BAB 2
LANDASAN TEORI

Bab ini menguraikan landasan teori yang digunakan sebagai dasar dalam pengembangan dan pengujian sistem rekomendasi pemilihan laptop. Pembahasan mencakup konsep sistem rekomendasi, metode *Content-Based Filtering*, metadata, pemrosesan teks, pembobotan TF-IDF, normalisasi vektor, perhitungan *Cosine Similarity*, pemeringkatan rekomendasi *Top-k*, evaluasi relevansi rekomendasi, serta evaluasi kebergunaan. Bab ini juga membahas perbandingan *System Usability Scale* (SUS) dan *End-User Computing Satisfaction* (EUCS), serta menyajikan pemetaan penelitian terdahulu dan posisi penelitian secara ringkas untuk menunjukkan kesenjangan penelitian.

2.1 Landasan Teori

Bagian ini menjelaskan teori-teori utama yang mendukung penelitian. Fokus utama landasan teori diarahkan pada model rekomendasi berbasis konten, representasi teks menggunakan TF-IDF, pengukuran kemiripan menggunakan *Cosine Similarity*, serta evaluasi relevansi dan kebergunaan. Penerapan metode dijelaskan pada Bab III, sedangkan implementasi dan hasil penelitian disajikan pada Bab IV.

2.1.1 Sistem Rekomendasi

Sistem rekomendasi merupakan sistem penyaringan informasi (*information filtering system*) yang bertujuan membantu pengguna menemukan item yang sesuai dengan kebutuhan, preferensi, atau konteks tertentu dari sekumpulan alternatif yang tersedia [4]. Sistem rekomendasi banyak digunakan untuk mengatasi permasalahan *information overload*, yaitu kondisi ketika pengguna mengalami kesulitan dalam mengambil keputusan karena banyaknya pilihan dan informasi yang harus dipertimbangkan [1].

Dalam konteks pemilihan laptop, sistem rekomendasi berfungsi sebagai alat bantu pengambilan keputusan yang menyaring dan mengurutkan produk berdasarkan tingkat kesesuaian antara kebutuhan pengguna dan karakteristik laptop yang tersedia. Sistem ini membantu pengguna, khususnya pengguna non-teknis, dalam memahami alternatif produk tanpa harus melakukan analisis spesifikasi

secara mendalam. Istilah pengguna non-teknis dalam penelitian ini merujuk pada pengguna yang tidak memiliki pemahaman mendalam terhadap spesifikasi laptop, meskipun masih dapat memiliki pengetahuan dasar mengenai penggunaan laptop.

Secara umum, pendekatan sistem rekomendasi dapat diklasifikasikan ke dalam beberapa kategori, yaitu *Collaborative Filtering*, *Content-Based Filtering*, *Knowledge-Based Recommendation*, dan *Hybrid Recommendation* [4]. Masing-masing pendekatan memiliki karakteristik, kebutuhan data, serta mekanisme rekomendasi yang berbeda.

Penelitian ini menggunakan pendekatan *Content-Based Filtering* karena proses rekomendasi berfokus pada kesesuaian antara deskripsi kebutuhan pengguna dan representasi metadata laptop yang dibentuk dari atribut spesifikasi, kata kunci penggunaan, dan informasi pendukung lainnya. Pendekatan ini tidak memerlukan data rating, riwayat transaksi, maupun interaksi pengguna lain sehingga sesuai untuk kondisi dataset yang tidak memiliki informasi perilaku pengguna.

2.1.2 Content-Based Filtering

Content-Based Filtering (CBF) merupakan pendekatan sistem rekomendasi yang menghasilkan rekomendasi berdasarkan kemiripan antara representasi kebutuhan pengguna dan karakteristik item yang tersedia [5]. Dalam pendekatan ini, setiap item direpresentasikan melalui sekumpulan fitur atau atribut yang menggambarkan karakteristik item tersebut. Selanjutnya, sistem menghitung tingkat kemiripan antara representasi kebutuhan pengguna dan representasi item untuk menghasilkan rekomendasi yang paling relevan.

Pada penelitian ini, setiap laptop direpresentasikan dalam bentuk metadata yang dibentuk dari gabungan atribut spesifikasi, informasi penggunaan, serta hasil *auto-tagging* yang diperoleh melalui proses pengolahan data. Metadata tersebut mencakup informasi seperti merek, prosesor, kartu grafis, kapasitas RAM, media penyimpanan, ukuran layar, harga, karakteristik perangkat, serta kata kunci penggunaan yang relevan dengan kebutuhan pengguna. Representasi metadata ini kemudian digunakan sebagai dokumen dalam proses pembobotan TF-IDF. Tahapan teknis pembentukan metadata dijelaskan pada Bab III karena merupakan bagian dari prosedur penelitian.

Sementara itu, kebutuhan pengguna direpresentasikan sebagai kueri teks yang berisi deskripsi penggunaan laptop, misalnya untuk kuliah, skripsi, pemrograman, pekerjaan kantor, desain grafis, atau bermain gim. Sistem

kemudian membandingkan representasi kueri pengguna dengan metadata setiap laptop menggunakan pembobotan TF-IDF dan perhitungan *Cosine Similarity* untuk memperoleh tingkat kemiripan antar dokumen.

Kelebihan utama CBF adalah tidak bergantung pada data rating, riwayat transaksi, maupun interaksi pengguna lain, sehingga metode ini dapat diterapkan pada sistem yang belum memiliki data historis pengguna [5, 4]. Karakteristik tersebut sesuai dengan penelitian ini yang menggunakan dataset spesifikasi dan metadata laptop sebagai sumber utama proses rekomendasi. Dengan demikian, sistem tetap dapat menghasilkan rekomendasi yang relevan meskipun tidak tersedia informasi perilaku pengguna sebelumnya.

2.1.3 Metadata, Auto-Tag, dan Label

Metadata merupakan representasi karakteristik item yang digunakan oleh sistem rekomendasi berbasis konten. Pada domain laptop, metadata dapat memuat identitas produk, spesifikasi teknis, karakteristik fisik, dan kata kunci penggunaan. Pemilihan atribut metadata mengacu pada atribut laptop yang digunakan dalam penelitian terkait dan disesuaikan dengan ketersediaan data produk [6, 10, 11].

Dalam penelitian ini, *auto-tag* dan label digunakan sebagai bagian dari rekayasa fitur teks. *Auto-tag* memperluas istilah penggunaan agar spesifikasi teknis dapat dicocokkan dengan bahasa sehari-hari pengguna, sedangkan label merangkum karakteristik utama laptop. Aturan pembentukan keduanya dirumuskan oleh peneliti berdasarkan hubungan antara spesifikasi laptop dan kebutuhan penggunaan. Aturan tersebut bukan model klasifikasi terawasi. Rincian atribut dan aturan pembentukannya dijelaskan pada Bab III.

2.1.4 Pemrosesan Teks

Pemrosesan teks (*text preprocessing*) merupakan tahap untuk menyiapkan data tekstual sebelum diubah menjadi representasi numerik. Tahap ini diperlukan karena metadata laptop dan deskripsi kebutuhan pengguna dapat mengandung variasi huruf kapital, simbol, frasa, serta kata umum yang tidak memberikan kontribusi besar terhadap proses pencocokan.

Dalam penelitian ini, pemrosesan teks digunakan pada dua konteks. Pertama, pembersihan atribut spesifikasi untuk mendukung pembentukan metadata. Kedua, pemrosesan metadata laptop dan kueri pengguna sebelum pembobotan TF-

IDF. Tahapan yang digunakan meliputi:

1. **Case Folding**, yaitu mengubah seluruh karakter menjadi huruf kecil agar bentuk kata yang sama tidak dianggap sebagai term berbeda.
2. **Normalisasi Teks dan Frasa Domain**, yaitu menyeragamkan angka, satuan, serta frasa yang mempunyai satu kesatuan makna. Contohnya, "game ringan" dinormalisasi menjadi `game_ringan`, sedangkan "laptop ringan" menjadi `bobot_ringan`.
3. **Tokenization**, yaitu memecah teks menjadi token yang digunakan dalam proses pembobotan. Nama produk dapat dipertahankan sebagai token terpisah sekaligus ditambahkan dalam bentuk token nama utuh.
4. **Stopword Removal**, yaitu menghapus kata umum yang tidak memiliki kemampuan diskriminatif tinggi.
5. **Pembentukan Korpus**, yaitu membentuk kumpulan dokumen yang berasal dari metadata laptop. Kueri pengguna tidak dimasukkan sebagai dokumen tambahan dalam perhitungan IDF.

Penelitian ini tidak menggunakan *stemming* karena istilah teknis, nama seri produk, dan token frasa domain perlu dipertahankan. Metadata dan kueri diproses dengan aturan dasar yang konsisten agar keduanya dapat direpresentasikan pada ruang term yang sama.

2.1.5 Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) adalah teknik pembobotan term yang digunakan untuk mengukur tingkat kepentingan suatu term dalam sebuah dokumen relatif terhadap seluruh dokumen dalam korpus [8, 9]. TF-IDF terdiri atas *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). *Term Frequency* mengukur proporsi kemunculan term dalam suatu dokumen. Persamaan 2.1 menunjukkan perhitungan TF.

$$TF(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2.1}$$

dengan $f_{t,d}$ adalah jumlah kemunculan term t dalam dokumen d, sedangkan penyebut menunjukkan jumlah seluruh kemunculan term pada dokumen tersebut.

Inverse Document Frequency menunjukkan tingkat keunikan term pada korpus. Term yang terdapat pada banyak dokumen memperoleh IDF lebih rendah, sedangkan term yang hanya terdapat pada sedikit dokumen memperoleh IDF lebih tinggi. Persamaan 2.2 menunjukkan perhitungan IDF dengan *smoothing*.

$$IDF(t,D) = \ln\left(\frac{1+N}{1+df_t}\right)+1 \tag{2.2}$$

dengan N adalah jumlah dokumen metadata laptop dalam korpus dan df_t adalah jumlah dokumen yang mengandung term t. Kueri pengguna tidak termasuk dalam nilai N dan tidak mengubah nilai IDF korpus. Bobot TF-IDF diperoleh dari perkalian TF dan IDF sebagaimana ditunjukkan pada Persamaan 2.3.

$$W_{t,d} = TF(t,d) \times IDF(t,D) \tag{2.3}$$

dengan $W_{t,d}$ adalah bobot term t pada dokumen d. Kueri pengguna direpresentasikan sebagai vektor TF-IDF menggunakan term dan nilai IDF yang telah diperoleh dari korpus metadata. Term kueri yang tidak terdapat dalam korpus tidak memiliki dimensi pembanding. Term spesifik dapat memperoleh bobot lebih tinggi apabila muncul pada lebih sedikit dokumen, tetapi besar bobot tetap ditentukan oleh nilai TF dan IDF pada korpus.

2.1.6 Normalisasi Vektor TF-IDF

Sebelum dilakukan perhitungan *Cosine Similarity*, setiap vektor TF-IDF dinormalisasi menggunakan norma Euclidean atau normalisasi L2. Normalisasi dilakukan untuk menghilangkan pengaruh perbedaan skala atau besar geometris vektor sehingga perbandingan lebih berfokus pada arah dan distribusi bobot term.

Normalisasi tersebut tidak menyamakan jumlah token dalam dokumen maupun jumlah term yang memiliki bobot bukan nol. Seluruh vektor tetap berada dalam ruang fitur yang sama, sedangkan nilai setiap komponennya disesuaikan terhadap norma vektor masing-masing. Persamaan 2.4 menunjukkan perhitungan normalisasi L2.

$$\hat{W}_{i,d} = \frac{W_{i,d}}{\sqrt{\sum_{j=1}^{m} W_{j,d}^2}} \tag{2.4}$$

dengan $\hat{W}_{i,d}$ adalah bobot TF-IDF term ke-i pada dokumen d setelah normalisasi, $W_{i,d}$ adalah bobot TF-IDF sebelum normalisasi, dan m adalah jumlah dimensi dalam ruang vektor. Setiap dimensi merepresentasikan satu term dalam kosakata korpus. Setelah normalisasi, setiap vektor bukan nol memiliki norma L2 sebesar satu. Sementara itu, vektor nol tetap tidak dapat dinormalisasi karena tidak memiliki term dengan bobot TF-IDF lebih besar dari nol.

2.1.7 Cosine Similarity

Cosine Similarity mengukur kemiripan antar-vektor berdasarkan kosinus sudut di antara keduanya [8, 15]. Dalam penelitian ini, metode tersebut digunakan untuk membandingkan vektor kueri pengguna dengan vektor metadata laptop. Persamaan 2.5 menunjukkan rumus umum *Cosine Similarity*.

$$\text{cosine}(A,B) = \frac{\sum_{i=1}^{m} A_i B_i}{\sqrt{\sum_{i=1}^{m} A_i^2}\sqrt{\sum_{i=1}^{m} B_i^2}} \tag{2.5}$$

dengan A dan B merupakan vektor pada ruang term yang sama. Untuk vektor berbobot non-negatif, nilainya berada pada rentang 0 sampai 1. Nilai 1 menunjukkan bahwa arah kedua vektor sama atau proporsional, bukan bahwa teks aslinya harus identik.

Karena vektor kueri dan metadata telah dinormalisasi menggunakan L2, penyebut pada Persamaan 2.5 bernilai satu. Oleh karena itu, kemiripan dapat dihitung sebagai hasil kali titik:

$$\text{cosine}(\hat{A},\hat{B}) = \hat{A}\cdot\hat{B} = \sum_{i=1}^{m} \hat{A}_i \hat{B}_i \tag{2.6}$$

Skor *Cosine Similarity* digunakan untuk membandingkan dan mengurutkan laptop, bukan sebagai persentase kecocokan absolut. Nilai yang relatif kecil masih dapat terjadi karena kueri pengguna biasanya singkat, sedangkan metadata laptop memuat lebih banyak term.

2.1.8 Pemeringkatan Rekomendasi Top-k

Setelah nilai kemiripan diperoleh, laptop diurutkan dari skor tertinggi ke skor terendah. Pemeringkatan *Top-k* mengambil sejumlah k item pada urutan teratas sebagaimana ditunjukkan pada Persamaan 2.7.

$$Top_k(q) = \{d_1, d_2, \ldots, d_k\}, \quad sim(q,d_1) \geq sim(q,d_2) \geq \cdots \geq sim(q,d_k) \tag{2.7}$$

dengan q adalah kueri pengguna, d_i adalah dokumen laptop pada urutan ke-i, dan $sim(q,d_i)$ adalah skor kemiripan. Penelitian ini menggunakan $k = 6$ sebagai batas keluaran utama sistem dan titik potong evaluasi. Sebagai analisis tambahan, beberapa nilai k dibandingkan untuk melihat perubahan *Precision@k*, *Recall@k*, dan *F1-Score@k*. Nilai $k = 6$ tetap diperlakukan sebagai keputusan perancangan antarmuka dan tidak dinyatakan sebagai nilai optimal secara universal.

2.1.9 Filter Terstruktur sebagai Fitur Pendukung

Selain pencocokan teks berbasis TF-IDF dan *Cosine Similarity*, sistem menyediakan filter terstruktur berupa harga, berat, ukuran layar, sistem operasi, RAM, penyimpanan, CPU, tipe GPU, panel, dan kualitas layar. Filter digunakan untuk mempersempit kandidat berdasarkan batasan eksplisit pengguna. Tipe GPU dalam konteks filter dibedakan menjadi GPU terintegrasi dan GPU terpisah, sedangkan nama vendor GPU merupakan informasi atribut yang berbeda.

Filter tidak dimasukkan ke dalam vektor TF-IDF. Vektor teks digunakan untuk mengukur kemiripan semantik, sedangkan filter digunakan sebagai persyaratan terstruktur. Sebagai contoh, harga maksimum merupakan batas tegas dan tidak tepat apabila hanya direpresentasikan sebagai term yang memengaruhi skor kemiripan. Rincian penerapan filter dijelaskan pada Bab III.

2.1.10 Evaluasi Relevansi Rekomendasi

Evaluasi relevansi dilakukan dengan membandingkan daftar rekomendasi *Top-k* dengan himpunan item relevan pada *ground truth* [16]. Misalkan R_k adalah himpunan rekomendasi pada posisi *Top-k* dan G adalah himpunan item relevan.

Precision@k mengukur proporsi item relevan pada hasil *Top-k*:

$$Precision@k = \frac{|R_k \cap G|}{k} \tag{2.8}$$

Recall@k mengukur proporsi item relevan yang berhasil ditemukan dari seluruh item relevan:

$$Recall@k = \frac{|R_k \cap G|}{|G|} \tag{2.9}$$

F1-Score@k merupakan rata-rata harmonik *precision* dan *recall*:

$$F1@k = 2 \times \frac{Precision@k \times Recall@k}{Precision@k + Recall@k} \tag{2.10}$$

Nilai metrik berada pada rentang 0 sampai 1. Pada penelitian ini, ketiga metrik diterapkan dengan $k = 6$, sehingga dilaporkan sebagai *Precision@6*, *Recall@6*, dan *F1-Score@6*.

2.1.11 Pembentukan Ground Truth Berbasis Kriteria Relevansi

Ground truth merupakan himpunan item yang digunakan sebagai acuan relevansi. Himpunan tersebut dapat berasal dari penilaian ahli, data interaksi pengguna, atau aturan yang telah ditetapkan sesuai rancangan evaluasi.

Pada penelitian ini, *ground truth* dibentuk menggunakan aturan yang dirumuskan oleh peneliti berdasarkan kesesuaian spesifikasi dan metadata laptop terhadap setiap skenario kebutuhan. Prosesnya tidak dilakukan dengan memilih laptop secara manual satu per satu dan tidak dinyatakan sebagai penilaian ahli. Setiap laptop diperiksa menggunakan aturan yang sama untuk menentukan apakah termasuk dalam himpunan relevan. Rincian aturan disajikan pada Bab III.

Karena aturan memanfaatkan atribut dan metadata yang juga digunakan oleh sistem, hasil evaluasi diinterpretasikan sebagai evaluasi berbasis aturan dalam ruang lingkup penelitian, bukan sebagai ukuran mutlak terhadap seluruh kemungkinan preferensi pengguna.

2.1.12 System Usability Scale (SUS)

System Usability Scale (SUS) merupakan instrumen evaluasi kebergunaan yang diperkenalkan oleh Brooke dan terdiri atas 10 pernyataan dengan skala Likert lima poin [14]. SUS menghasilkan ukuran global mengenai persepsi pengguna terhadap kebergunaan sistem.

Untuk pernyataan bernomor ganjil, skor kontribusi diperoleh dengan mengurangi nilai respons dengan 1. Untuk pernyataan bernomor genap, skor kontribusi diperoleh dengan mengurangi 5 dengan nilai respons. Seluruh skor kontribusi dijumlahkan dan dikalikan 2,5 sebagaimana Persamaan 2.11.

$$SUS_i = \left(\sum_{j \in \{1,3,5,7,9\}} (Q_{ij} - 1) + \sum_{j \in \{2,4,6,8,10\}} (5 - Q_{ij}) \right) \times 2.5 \qquad (2.11)$$

dengan Q_{ij} adalah respons responden ke-i pada pernyataan ke-j. Skor SUS berada pada rentang 0 sampai 100, tetapi bukan persentase. Nilai 68 sering digunakan sebagai nilai pembanding terhadap rata-rata acuan, bukan sebagai batas lulus yang mutlak [17].

Dalam penelitian ini, SUS digunakan untuk mengukur persepsi kebergunaan setelah responden mencoba sistem. Instrumen ini tidak digunakan untuk menentukan benar atau salahnya pilihan laptop pengguna.

2.1.13 Perbandingan SUS dan EUCS

System Usability Scale (SUS) dan *End-User Computing Satisfaction* (EUCS) sama-sama melibatkan penilaian pengguna, tetapi keduanya tidak mengukur konstruk yang sama. SUS digunakan untuk mengukur *usability* atau kebergunaan sistem, sedangkan EUCS digunakan untuk mengukur kepuasan pengguna akhir terhadap informasi dan layanan yang dihasilkan oleh sistem.

EUCS menilai kepuasan pengguna melalui lima dimensi, yaitu isi (*content*), akurasi yang dirasakan (*perceived accuracy*), format, kemudahan penggunaan, dan ketepatan waktu [18]. Oleh karena itu, EUCS banyak digunakan pada penelitian sistem rekomendasi yang ingin mengetahui apakah pengguna merasa puas terhadap isi rekomendasi, penyajian informasi, dan kecepatan sistem dalam memberikan hasil. Penggunaan EUCS pada sistem rekomendasi bukan berarti instrumen tersebut

lebih tepat daripada SUS, tetapi menunjukkan bahwa fokus evaluasi penelitiannya adalah kepuasan terhadap keluaran sistem.

Penelitian ini memilih SUS karena rumusan evaluasi pengguna diarahkan pada tingkat kebergunaan sistem rekomendasi berbasis web. Pengguna perlu menuliskan kebutuhan dalam bahasa sehari-hari, mengatur filter spesifikasi, menjalankan proses pencarian, dan memahami hasil rekomendasi. Dengan demikian, aspek yang perlu dinilai adalah apakah alur tersebut mudah dipelajari, tidak rumit, konsisten, dan dapat digunakan dengan percaya diri. Aspek-aspek tersebut sesuai dengan konstruk yang diukur oleh SUS.

Karakteristik pengguna non-teknis menjadi pertimbangan pendukung dalam pemilihan SUS, tetapi bukan karena pengguna non-teknis dianggap tidak mampu memberikan penilaian terhadap hasil rekomendasi. Pengguna tetap dapat menilai apakah rekomendasi terasa sesuai dengan kebutuhannya. Namun, penilaian tersebut merupakan relevansi atau akurasi yang dirasakan (*perceived relevance* atau *perceived accuracy*), bukan bukti objektif mengenai kinerja algoritma.

Dalam penelitian ini, kualitas rekomendasi telah dievaluasi secara terpisah menggunakan *Precision*, *Recall*, dan *F1-Score* terhadap *ground truth* berbasis aturan. Oleh karena itu, pembagian evaluasi dilakukan sebagai berikut:

1. Kualitas relevansi hasil rekomendasi dievaluasi menggunakan *Precision*, *Recall*, dan *F1-Score*; dan
2. Kebergunaan sistem dari sudut pandang pengguna dievaluasi menggunakan SUS.

Pembagian tersebut membuat setiap instrumen digunakan sesuai dengan tujuan pengukurannya. EUCS tetap dapat digunakan sebagai evaluasi tambahan apabila penelitian ingin mengukur kepuasan pengguna terhadap isi, format, kecepatan, dan akurasi yang dirasakan dari hasil rekomendasi. Namun, EUCS tidak menggantikan evaluasi relevansi objektif, dan pada penelitian ini tidak digunakan karena fokus evaluasi pengguna adalah kebergunaan sistem, bukan kepuasan terhadap keluaran sistem pada setiap dimensi EUCS.

2.2 Penelitian Terdahulu dan Kesenjangan Penelitian

Penelitian terdahulu digunakan untuk memetakan metode, fokus, dan kesenjangan penelitian yang berkaitan dengan sistem rekomendasi laptop, pendekatan berbasis teks, dan evaluasi kebergunaan.

Berbagai penelitian telah mengembangkan sistem rekomendasi laptop menggunakan pendekatan seperti *Content-Based Filtering*, *Cosine Similarity*, pembobotan kriteria, maupun pembelajaran mesin. Namun, sebagian besar penelitian masih berfokus pada proses rekomendasi dan kesesuaian spesifikasi produk, sedangkan evaluasi kebergunaan umumnya diterapkan pada aplikasi e-commerce atau sistem informasi umum.

Selain itu, penelitian yang secara khusus menargetkan pengguna non-teknis melalui kebutuhan berbasis teks sebagai masukan utama serta mengevaluasi relevansi rekomendasi dan kebergunaan sistem secara bersamaan masih terbatas. Oleh karena itu, dilakukan pemetaan penelitian terdahulu sebagaimana disajikan pada Tabel 2.1.

Tabel 2.1. Pemetaan Penelitian Terdahulu dan Kesenjangan Penelitian

Peneliti	**Metode**	**Fokus**	**Kesenjangan**
Pratiwi *et al.* (2025) [6]	CBF berbasis spesifikasi	Rekomendasi laptop berdasarkan kesesuaian spesifikasi.	Belum mengevaluasi kebergunaan bagi pengguna non-teknis.
Rojabi *et al.* (2024) [10]	CBF dan KNN	Rekomendasi laptop berdasarkan kemiripan konten.	Belum menggabungkan evaluasi relevansi dan kebergunaan.
Oktavian dan Amin (2024) [19]	CBF, encoding, dan cosine	Rekomendasi berdasarkan kemiripan spesifikasi.	Masih memerlukan laptop acuan dan belum mendukung masukan teks bebas.
Jadhav (2024) [11]	Pembelajaran mesin berbasis web	Rekomendasi dan perbandingan laptop berbasis web.	Belum mengevaluasi relevansi matematis dan kebergunaan pengguna.
Yustinus (2025) [20]	SAW	Rekomendasi smartphone berdasarkan bobot kriteria.	Pengguna harus menentukan bobot kriteria secara manual.
Mahardika dan Putra (2025) [21]	SUS	Evaluasi kebergunaan aplikasi e-commerce.	Tidak membahas rekomendasi produk teknis.
Penelitian ini	**CBF, TF-IDF, cosine, Top-6, dan SUS**	**Rekomendasi laptop berbasis kebutuhan teks bagi pengguna non-teknis.**	**Menggabungkan evaluasi relevansi dan kebergunaan sistem.**

Berdasarkan Tabel 2.1, penelitian ini mengisi kesenjangan tersebut dengan menerapkan *Content-Based Filtering* berbasis TF-IDF dan *Cosine Similarity* pada metadata laptop, kemudian mengevaluasi kualitas rekomendasi menggunakan *Precision@6*, *Recall@6*, dan *F1-Score@6*, serta mengevaluasi kebergunaan sistem menggunakan *System Usability Scale* (SUS).

2.3 Posisi Penelitian

Penelitian ini diposisikan sebagai pengembangan sistem rekomendasi pemilihan laptop berbasis teks untuk pengguna non-teknis. Berbeda dengan pendekatan yang mengharuskan pengguna menentukan bobot kriteria secara manual, penelitian ini menggunakan deskripsi kebutuhan sebagai masukan utama. Kueri direpresentasikan menggunakan term dan nilai IDF dari korpus metadata laptop, kemudian dibandingkan dengan vektor metadata menggunakan *Cosine Similarity*. Filter terstruktur diterapkan secara terpisah sebagai batasan tambahan.

Hasil rekomendasi utama ditampilkan dalam bentuk *Top-6* dan dievaluasi menggunakan *Precision@6*, *Recall@6*, dan *F1-Score@6*. Pengaruh perubahan nilai k dianalisis secara tambahan pada Bab IV. Nilai $k = 6$ tetap merupakan keputusan perancangan untuk menjaga hasil tetap ringkas, bukan nilai yang dinyatakan optimal secara universal.

Selain evaluasi relevansi, penelitian ini mengevaluasi kebergunaan antarmuka menggunakan SUS. Pemilihan SUS didasarkan pada fokus penelitian terhadap kemudahan interaksi secara keseluruhan. Dengan demikian, posisi penelitian terletak pada integrasi masukan kebutuhan berbasis teks, pemeringkatan berbasis konten, evaluasi relevansi, dan evaluasi kebergunaan dalam konteks pemilihan laptop untuk pengguna non-teknis.