pada website dengan menggunakan framework Flask[7] dan dapat berjalan dengan baik pada sebuah cloud platform Heroku.

Tujuan dari penelitian implementasi node.js[8] dan python[9] untuk sistem rekomendasi laptop adalah untuk membantu masyarakat atau Calon pembeli yang kurang mengerti dan paham mengenai spesifikasi sebuah laptop bisa terbantu dengan diberikannya rekomendasi laptop-laptop yang bisa menunjang kebutuhan sistem aplikasi yang akan digunakan oleh Calon pembeli dengan menggunakan *Euclidean Distance* sedangkan untuk memberikan rekomendasi laptop yang memiliki spesifikasi yang sama atau mirip ketika melihat detail laptop sebagai alternatif menggunakan *Cosine Similarity*.

Pembaharuan pada penelitian ini dari penelitian terdahulu adalah bahasa pemrograman yang digunakan adalah Node.js yang digunakan untuk operasi create, read,

update dan delete, sedangkan Python digunakan untuk membangun sistem rekomendasi laptop menggunakan *metode Content Based Filtering*, algoritma yang digunakan adalah *Euclidean Distance* dan *Cosine Similarity* yang masing-masing digunakan untuk dua aspek yang berbeda. Euclidean Distance digunakan untuk merekomendasikan laptop berdasarkan kebutuhan sistem minimum aplikasi pilihan Calon pembeli dengan mengukur jarak antara kebutuhan sistem minimum aplikasi dengan semua spesifikasi data laptop pada database, sedangkan *Cosine Similarity* digunakan untuk merekomendasikan laptop dengan spesifikasi laptop yang mirip ketika Calon pembeli melihat detail sebuah laptop sebagai pilihan alternatif dengan mengukur kemiripan spesifikasi laptop yang dilihat dengan semua spesifikasi data laptop pada database.

2. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini adalah *Content Based Filtering, Content Based Filtering* adalah sistem rekomendasi yang memberikan rekomendasi kepada Pengguna berdasarkan pada profil preferensi Pengguna dan deskripsi item. Umumnya dalam sistem rekomendasi, data dari variabel akan dianggap sebagai vektor. Ada banyak algoritma yang sering digunakan untuk mengukur kemiripan antara dua set data atau vektor di antaranya adalah *Adjusted Cosine, Cosine Similarity* atau *Pearson Coefficient* [5]. Algoritma yang digunakan untuk *Content Based Filtering* pada penelitian adalah *Euclidean Distance* dan *Cosine Similarity*

2.1. Euclidean Distance

Jarak *euclidean* atau *Euclidean Distance* adalah algoritma perhitungan jarak dari dua buah titik dalam ruang *euclidean* untuk mempelajari hubungan antara sudut dan jarak[10]. *Euclidean Distance* juga biasanya diterapkan untuk membantu proses klasifikasi pada data mining. *K-Nearest Neighbors* adalah salah satu contoh algoritma yang menggunakan perhitungan jarak menggunakan *Euclidean Distance*. Berikut adalah rumus dari *Euclidean Distance*.

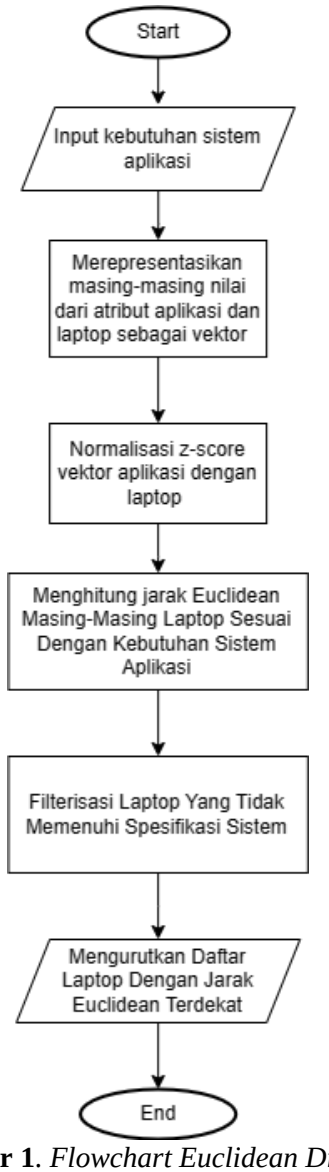
$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Keterangan:

d = Jarak antar dua titik

x,y = koordinat titik pertama dan koordinat titik kedua.

Flowchart pada Gambar 1. menjelaskan proses rekomendasi *menggunakan Euclidean Distance*.



Gambar 1. *Flowchart Euclidean Distance*

Input kebutuhan sistem aplikasi ini didapat oleh nilai tertinggi dari masing-masing variabel aplikasi yang dimasukkan oleh Calon pembeli. Variabel pada data laptop dan kebutuhan sistem aplikasi kemudian dijadikan vektor, agar tidak ada nilai variabel yang terlalu mendominasi karena angka yang terlalu tinggi maka vektor harus dinormalisasikan, normalisasi yang digunakan adalah normalisasi *Z-Score* agar data tidak ada nilai di salah satu nilai variabel yang terlalu mendominasi sebelum mencari

Euclidean Distance. Setelah ditemukan jarak *euclidean* antara kebutuhan sistem aplikasi dengan laptop, sistem akan mengurutkan laptop dengan jarak *Euclidean Distance* terdekat sebagai rekomendasi laptop.

2.2. Cosine Similarity

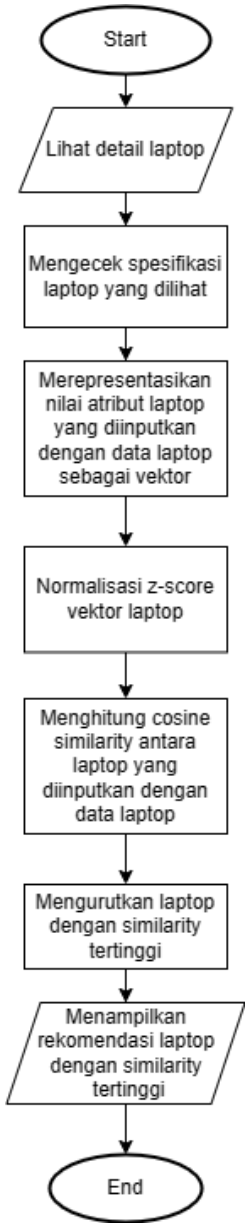
Cosine Similarity adalah algoritma yang digunakan untuk mengukur kemiripan antara dua dokumen atau teks. Pada *Cosine Similarity* menganggap dokumen atau teks sebagai vektor. Kemiripan antara vektor-vektor tersebut diukur berdasarkan sudut yang terbentuk dalam ruang[5].

Hasil dari *Cosine Similarity* akan selalu berada dalam rentang nilai negatif satu (-1) sampai satu (1). Nilai satu menunjukkan bahwa kedua vektor sejajar atau sama persis, nilai nol menunjukkan bahwa vektor-vektor tersebut tegak lurus, dan nilai negatif satu menunjukkan bahwa vektor-vektor tersebut sejajar dengan arah yang berlawanan atau tidak mirip sama sekali. Jika nilai *Cosine Similarity* semakin tinggi maka, semakin mirip vektor-vektor tersebut. Untuk menghitung *Cosine Similarity* dari dua buah vektor, rumus yang digunakan adalah sebagai berikut.

$$cosine\ similarity(A,B) = \frac{A \cdot B}{\|A\|\|B\|}$$

Keterangan:
(*A,B*) = dua vektor yang akan dibandingkan kesamaannya
A·B = hasil perkalian dot product antara vektor *A* dan *B*
‖*A*‖‖*B*‖ = panjang atau magnitude dari vektor *A* dan *B.*

Flowchart pada Gambar 2 menjelaskan tentang proses rekomendasi menggunakan *Cosine Similarity* ketika Calon pembeli melihat detail sebuah laptop, maka sistem akan memberikan rekomendasi laptop dengan spesifikasi yang mirip sebagai alternatif pilihan. Sama seperti *Euclidean Distance*, agar tidak ada nilai dari variabel yang terlalu mendominasi maka vektor laptop yang dilihat dengan laptop lain dilakukan normalisasi menggunakan *Z-Score*.



Gambar 2. *Flowchart Cosine Similarity*

3. HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk mengimplementasikan sistem rekomendasi laptop menggunakan metode *Content Based Filtering* dengan menggunakan algoritma *Euclidean Distance* dan *Cosine Similarity*. Pembuatan sistem ini menggunakan data melalui situs Laptop Media, Kaggle[11] dan diambil seacara manual dari official website berhasil memberikan referensi dan rekomendasi kepada Calon pembeli mengenai laptop yang spesifikasinya mendekati dengan kebutuhan minimum sistem aplikasi dan laptop yang memiliki kemiripan dengan laptop yang dilihat detailnya.

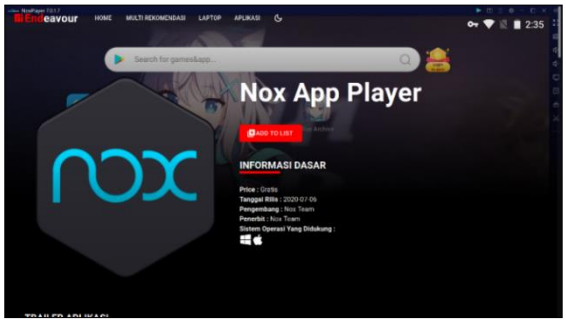
3.1 Hasil Implementasi GUI

Hasil dari Sistem rekomendasi laptop yang telah berhasil dibuat kemudian akan diimplementasikan dalam bentuk *Graphical User Interface* (GUI) pada sebuah website. Dengan GUI, Calon pembeli akan berinteraksi dengan mudah untuk menggunakan sistem rekomendasi yang dibangun pada penelitian ini sehingga Calon pembeli tidak harus menjalankan kode program secara manual namun dapat dijalankan menggunakan GUI. Berikut adalah pembahasan implementasi GUI pada sistem rekomendasi laptop.

3.1.1 Rekomendasi Dari Euclidean Distance

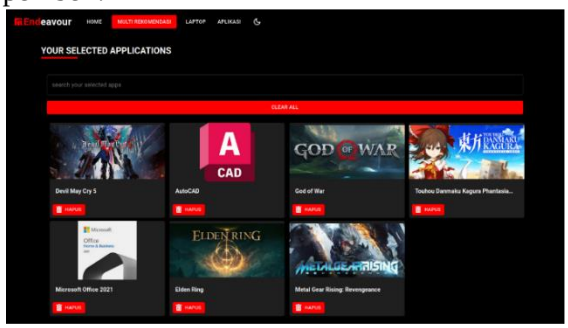
Hasil rekomendasi dari algoritma *Euclidean Distance* akan ditampilkan pada dua halaman yang berbeda, yang pertama hasil rekomendasi dapat dilihat ketika melihat detail aplikasi dan yang kedua pada halaman multi rekomendasi, hasil rekomendasi dari *Euclidean Distance* akan ditampilkan pada bagian bawah halaman.

Pada Gambar 3 merupakan halaman yang berisi informasi lebih lanjut mengenai aplikasi yang Calon pembeli akan gunakan pada laptop yang akan dibeli, Selain berisi informasi lebih lanjut tentang aplikasi, halaman ini juga akan merekomendasikan laptop yang mendekati kebutuhan sistem aplikasi yang sedang dilihat detailnya.



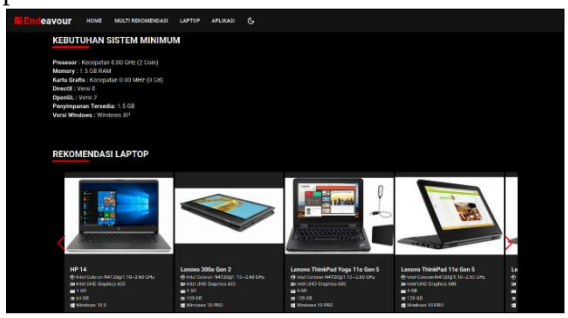
Gambar 3. Halaman Detail Aplikasi

Gambar 4 merupakan Halaman *multi* rekomendasi yang berfungsi untuk menyimpan banyak data aplikasi pilihan Calon pembeli yang digunakan sebagai parameter untuk merekomendasikan laptop kepada Calon pembeli. Sistem akan memberikan lima rekomendasi laptop dengan *Euclidean Distance* terdekat dan juga melakukan proses *filter* jika spesifikasi laptop terdekat ternyata tidak memenuhi kebutuhan sistem aplikasi Calon pembeli.



Gambar 4. Halaman Multi Rekomendasi

Pada bagian bawah halaman detail aplikasi dan multi rekomendasi terdapat lima rekomendasi laptop yang memenuhi kebutuhan minimum sistem aplikasi yang dapat dilihat pada Gambar 5.



Gambar 5. Hasil Rekomendasi Euclidean Distance

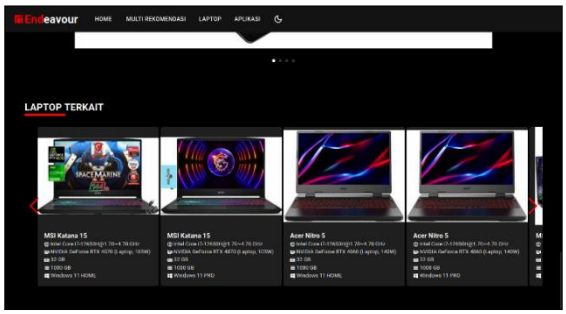
Dengan adanya rekomendasi ini mempermudah Calon pembeli untuk mengetahui laptop apa yang dapat digunakan untuk menjalankan aplikasi yang sedang dilihat. Rekomendasi ini secara otomatis dilakukan oleh Next.js[12]. Ketika Calon pembeli melihat detail aplikasi, Next.js akan mengirimkan id aplikasi yang sedang dilihat ke API Flask untuk mendapatkan lima rekomendasi laptop dengan spesifikasi yang mendekati kebutuhan sistem aplikasi yang dilihat detailnya.

3.1.2 Rekomendasi Dari Cosine Similarity

Hasil rekomendasi dari algoritma *Cosine Similarity* akan ditampilkan pada Gambar 6, Calon pembeli akan mendapatkan informasi spesifikasi detail dari laptop dan juga rekomendasi lima laptop terkait. Laptop terkait memiliki kemiripan dengan laptop yang sedang dilihat detailnya berdasarkan tingkat *Cosine Similarity* tertinggi.

Gambar 6. Detail Laptop Bagian Spesifikasi

Pada Gambar 7 menunjukkan bagian bawah detail laptop, akan ada bagian yang menampilkan lima laptop yang memiliki kesamaan spesifikasi dengan laptop yang sedang dilihat detailnya.



Gambar 7. Hasil Rekomendasi *Cosine Similarity*

Rekomendasi ini berguna untuk mempermudah Calon pembeli apabila kurang suka dengan hasil rekomendasi dari multi rekomendasi atau ingin mencari alternatif laptop dengan *brand* lain, tapi memiliki spesifikasi yang mirip dengan laptop yang dilihat detailnya.

3.2 Pembahasan

Untuk lebih memahami konsep dan prinsip dari metode yang digunakan, maka penting sekali untuk melakukan perhitungan metode secara manual. Berikut ini adalah berbagai tahapan untuk perhitungan manual metode untuk sistem rekomendasi laptop.

3.2.1. Persiapan Data

Berikut adalah pembahasan tentang persiapan data untuk perhitungan metode secara manual.

a) Data Aplikasi Kebutuhan Calon Pembeli

Data aplikasi pilihan yang Calon pembeli akan digunakan pada laptop yang akan dibeli ditunjukkan pada Tabel 1.

Tabel 1. Data Aplikasi Pilihan

Nama	A1	A2	A3	A4	A5	A6	A7
Min Cpu Speed	3.3	2.7	3.2	1.6	2.5	3.3	2.85
Min Cpu Cores	4	4	4	2	1	5	6
Min Gpu Boost Clock	1178	950	1032	0	0	1524.5	0
Min VRAM	4	0	2	0	1	5.12	0.768
Min DirectX	11	11	11	10	11	12.05	10.55
Min OpenGL	4.6	4.6	4.6	0	0	4.5	3.65
Min RAM	8	4	8	2	8	12	2
Min Storage	70	10	35	4	10	60	25
Min Windows Build Number	10240	10240	10240	10240	10240	10240	2600

b) Data Laptop

Data tabel yang memuat data laptop di bawah sudah melalui tahap *preprocessing* ditunjukkan pada Tabel 2. *Euclidean Distance* memerlukan data yang berupa numerik, karena itu yang diambil adalah spesifikasi dari CPU dan GPU yang berupa angka, untuk kapasitas SSD dan HDD laptop akan digabung menjadi *storage*.

Tabel 2. Data Laptop

Id	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
CPU Max Speed	5.00	4.60	4.60	4.60	4.70	3.40	3.40	3.50	2.80	2.60
CPU Cores	6	8	8	8	10	2	2	2	2	4
GPU Max Speed	1740	1620	1620	1620	2175	1100	1100	1200	700	700
VRAM	8	8	8	8	8	0	0	0	0	0
DirectX	12.1	12.2	12.2	12.2	12.2	12.1	12.1	12.1	12.1	12.1
OpenGL	4.6	4.6	4.6	4.6	4.6	4.5	4.5	4.6	4.4	4.4
RAM	32	16	16	32	16	4	4	8	8	4
Storage	1000	1000	2000	1000	1000	1288	1288	512	1288	64
Windows Build Number	10240	10240	10240	10240	10240	10240	10240	10240	2200	10240

3.2.2. Mencari Kebutuhan Sistem Tertinggi

Untuk mencari kebutuhan sistem tertinggi maka hanya perlu mencari nilai tertinggi dari

masing-masing variabel, sedangkan pada bagian *min storage* akan dilakukan penjumlahan dari masing-masing aplikasi agar penyimpanan laptop cukup untuk memasang aplikasi pilihan Calon pembeli. Seperti pada Tabel 3, aplikasi dan kebutuhan sistem tertinggi akan diberikan id X agar tidak terlalu panjang. Sehingga ditemukan kebutuhan tertinggi adalah sebagai berikut.

Tabel 3. Kebutuhan Sistem Aplikasi Tertinggi

Variabel	Kebutuhan Sistem Aplikasi
Id	***X***
Min CPU Speed	***3.3***
Min CPU Cores	***4***
Min GPU Boost Clock	***1178***
Min VRAM	***4***
Min DirectX	***11***
Min OpenGL	***4.6***
Min RAM	***8***
Min Storage	***70***
Min Windows Build Number	***10240***

3.2.3. Menggabungkan Data Laptop Dengan Kebutuhan Sistem

Ssss

Untuk mencari laptop terdekat sesuai kebutuhan sistem aplikasi, maka perlu melakukan penggabungan data seperti pada Tabel 4 agar bisa mencari *Euclidean Distance*.

Tabel 4. Data Gabungan

Id	X	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
Cpu *Max Speed*	3.3	5.0	4.6	4.6	4.6	4.7	3.4	3.4	3.5	2.8	2.6
Cpu *Cores*	6	6	8	8	8	10	2	2	2	2	4
Gpu *Max Speed*	1524.5	1744	1622	1622	1622	2175	1100	1100	1200	700	700
VRAM	5.12	8	8	8	8	8	0	0	0	0	0
DirectX	12.05	12.1	12.2	12.2	12.2	12.2	12.1	12.1	12.1	12.1	12.1
OpenGL	4.6	4.6	4.6	4.6	4.6	4.6	4.5	4.5	4.6	4.4	4.4
RAM	12	32	16	16	32	16	4	4	8	8	4
Storage	214	100000000	100000000	200000000	100000000	100000000	128	128	512	128	64
Windows Build Number	10240	1000240	1000240	1000240	1000240	1000240	1000240	1000240	1000240	2000400	1000240

3.2.4. Menghitung *Euclidean Distance*

Agar tidak ada nilai variabel atau atribut yang terlalu mendominasi maka data perlu dinormalisasikan menggunakan Z-Score.

$$z = \frac{x - \mu}{\sigma}$$

Keterangan:
Z = *Z-score* (nilai standar atau nilai baku)
x = Nilai yang diamati (nilai mentah)
μ = Rata-rata
σ = Standar deviasi

Untuk mendapatkan nilai standar deviasi menggunakan rumus berikut:

$$S = \sqrt{\frac{\Sigma_i^N = 1(x_i - \bar{x})^2}{n-1}}$$

Keterangan:
S = Standar deviasi
x_i = nilai x ke-i
$\bar{x}$ = nilai rata-rata data
n = jumlah data

Mencari standar deviasi variabel Minimum CPU speed adalah sebagai berikut:

$\bar{x}$ = 42.50 / 11 = 3.86

$$S = \sqrt{\frac{\Sigma_i^N = 1(x_i - \bar{x})^2}{n-1}}$$

$$S = \sqrt{\frac{\begin{matrix}(3{,}3 - 3.86)^2 + (5 - 3.86)^2 + (4.6 - 3.86)^2 + \\ (4.6 - 3.86)^2 + (4.6 - 3.86)^2 + (4.7 - 3.86)^2 + \\ (3.4 - 3.86)^2 + (3.4 - 3.86)^2 + (3.5 - 3.86)^2 \\ + (2.8 - 3.86)^2 + (2.6 - 3.86)^2\end{matrix}}{11}}$$

$S = \sqrt{0.66}$ = 0.81

Langkah selanjutnya adalah mencari z-score. Menggunakan rumus berikut:

$$z = \frac{x - \mu}{\sigma} = \frac{3.3 - 3.86}{0.81} = 0.70$$

Hasil normalisasi di atas adalah nilai variabel CPU *max speed* pada baris pertama, untuk seluruh nilai dan baris variabel ditunjukkan pada Tabel 5.

Tabel 5. Hasil Normalisasi *Z-Score*

Id	X	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
CPU Max Speed	-070	140	091	091	091	103	-057	-057	-045	131	-156
CPU Cores	025	025	095	095	095	165	-114	-114	-114	-114	-044
GPU Max Speed	035	085	057	057	057	185	-063	-063	-040	-155	-155
VRAM	027	102	102	102	102	102	-107	-107	-107	-107	-107
DirectX	-153	-060	128	128	128	128	-060	-060	-060	-060	-060
OpenGL	070	070	070	070	070	070	-058	-058	070	-186	-186
RAM	-019	187	022	022	187	022	-101	-101	-060	-060	-101
Storage	-076	060	060	233	060	060	-091	-091	-024	-091	-102
Windows Build Number	-032	-032	-032	-032	-032	-032	-032	-032	-032	316	-032

Setelah data sudah dinormalisasikan, Langkah terakhir adalah menghitung *Euclidean Distance* kebutuhan sistem dengan spesifikasi semua laptop, untuk mendapatkan nilai *Euclidean Distance* adalah dengan perhitungan di bawah:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$d = \sqrt{\begin{matrix}(-0.7 - 1.4)^2 + (0.25 - 0.25)^2 + (0.35 - 0.85)^2 \\ +(0.27 - 1.02)^2 + (-1.53 - (-0.6))^2 + (0.7 - 0.7)^2 \\ +(-0.19 - 1.87)^2 + (-0.76 - 0.6))^2 + (-0.32 - (-1.32))^2\end{matrix}}$$

$$d = \sqrt{12.18} = 3.49$$

Hasil perhitungan di atas adalah jarak *euclidean* dari kebutuhan sistem dengan laptop L1, Tabel 6 menunjukkan hasil dari keseluruhan jarak *euclidean* dari kebutuhan sistem dengan semua laptop.

Tabel 6. Hasil *Euclidean Distance*

Id	*Euclidean Distance*
X	0
L1	3.49
L2	3.69
L3	4.61
L4	4.20
L5	4.20
L6	2.82
L7	2.82
L8	2.38
L9	5.24
L10	3.85

Setelah jarak *euclidean* ditemukan, langkah selanjutnya adalah mengurutkan data berdasarkan jarak *euclidean* terdekat ke yang terjauh. Setelah diurutkan maka laptop yang tidak memenuhi kebutuhan sistem aplikasi harus disaring atau dihilangkan, contoh jika aplikasi membutuhkan empat core CPU, maka laptop dengan core CPU akan disaring. Khusus pada bagian VRAM [13] yang tidak memiliki *memory* bawaan, jika sisa RAM masih tersedia untuk alokasi VRAM maka laptop tidak akan disaring.

Menurut Acer Indonesia, Semakin banyak RAM maka ukuran VRAM juga akan bertambah karena memori GPU akan mengambil (*shared*) dari memori RAM. Pengaturan *shared memory* ini akan otomatis dilakukan oleh sistem. Oleh karena itu, apabila memori RAM di *upgrade* menjadi lebih besar, maka semakin besar juga *shared memory* untuk VGA. Jadi jika RAM laptop masih tersisa maka laptop tersebut tidak disaring. Apabila ada sebuah nilai dari variabel yang tidak memenuhi kebutuhan sistem, maka akan di berikan warna merah yang menjadi alasan laptop tersebut harus disaring karena tidak memenuhi kebutuhan sistem aplikasi yang akan digunakan oleh Calon pembeli. Seperti pada Tabel 7 yang menunjukkan proses *filter* data yang tidak relevan.

Tabel 7. Filter Data Tidak Relevan

Id	X	L8	L6	L7	L1	L2	L10	L5	L4	L3	L9
Cpu Max Speed	33	35	34	34	5	46	26	47	46	46	28
Cpu Cores	6	2	2	2	6	8	4	10	8	8	2
Gpu Max Speed	15245	1200	1100	1100	1740	1620	700	2175	1620	1620	700
VRAM	512	0	0	0	8	8	0	8	8	8	0
DirectX	1205	121	121	121	121	122	121	122	122	122	121
OpenGL	46	46	45	45	46	46	44	46	46	46	44
RAM	12	8	4	4	32	16	4	16	32	16	8
Storage	214	512	128	128	1000	1000	64	1000	1000	2000	128
Windows Build Number	10240	10240	10240	10240	10240	10240	10240	10240	10240	10240	22000
VRAM dari RAM		512	512	512	0	0	512	0	0	0	512
SisaRAM		-912	-13.12	-13.12	20	4	-13.12	4	20	4	-912

Laptop yang memenuhi kebutuhan sistem aplikasi adalah laptop yang akan

direkomendasikan kepada Calon pembeli, laptop yang direkomendasikan adalah lima laptop dengan jarak *euclidean* terdekat dengan spesifikasi yang memenuhi kebutuhan sistem minimum aplikasi yang dipilih oleh Calon pembeli, Tabel 8 adalah hasil rekomendasi dari *Euclidean Distance* dan proses *filter* data yang tidak relevan.

Tabel 8. Hasil Rekomendasi *Euclidean Distance*

Id	L1	L2	L5	L4	L3
Cpu Speed	5	4.6	4.7	4.6	4.6
Cpu Cores	6	8	10	8	8
Gpu Max Speed	1740	1620	2175	1620	1620
VRAM	8	8	8	8	8
DirectX	12.1	12.2	12.2	12.2	12.2
OpenGL	4.6	4.6	4.6	4.6	4.6
RAM	32	16	16	32	16
Storage	1000	1000	1000	1000	2000
Build Number	10240	10240	10240	10240	10240

3.2.5. Menghitung *Cosine Similarity*

Untuk merekomendasikan laptop berdasarkan kemiripan dari spesifikasi ketika Calon pembeli melihat detail laptop, maka algoritma yang digunakan untuk mencari laptop dengan spesifikasi mirip adalah algoritma *Cosine Similarity*. Tabel 9 adalah tabel yang berisi data laptop yang akan digunakan untuk melakukan perhitungan manual algoritma *Cosine Similarity*. Pada bagian ini laptop yang akan dicari kemiripan datanya adalah laptop dengan id L1.

Tabel 9. Data Laptop

Id	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
CPU *Max Speed*	4.60	5.00	4.60	4.60	4.70	3.40	3.40	3.50	2.80	4.70
CPU *Cores*	8	6	8	8	10	2	2	2	2	14
GPU *Max Speed*	1620	1740	1620	1620	2175	1100	1100	1200	700	1485
VRAM	8	8	8	8	8	0	0	0	0	8
DirectX	12.2	12.1	12.2	12.2	12.2	12.1	12.1	12.1	12.1	12.2
OpenGL	4.6	4.6	4.6	4.6	4.6	4.5	4.5	4.6	4.4	4.6
RAM	16	32	16	32	16	4	4	8	8	16
Storage	1000	1000	2000	1000	1000	128	128	512	128	1000
Windows Build Number	10240	10240	10240	10240	10240	10240	10240	10240	22000	10240

Sama seperti *Euclidean Distance* agar tidak ada data yang terlalu mendominasi maka langkah selanjutnya yang perlu dilakukan adalah melakukan normalisasikan kembali semua nilai variabel menggunakan *z-score*, rumus yang digunakan sama seperti saat menghitung *Euclidean Distance*. Tabel 10 merupakan hasil dari normalisasi nilai setiap variabel menggunakan *Z-Score*.

Tabel 10. Hasil Normalisasi *Z-Score*

Id	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
CPU *Max Speed*	0.65	1.19	0.65	0.65	0.78	-1.00	-1.00	-0.87	-1.83	0.78
CPU *Cores*	0.46	-0.05	0.46	0.46	0.96	-1.06	-1.06	-1.06	-1.06	1.98
GPU *Max Speed*	0.47	0.77	0.47	0.47	1.87	-0.85	-0.85	-0.60	-1.86	0.12
VRAM	0.82	0.82	0.82	0.82	0.82	-1.22	-1.22	-1.22	-1.22	0.82
DirectX	1.00	-1.00	1.00	1.00	1.00	-1.00	-1.00	-1.00	-1.00	1.00
OpenGL	0.60	0.60	0.60	0.60	0.60	-0.90	-0.90	0.60	-2.41	0.60
RAM	0.08	1.75	0.08	1.75	0.08	-1.17	-1.17	-0.75	-0.75	0.08
Storage	0.38	0.38	2.18	0.38	0.38	-1.19	-1.19	-0.50	-1.19	0.38
Windows Build Number	-0.33	-0.33	-0.33	-0.33	-0.33	-0.33	-0.33	-0.33	3.00	-0.33

Setelah dinormalisasi, langkah terakhir adalah mencari *Cosine Similarity* antara laptop L1 dengan laptop lainnya. Di bawah adalah contoh langkah-langkah untuk mencari nilai *Cosine Similarity* dari laptop *L1* dan laptop *L2*.

a. Pertama, menjadikan nilai tiap variabel menjadi vektor:
   L1 = [0.65, 0.46, 0.47, 0.82, 1.00, 0.60, 0.08, 0.38, -0.33]
   L2 = *[1.19, -0.05, 0.77, 0.82, -1.00, 0.60, 1.75, 0.38, -0.33]*
b. Kedua, Menghitung dot product dari *L1* dan *L2*:
   $L1 \cdot L2$ = *(0.65 x 1.19) + (0.46 x (-0.05)) + (0.47 x 0.77) + (0.82 x 0.82) + (1.00 x 1.00)+ (0.60 x 0.60) + (0.08 x 1.75) + (0.38 x 0.38) + ((-0.33) x (-0.33))*
   $L1 \cdot L2$ = *1.54*
c. Ketiga, Menghitung panjang (*magnitude*) dari masing-masing vektor:

$$||L1|| = \sqrt{0.65^2 + 0.46^2 + 0.47^2 + 0.82^2 + 1.00^2 + 0.60^2 + 0.08^2 + 0.38^2 + (-0.33)^2} = \sqrt{3.13}$$

$$||L2|| = \sqrt{1.19^2 + (-0.05)^2 + 0.77^2 + 0.82^2 + (-1.00)^2 + 0.60^2 + 1.75^2 + 0.38^2 + (-0.33)^2} = \sqrt{7.37}$$

d. Terakhir menghitung *Cosine Similarity* dengan membagi *dot product* dengan perkalian panjang kedua vektor:

$$cosine\ similarity\ (L11, L1\ ) = \frac{L11 \cdot L1}{\|L11\|\|L1\|} = \frac{1.54}{\sqrt{3.13}\ \sqrt{7.37}}$$
$$= \frac{1.54}{\sqrt{23.09}} = 0.32$$

Hasil *Cosine Similarity* dari laptop *L11* dengan *L*1 adalah sebesar 0.32, artinya tingkat kemiripan laptop L1 dengan laptop *L2* adalah sebesar 32%. untuk menghitung *Cosine Similarity* untuk laptop lain juga menggunakan rumus yang sama seperti di atas. Pada Tabel 11 menunjukkan tingkat kemiripan spesifikasi yang sama dengan laptop L1.

Tabel 11. Hasil *Cosine Similarity*

Id	*Cosine Similarity*
L1	1.00
L2	0.32
L3	0.77
L4	0.74
L5	0.86
L6	-0.85
L7	-0.85
L8	-0.71
L9	-0.81
L10	0.81

laptop *L1* mendapatkan nilai *Cosine Similarity* sebesar 1.00 atau 100% karena memang data Laptop 1 dengan Laptop 1 sangat mirip karena memang dua laptop yang sama. Apabila nilai *Cosine Similarity* di bawah dari nol, maka laptop tersebut memiliki spesifikasi yang tidak sama sekali mirip atau tidak relevan, Laptop dengan nilai *Cosine Similarity* di bawah angka nol atau angka negatif akan, laptop tersebut akan disaring oleh sistem dan tidak ditampilkan ke aplikasi karena tidak relevan dengan data laptop L1.

Dari tabel hasil perhitungan *Cosine Similarity* di atas, laptop yang memiliki kemiripan tertinggi dengan laptop L1 adalah laptop L5, laptop L5 memiliki nilai *Cosine Similarity* sebesar 0.86 yang artinya, laptop L5 memiliki tingkat kemiripan sebesar 86% dengan laptop L1. Setelah penyaringan laptop dengan *Cosine Similarity* lebih kecil sama dengan nol, maka sistem akan menampilkan lima laptop dengan *Cosine Similarity* tertinggiyang ditunjukkan pada Tabel 12, dimana didalamnya merupakan laptop rekomendasi dengan tingkat kemiripan tertinggi dengan laptop L1.

Tabel 12. Hasil Rekomendasi *Cosine Similarity*

Id	L5	L10	L3	L4	L2
CPU *Max Speed*	4.7	4.7	4.6	4.6	5
CPU *Cores*	10	14	8	8	6
GPU *Max Speed*	2175	1485	1620	1620	1740
VRAM	8	8	8	8	8
DirectX	12.2	12.2	12.2	12.2	12.1
OpenGL	4.6	4.6	4.6	4.6	4.6
RAM	16	16	16	32	32
Storage	1000	1000	2000	1000	1000
Build Number	10240	10240	10240	10240	10240

4. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan dalam membuat sistem rekomendasi laptop berdasarkan kebutuhan sistem minimum aplikasi dan kemiripan spesifikasi laptop, maka dapat diambil kesimpulan bahwa

a. Sistem rekomendasi laptop berdasarkan kebutuhan sistem minimum aplikasi dan kemiripan spesifikasi laptop berhasil dibangun dan dapat berjalan dengan baik pada *website* dengan menggunakan Next.js sebagai sisi *frontend* dan Express.js[14] dengan Flask untuk sisi *backend*. Flask berhasil mengimplementasikan algoritma *Euclidean Distance* dan *Cosine Similarity* dengan baik sedangkan Express.js berhasil melakukan operasi *create, read, update* dan *delete* pada data laptop dan aplikasi.
b. Sistem rekomendasi laptop dengan menggunakan algoritma *Euclidean Distance* memiliki kinerja yang cukup baik dan dapat memberikan rekomendasi laptop pada Calon pembeli berdasarkan kebutuhan sistem minimum aplikasi yang dipilih. Laptop yang ditampilkan adalah laptop yang spesifikasinya dekat dengan kebutuhan sistem aplikasi. *Euclidean Distance* memiliki kekurangan pada laptop yang ditampilkan kadang tidak memenuhi kebutuhan sistem minimum [15] aplikasi pilihan, namun hal ini dapat diatasi dengan membuat *filter* pada laptop yang tidak memenuhi kebutuhan sistem aplikasi

pilihan sehingga laptop terdekat yang memenuhi kebutuhan sistem dapat direkomendasikan pada Calon pembeli.

c. Sistem rekomendasi laptop dengan menggunakan algoritma *Cosine Similarity* memiliki kinerja yang baik dan dapat memberikan rekomendasi laptop pada Calon pembeli sesuai dengan kemiripan spesifikasi laptop saat melihat detail laptop. Sistem rekomendasi laptop dengan menggunakan *Cosine Similarity* pada penelitian ini memiliki rata-rata *precision* sebesar 98.4%.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada rekan-rekan yang telah memberikan dukungan yang berharga sehingga penelitian ini dapat dilaksanakan dengan baik. Penulis juga berterima kasih kepada bapak Munif selaku dosen pengampu mata kuliah Analisa Perancangan Sistem dan seluruh teman-temanyang membantu penelitian ini.

DAFTAR PUSTAKA

[1] Primandha, "Cara Melihat Spesifikasi Laptop dengan Cepat dan Mudah," Best Seller Gramedia. Accessed: Jul. 10, 2024. [Online]. Available: https://www.gramedia.com/best-seller/cara-melihat-spesifikasi-laptop-dengan-cepat-dan-mudah/

[2] H. Februariyanti, A. D. Laksono, J. S. Wibowo, and M. S. Utomo, "Implementasi Metode Collaborative Filtering Untuk Sistem Rekomendasi Penjualan Pada Toko Mebel," *J. Khatulistiwa Inform.*, vol. 9, no. 1, Art. no. 1, Jun. 2021, doi: 10.31294/jki.v9i1.9859.

[3] E. T. B. Ginting and I. Pratama, "Filtering Di Kabupaten Sleman," vol. 3, no. 2, 2023.

[4] M. Fajriansyah, P. P. Adikara, and A. W. Widodo, "Sistem Rekomendasi Film Menggunakan Content Based Filtering," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 5, no. 6, Art. no. 6, May 2021.

[5] M. R. A. Zayyad, "Sistem Rekomendasi Buku Menggunakan Metode Content Based Filtering," Jul. 2021, Accessed: Jul. 10, 2024. [Online]. Available: https://dspace.uii.ac.id/handle/123456789/35942

[6] R. Ardiansyah, M. A. Bianto, and B. D. Saputra, "Sistem Rekomendasi Buku Perpustakaan Sekolah menggunakan Metode Content-Based Filtering," *J. CoSciTech Comput. Sci. Inf. Technol.*, vol. 4, no. 2, Art. no. 2, Oct. 2023, doi: 10.37859/coscitech.v4i2.5131.

[7] R. Maulid, "Mengenal Flask, Library Machine Learning Python Idaman Devel..." Accessed: Jul. 10, 2024. [Online]. Available: https://dqlab.id/mengenal-flask-library-machine-learning-python-idaman-developer

[8] D. Indonesia, "Tutorial Node.js: Apa itu Node.js dan dasar-dasar Node.js," Dicoding Blog. Accessed: Jul. 10, 2024. [Online]. Available: https://www.dicoding.com/blog/apa-itu-node-js/

[9] D. Intern, "Python: Pengertian, Contoh Penggunaan, dan Manfaat Mempelajarinya," Dicoding Blog. Accessed: Jul. 10, 2024. [Online]. Available: https://www.dicoding.com/blog/python-pengertian-contoh-penggunaan-dan-manfaat-mempelajarinya/

[10] "Fauzan: Mengukur Jarak Euclidean: Teori dan Implementasi... - Google Scholar." Accessed: Jul. 10, 2024. [Online]. Available: https://scholar.google.com/scholar?cluster=26651723940605586 29&hl=id&as_sdt=2005&sciodt=0,5

[11] N. Rahmalia, "Kaggle, Tempat Belajar dan Berkompetisi di Bidang Data Science," glints. Accessed: Jul. 10, 2024. [Online]. Available: https://glints.com/id/lowongan/kaggle-adalah/

[12] A. Muhammad, "Next.js: Pengertian, Cara Kerja, dan Tutorial Project Next.js," Niagahoster Blog. Accessed: Jul. 10, 2024. [Online]. Available: https://www.niagahoster.co.id/blog/next-js-adalah/

[13] Acer, "Ketahui Apa Itu VRAM, Cara Cek dan Menambah, hingga Fungsinya di Laptop." Accessed: Jul. 10, 2024. [Online]. Available: https://www.acerid.com/berita-bisnis/apa-itu-vram-cara-cek-dan-fungsinya-pada-laptop

[14] A. Muhammad, "Mengenal Express.js: Pengertian, Cara Kerja, Keunggulan, Tutorial," Niagahoster Blog. Accessed: Jul. 10, 2024. [Online]. Available: https://www.niagahoster.co.id/blog/express-js-adalah/

[15] H. Aliya, "4 Perbedaan Black Box dan White Box Testing - Glints Blog." Accessed: Jul. 10, 2024. [Online]. Available: https://glints.com/id/lowongan/black-box-dan-white-box-testing/