

MENINGKATKAN PENGALAMAN TOKO BAJU BAYI ONLINE: IMPLEMENTASI *PROGRESSIVE WEB APPS* DENGAN REKOMENDASI BERBASIS *COLLABORATIVE FILTERING*

Raihan Jan'smaillendra Suryono, Arief Ichwani

Teknik Informatika, Universitas Esa Unggul

Jl. Arjuna Utara No.9, KebonJeruk, Kota Jakarta Barat, Indonesia

Raihanjansmaillendra.rjs@gmail.com

ABSTRAK

Toko baju bayi online semakin populer di Indonesia, namun masih ditemukan sejumlah kendala yang memengaruhi pengalaman *user*, seperti akses yang lambat, ketergantungan pada koneksi internet, dan rekomendasi produk yang kurang relevan. Permasalahan ini dapat menurunkan kepuasan serta keterlibatan *user* dalam proses belanja. Penelitian ini membantu mencari solusi berupa integrasi *Progressive Web Apps* (PWA) dan sistem rekomendasi berbasis *Collaborative Filtering*. PWA menawarkan fitur seperti instalasi langsung di perangkat, akses *offline*, yang dapat meningkatkan performa dan ketersediaan layanan. Sementara itu, *Collaborative Filtering* memungkinkan sistem memberikan rekomendasi produk berdasarkan preferensi *user* lain yang memiliki pola perilaku serupa. Metode yang digunakan adalah eksperimen untuk mengevaluasi efektivitas integrasi kedua teknologi tersebut pada platform toko baju bayi online. Hasil penelitian menunjukkan bahwa solusi ini berhasil meningkatkan kecepatan akses, kenyamanan *user*, serta relevansi rekomendasi produk. Dengan demikian, penelitian ini berkontribusi dalam pengembangan *e-commerce* di Indonesia melalui pemanfaatan PWA dan sistem rekomendasi sebagai strategi untuk meningkatkan pengalaman dan kepuasan *user*.

Kata kunci: Toko Online, Item-based Collaborative Filtering, CF, Sistem Rekomendasi, PWA

1. PENDAHULUAN

Perkembangan pesat dalam teknologi informasi dan komunikasi telah merubah cara masyarakat mengakses dan memperoleh informasi secara signifikan. Salah satu dampak positif dari kemajuan ini adalah kemudahan dalam berbelanja secara online, khususnya dalam membeli baju bayi. Platform toko baju bayi online kini memberikan kemudahan bagi pembeli untuk menemukan, memilih, dan membeli produk tanpa harus meninggalkan kenyamanan rumah. Meskipun demikian, terdapat tantangan yang masih perlu diatasi untuk meningkatkan pengalaman *user*, seperti kesulitan dalam menemukan produk yang sesuai dengan preferensi dan kurangnya interaktivitas yang dapat memperkaya pengalaman berbelanja online.

Penerapan sistem rekomendasi berbasis *Collaborative Filtering* dapat meningkatkan relevansi produk dan pengalaman *user* dengan menyediakan rekomendasi yang sesuai berdasarkan pola pembelian dan minat *user* lain. Penelitian ini bertujuan untuk mengeksplorasi bagaimana penerapan PWA dan sistem rekomendasi berbasis *Collaborative Filtering* dapat memperbaiki pengalaman *user* dalam toko baju bayi online.

2. TINJAUAN PUSTAKA

2.1. Collaborative Filtering

Collaborative filtering (CF) adalah salah satu teknik rekomendasi yang paling sukses dan efektif. Teknik ini banyak digunakan untuk memprediksi preferensi *user* pada item yang belum diberi peringkat dengan menganalisis perilaku peringkat historis mereka dalam sistem rekomendasi, terutama di area

bisnis[1]. *Collaborative filtering* tidak membutuhkan pengetahuan tentang *user* atau *item*, melainkan membuat rekomendasi berdasarkan interaksi di antara keduanya. Meskipun efisien dan sederhana, teknik ini memiliki beberapa masalah, seperti *cold start*, akurasi prediksi, dan ketidakmampuan untuk menangkap interaksi yang kompleks antara *user* dan *item*[2]. *Cold start*, Masalah ini terjadi ketika *user/item* baru ditambahkan ke dalam matriks *rating*[3], *Sparsity*, data dalam kondisi jarang; hal ini disebabkan karena matriks data yang berisi tidak lengkap atau banyak ditemukan kekosongan data, *Scalability*, sistem rekomendasi perlu meningkatkan daya komputasi mereka untuk menawarkan rekomendasi yang tepat waktu[4].

Perbedaan mendasar antara User-based CF dengan Item-based CF adalah korelasi antara *user* dan *item*. Model ICF memiliki skema yang hampir sama dengan UCF. UCF yang dicari sebelumnya adalah korelasi antar *user*. ICF dicari antara *item user*, *item* yang berkorelasi direkomendasikan terhadap beberapa *user* lain[5].

2.2. Cosine Similarity

Persamaan *adjusted cosine Similarity* digunakan untuk menghitung nilai kemiripan antar *item*. Perhitungan kemiripan ini merupakan modifikasi dari perhitungan kemiripan dengan melihat setiap *user* memiliki skema *rating* yang berbeda-beda[6]. Berikut rumus perhitungan *Adjusted Cosine Similarity*.

$$sim(i, j) = \frac{\sum u \in U_{ij} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum u \in U_i (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum u \in U_j (R_{u,j} - \bar{R}_u)^2}}$$

Keterangan:

$sim(i, j)$ adalah *similarity* antara *item i* dan *j*.

u adalah notasi *user*.

U Ini adalah himpunan *user* yang telah memberi *rating* pada *item i* dan *j*.

$R_{u,i}$ Ini adalah *rating* yang diberikan *user u* untuk *item i*.

$R_{u,j}$ Ini adalah *rating* yang diberikan *user u* untuk *item j*.

$\bar{R}_u$ adalah nilai rata-rata *rating user u*.

2.3. Penelitian Terdahulu

Berdasarkan Penelitian “An optimized item-based Collaborative Filtering algorithm” berhasil mengembangkan algoritma kolaboratif berbasis item yang lebih efektif dalam mengatasi masalah *cold-start* dan data *sparsity* pada sistem rekomendasi. Dengan menggabungkan dan menyeimbangkan tiga metode pengukuran *Similarity* tradisional—*Cosine*, *Pearson*, dan *Adjusted Cosine*—dengan menambahkan faktor penyeimbang yang berbasis nilai α , algoritma ini mampu meningkatkan akurasi rekomendasi tanpa mengurangi kekuatan dari metode asli. Pengujian menggunakan dataset *Movielens* menunjukkan bahwa pendekatan ini mampu menurunkan tingkat error rata-rata dan memberikan rekomendasi yang lebih relevan[7].

Berdasarkan Penelitian “Recommender System with User-Based and Item-Based Collaborative Filtering on Twitter using K-Nearest Neighbors Classification” menggabungkan metode *Collaborative Filtering* dan *K-Nearest Neighbors (KNN)* menggunakan data dari Twitter serta *rating* tambahan dari IMDB, Rotten Tomatoes, dan Metacritic, diperoleh hasil bahwa metode *item-based* menghasilkan RMSE lebih rendah (0.5449) dibanding *user-based* (1.8244). Pengujian KNN dengan nilai $K = 3$ hingga 10 menunjukkan hasil optimal pada $K = 3$, dengan *precision* dan *recall* tertinggi masing-masing 91.22% dan 91.07% untuk *user-based*, serta 89.44% dan 91.22% untuk *item-based*. Kombinasi metode ini efektif dalam menentukan film yang direkomendasikan dan tidak direkomendasikan berdasarkan nilai *precision* dan *recall* terbaik[8].

3. METODE PENELITIAN

Metode penelitian yang digunakan pada penelitian ini adalah

3.1. Metode pengambilan data

a. Wawancara

Wawancara merupakan metode pengumpulan data yang melibatkan antara pewawancara dan responden untuk mendapatkan deskripsi dan pemahaman mendalam mengenai tujuan penelitian.

b. Observasi

Observasi merupakan cara atau teknik pengumpulan data dengan melakukan

pengamatan langsung terhadap suatu obyek yang ingin diselidiki.

c. Studi literatur

Metode yang dilakukan dengan cara mengkaji dan menganalisis literatur, jurnal, artikel, buku, dan sumber-sumber lain yang relevan dengan topik penelitian yang sedang dibahas.

3.2. Agile

metode *Agile* mengatasi dua masalah mendasar dalam *Software Development* yaitu kecepatan tinggi yang diperlukan untuk mengembangkan perangkat lunak saat ini dan wawasan yang sesuai dengan persyaratan asli adalah aturannya, bukan pengecualian[9].

3.3. Scrum

Pertama, daftar tugas yang harus dilakukan yang disebut *product backlog* dibuat dan dipelihara. Kemudian *item* prioritas utama adalah dipilih untuk periode waktu yang tetap dan teratur disebut *sprint* di mana tim *Scrum* berusaha untuk mencapai tujuan yang telah ditentukan dan disepakati bersama[10].

Tahapan dari metode *Scrum* seperti *requirements*, fitur, fungsi, dan spesifikasi yang harus dimiliki oleh sebuah produk. *product backlog*, rancangan daftar fitur yang dinamis dan diprioritaskan untuk produk, dengan mengutamakan fitur-fitur yang perlu dikembangkan. *sprint planning*, proses yang bertujuan untuk merinci apa yang dapat diselesaikan dalam suatu *sprint* dan bagaimana pekerjaan tersebut akan dilaksanakan. *sprint*, mengerjakan tugas-tugas yang telah ditetapkan dalam *sprint planning*. *sprint review*, mengevaluasi apakah tujuan *sprint* telah tercapai dan mendapatkan *feedback* dari *stakeholder*. *retrospective process*, mendiskusikan apa yang telah berjalan baik, apa yang dapat ditingkatkan, dan langkah-langkah perbaikan.

4. HASIL DAN PEMBAHASAN

Hasil penelitian akan disajikan dengan menggunakan kerangka metode *scrum*, sebagai berikut

4.1. Requirement

Pada tahap ini, Identifikasi kebutuhan ini bertujuan untuk menentukan komponen-komponen yang diperlukan dalam membangun website, baik dari sisi pengguna maupun administrator. Analisis ini mencakup kebutuhan fungsional dan non-fungsional sebagai berikut:

a. Analisis Kebutuhan Fungsional

Kebutuhan fungsional mencakup fitur-fitur utama yang harus dimiliki sistem agar pengguna dapat berinteraksi secara optimal. Kebutuhan fungsional yang telah diidentifikasi seperti autentikasi dan *user role*, *user* dapat melakukan pendaftaran akun (*register*), *login*, dan *logout*, sementara admin memiliki akses untuk mengelola data, *data management*, admin dapat mengelola data

pengguna dan produk, termasuk menambah, mengubah, dan menghapus data, *order history*, *user* dapat melihat riwayat transaksi mereka, rekomendasi produk, *user* mendapatkan rekomendasi produk berdasarkan pembelian sebelumnya, sales, admin dapat memantau dan mengecek data penjualan, *user* dapat menentukan ukuran barang yang akan dibeli dan jumlah barang yang akan dibeli,

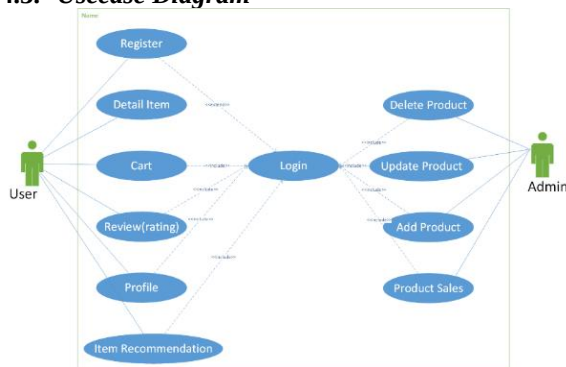
b. Analisis Kebutuhan non-Fungsional

Kebutuhan non-fungsional menggambarkan batasan, karakteristik, dan properti sistem, baik selama proses pengembangan maupun saat operasional. Adapun kebutuhan non-fungsional yang diidentifikasi seperti aplikasi dapat dijalankan di *windows*, *linux*, *android*, *ios* dan *macos*. *user* tidak dapat mendaftarkan dengan akun email yang sama, *user* dapat memberikan *rating* dalam bilangan bulat (satu bintang, dua, tiga, empat dan lima bintang) *user* harus *login* untuk mengakses data pribadi, *user* harus login untuk menulis *review* produk yang telah dibeli. *user* bisa menulis *review* setelah membeli produk, *user* dapat melakukan transaksi setelah *login*.

4.2. Design

UML, atau Unified Modelling Language, adalah suatu bahasa pemodelan visual yang digunakan untuk merancang dan mendokumentasikan sistem perangkat lunak. UML menyediakan notasi dan metode standar yang memungkinkan pengembang perangkat lunak untuk menggambarkan, merancang, dan menyusun struktur dan interaksi dalam suatu sistem.

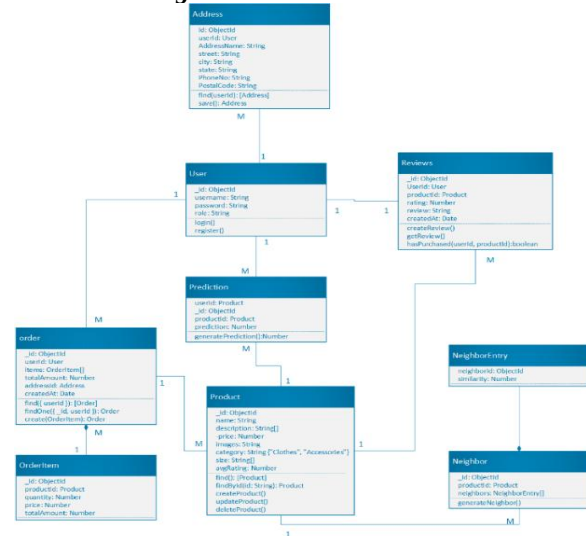
4.3. Usecase Diagram



Gambar 1. Usecase Diagram

Berdasarkan Usecase Diagram diatas, Terdapat user, di dalam sistem,terdapat enam kasus penggunaan untuk user, seperti Register, Detail item, Cart, Review (Rating), Profile, dan Item Recommendation. Setiap kasus penggunaan terhubung dengan aktor yang terlibat melalui garis. Garis putus-putus dengan "«include»" menunjukkan bahwa Login merupakan prasyarat untuk beberapa kasus penggunaan, seperti mengakses Cart dan menulis reviews dan «extend» menunjukkan bahwa setelah melakukan register user dapat melakukan login.

4.4. Class Diagram

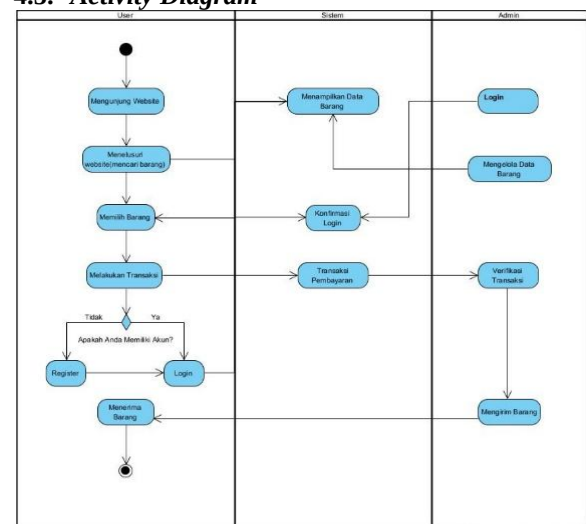


Gambar 2. *Class Diagram*

Keterangan:

- *product*, sebagai produk yang dijual di toko baju bayi *online*.
- *users*, sebagai pelanggan yang dapat melakukan pembelian.
- *order*, sebagai pesanan yang dibuat pelanggan.
- *address*, sebagai alamat tujuan pengiriman.
- *neighbor*, sebagai tempat menyimpan hasil kalkulasi untuk *neighbor*.
- *prediction*, sebagai menampung nilai prediksi *item*
- *reviews*, sebagai *review* yang diberikan *user*

4.5. Activity Diagram



Gambar 3. *Activity* Diagram

4.6. Perhitungan Kemiripan Item (Cosine Similarity)

Sebagai contoh proses perhitungan, tabel berikut ini menunjukkan data *rating* dari 5 pengguna terhadap 5 *item* produk, seperti Tas Popok Bayi, Handuk Mandi Bayi, dan produk lainnya.

4.7. Product Backlog

Berdasarkan Requirement diatas, product backlog sebagai berikut:

Table 1. Product Backlog

Feature	Prioritas	Jenis
Autentikasi dan user role	High	Fungsional
Manajemen Data (Admin)	High	Fungsional
Detail produk	High	Fungsional
Order History	Medium	Fungsional
Rekomendasi produk	Medium	Fungsional
Sales report	Medium	Fungsional
Ukuran dan jumlah produk	High	Fungsional
Lokasi pengiriman	High	Fungsional
Katalog produk	High	Fungsional
Fitur Cart	Medium	Fungsional
Multi-platform (Windows, Linux, Android, iOS, macOS) (PWA)	Medium	Non-fungsional
Email unik untuk registrasi	High	Non-fungsional
Rating dengan bilangan bulat	High	Non-fungsional
Login untuk akses data pribadi	High	Non-fungsional
Login sebelum menulis review	High	Non-fungsional
Menulis review setelah membeli barang	High	Non-fungsional
Transaksi bisa dilakukan setelah login	High	Non-fungsional

4.8. Perhitungan Kemiripan Item (Cosine Similarity)

Sebagai contoh proses perhitungan, tabel berikut ini menunjukkan data *rating* dari 5 pengguna terhadap 5 item produk, seperti Tas Popok Bayi, Handuk Mandi Bayi, dan produk lainnya. Dengan menggunakan *Cosine Similarity*, tujuan utamanya adalah untuk menemukan item yang memiliki kemiripan tertinggi dengan Jaket musim dingin bayi, sehingga rekomendasi dapat diberikan berdasarkan korelasi antar item.

Table 2. Matrix User dan Item

User	Baby Bibs	Dot Bayi	Baby Teething Toys	Sun Hat Bayi	Selimut Bedong
olivia999	3		4		4
Budi Santoso		5	3	3	4
Siti Rahayu	3	4	4	3	
Anita Wijaya	4	4	2	4	5
Ahmad Firdaus	4	4		3	3

$$sim_{(baby\ bibs, dot\ bayi)} = \frac{A \cdot B}{\|A\| \|B\|}$$

$$sim_{(baby\ bibs, dot\ bayi)} = \frac{(3 \times 4) + (4 \times 4) + (4 \times 4)}{\sqrt{(3)^2 + (4)^2 + (4)^2} \sqrt{(4)^2 + (4)^2 + (4)^2}}$$

$$sim_{(baby\ bibs, dot\ bayi)} = \frac{12 + 16 + 16}{\sqrt{41} \sqrt{48}}$$

$$sim_{(baby\ bibs, dot\ bayi)} = \frac{44}{\sqrt{1968}} = 0.991$$

Berdasarkan contoh perhitungan *cosine similarity* diatas, maka didapat hasil nilai *Similarity* sebagai berikut

Table 3. Similarity antar item

Similarity antar item	Nilai Similarity
$sim_{(baby\ bibs, dot\ bayi)}$	0.991
$sim_{(baby\ bibs, baby\ teething\ toys)}$	0.914
$sim_{(baby\ bibs, sun\ hat\ bayi)}$	0.991
$sim_{(baby\ bibs, selimut\ bedong)}$	0.972
$sim_{(dot\ bayi, baby\ teething\ toys)}$	0.958
$sim_{(dot\ bayi, sun\ hat\ bayi)}$	0.982
$sim_{(dot\ bayi, selimut\ bedong)}$	0.973
$sim_{(baby\ teething\ toys, sun\ hat\ bayi)}$	0.924
$sim_{(baby\ teething\ toys, selimut\ bedong)}$	0.935
$sim_{(sun\ hat\ bayi, selimut\ bedong)}$	0.994

4.9. Menghitung Nilai Prediksi (Weighted Sum)

Setelah mendapatkan nilai *similarity*, Langkah selanjutnya adalah menghitung nilai prediksi *rating* yang belum di berikan user menggunakan rumus *weighted sum*, berikut adalah contoh perhitungan nilai *weighted sum* berdasarkan *similarity* diatas.

$$P(u, i) = \frac{\sum_{i \in I} (R_{u,i} * S_{i,j})}{\sum_{i \in I} |S_{i,j}|}$$

$$P_{(budi\ santoso, baby\ bibs)} = \frac{(0.991 \times 5) + (0.914 \times 3) + (0.991 \times 3) + (0.972 \times 4)}{0.991 + 0.914 + 0.991 + 0.972}$$

$$P_{(budi\ santoso, baby\ bibs)} = \frac{(4.955) + (2.742) + (2.973) + (3.888)}{(4.955) + (2.742) + (2.973) + (3.888)}$$

$$P_{(budi\ santoso, baby\ bibs)} = \frac{0.991 + 0.914 + 0.991 + 0.972}{14.558} = 3.76$$

Dari hasil perhitungan *weighted sum* diatas ditemukan bahwa nilai prediksi *rating* dari baby bibs adalah 3.76 yang lalu dibulatkan keatas menjadi 4. Hasil perhitungan *weighted sum*, dapat dihasilkan sebagai berikut:

Table 4. Nilai Prediksi

User	Baby Bibs	Dot Bayi	Baby Teething Toys	Sun Hat Bayi	Selimut Bedong
olivia999	3	4(3.6)	4	4(3.66)	4
Budi Santoso	4(3.76)	5	3	3	4
Siti Rahayu	3	4	4	3	4(3.49)
Anita Wijaya	4	4	2	4	5
Ahmad Firdaus	4	4	4(3.50)	3	3

4.10. Evaluasi Kinerja Sistem Rekomendasi

Langkah terakhir, untuk mengetahui akurasi rekomendasi sistem, pengujian dilakukan berdasarkan hasil yang diperoleh setelah user melakukan order dan memberikan *rating* untuk item tersebut. Pertama menggunakan *Mean Absolute Error (MAE)* untuk

mengukur rata-rata kesalahan absolut antara *rating* yang diprediksi dan *rating* sebenarnya.

$$MAE = \frac{\sum_{i=1}^n |r_{ui} - \hat{r}_{ui}|}{n}$$

$$MAE = \frac{|4 - 3.76| + |4 - 3.66| + |4 - 3.50| + |4 - 3.66| + |4 - 3.49|}{5} \times 100\%$$

$$MAE = \frac{0.24 + 0.34 + 0.50 + 0.34 + 0.51}{5}$$

$$MAE = \frac{1.93}{5} = 0.386$$

Berdasarkan hasil dari perhitungan MAE diatas menghasilkan akurasi atau besar error pada nilai

prediksi sebesar 0.386 dan untuk *Mean Absolute Percentage Error* (MAPE) sebesar 9.65% yang menyatakan prediksi baik.

4.11. Pengujian Nilai Similarity

Data yang digunakan untuk pengujian terdiri dari 15 *user*, 16 *item*, dan 168 *rating* produk. *Rating* tersebut diberikan oleh *user* yang telah memberikan *review* dan *rating* pada 10 *item*.

Table 2. Nilai Similarity Antara Item

Item	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1.00	0.78	0.70	0.61	0.58	0.70	0.59	0.74	0.76	0.72	0.70	0.82	0.76	0.77	0.62	0.59
2	0.78	1.00	0.60	0.70	0.57	0.69	0.56	0.66	0.70	0.71	0.53	0.66	0.63	0.75	0.76	0.78
3	0.70	0.60	1.00	0.69	0.57	0.60	0.55	0.69	0.52	0.63	0.64	0.85	0.72	0.58	0.57	0.48
4	0.61	0.70	0.69	1.00	0.50	0.56	0.64	0.52	0.62	0.67	0.64	0.78	0.58	0.64	0.75	0.78
5	0.58	0.57	0.57	0.50	1.00	0.77	0.72	0.67	0.59	0.63	0.59	0.51	0.47	0.73	0.75	0.51
6	0.70	0.69	0.60	0.56	0.77	1.00	0.70	0.91	0.82	0.58	0.70	0.71	0.65	0.70	0.55	0.73
7	0.59	0.56	0.55	0.64	0.72	0.70	1.00	0.68	0.72	0.77	0.66	0.59	0.60	0.58	0.49	0.43
8	0.74	0.66	0.69	0.52	0.67	0.91	0.68	1.00	0.77	0.59	0.68	0.70	0.81	0.70	0.47	0.63
9	0.76	0.70	0.52	0.62	0.59	0.82	0.72	0.77	1.00	0.63	0.83	0.71	0.78	0.57	0.50	0.66
10	0.72	0.71	0.63	0.67	0.63	0.58	0.77	0.59	0.63	1.00	0.73	0.66	0.70	0.79	0.66	0.47
11	0.70	0.53	0.64	0.64	0.59	0.70	0.66	0.68	0.83	0.73	1.00	0.81	0.69	0.58	0.47	0.48
12	0.82	0.66	0.85	0.78	0.51	0.71	0.59	0.70	0.71	0.66	0.81	1.00	0.67	0.66	0.56	0.59
13	0.76	0.63	0.72	0.58	0.47	0.65	0.6	0.81	0.78	0.70	0.69	0.67	1.00	0.60	0.46	0.55
14	0.77	0.75	0.58	0.64	0.73	0.70	0.58	0.70	0.57	0.79	0.58	0.66	0.60	1.00	0.83	0.60
15	0.62	0.76	0.57	0.75	0.75	0.55	0.49	0.47	0.50	0.66	0.47	0.56	0.46	0.83	1.00	0.69
16	0.59	0.78	0.48	0.78	0.51	0.73	0.43	0.63	0.66	0.47	0.48	0.59	0.55	0.60	0.69	1.00

Tabel berikut menunjukkan nilai kemiripan antara *item-item* yang diuji. Nilai kemiripan berada dalam rentang 0 hingga 1, di mana 0 berarti tidak ada kemiripan sama sekali antara dua *item*, sementara 1 menunjukkan kemiripan yang sempurna. Data urutan *item* yang diuji ditampilkan di baris pertama dan kolom pertama tabel dengan angka 1 hingga 16. Sebagai contoh, angka 1 pada *item* 1 menunjukkan bahwa baris atau kolom tersebut menggambarkan nilai kemiripan antara *item* pertama dengan *item-item* lainnya. Nilai kemiripan tertinggi dalam baris atau

kolom tersebut mengidentifikasi *item* yang paling mirip (*neighbor* terdekat) dengan *item* yang dimaksud.

Misalnya, jika kita melihat baris untuk *item* 1, kita dapat melihat bahwa *item* tersebut memiliki kemiripan tertinggi dengan *item* 12 (nilai 0,82). Ini berarti *item* 12 adalah *item* yang paling mirip dengan *item* 1 dibandingkan dengan *item-item* lainnya, menurut data kemiripan yang dihitung. Kemudian, nilai kemiripan terdekat ini digunakan untuk perhitungan prediksi *rating* sebagai bagian dari hasil rekomendasi.

4.12. Pengujian Nilai Prediksi

Table 3. Hasil Prediksi Rating

	Table 1: Mean + Feature Rating																
	Item																
User	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	1	3.6	3	3.5 2	2	3.6 9	5	3.6 1	3	4	3.5 5	4	3	3.6 1	3.6	3.5 7	5
	2	4	3.99	4.04	4	5	3.95	4	3.96	3.92	5	3	3.95	3.97	4	5	2
	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	4	2	4	3	4	4	4	3.43	3	3.4	3.44	1	3	3.37	4	5	5
	5	5	4	2	3.72	4	3	5	3	5	4	5	2	3	3.75	3.76	3.73
	6	3	4	5	2	5	4	5	3	3.66	4	3.73	4	3.68	4	4	1
	7	5	4	4.08	5	3	4	4	4.06	5	3	3	4	4.09	4	5	4
	8	3.4	2	3.43	3.43	4	4	4	3	4	4	3	3.44	3	4	3	3.4
	9	4	3.8	4	4	3.78	3	4	4	4	2	4	5	5	3.79	3.77	3
	10	3.4	3.25	4	4	4	4	5	4	4	3.39	3	3	3	3.28	1	1
	11	5	4.05	4	4.01	3	5	1	5	2	4	5	5	4	5	4.04	4.07
	12	5	4	3	3	3.31	3.28	1	3	2	3	3	4	3	5	4	3.38
	13	3	5	3.65	3	3.83	4	5	5	3	4	3.66	1	4	4	3.75	4
	14	5	4	1	4.24	5	5	4.3	5	4	4.25	4.25	4.18	4	4	5	5
15	4	5	4	5	3.88	3.88	3.91	3.89	3	4	3	5	4	3	4	3	

Table menunjukkan hasil nilai prediksi *rating* yang dihitung menggunakan metode *weighted sum* dari nilai *Similarity* antara *item-item* yang ada. Pada tabel di atas, *user* 1 diprediksi akan memberikan *rating* sebesar 3,6 untuk *item* 1, sedangkan untuk *item* 5, *user* 1 diprediksi akan memberikan *rating* sebesar 3,69. Hasil ini menunjukkan bahwa *item* 5 cenderung lebih disukai oleh *user* 1 dibandingkan *item* 1.

4.13. Pengujian Nilai Mean Absolute Error (MAE)

Pengujian MAE bertujuan untuk mengukur sejauh mana kesalahan dalam hasil rekomendasi yang diberikan. Nilai MAE dihitung dengan membandingkan *rating* yang diprediksi oleh sistem dengan *rating* sebenarnya.

Table 4. Nilai Mean Absolute Error (MAE) semua Item

Item	Jumlah Rating	MAE
item 1	10	0.27
item 2	11	0.18
item 3	10	0.33
item 4	11	0.24
item 5	9	0.17
item 6	11	0.09
item 7	10	0.18
item 8	11	0.1
item 9	11	0.1
item 10	11	0.22
item 11	11	0.13
item 12	10	0.15
item 13	10	0.23
item 14	10	0.16
item 15	11	0.2
Rata-rata Nilai MAE		0.19

Tabel di atas menyajikan hasil perhitungan MAE untuk setiap *item* berdasarkan jumlah *rating* yang ada. Misalnya, *item* 1 memiliki nilai MAE sebesar 0,27 dari 10 *rating* yang tersedia, menunjukkan tingkat kesalahan dalam prediksinya. Nilai MAE yang lebih rendah mengindikasikan prediksi yang lebih akurat, sedangkan nilai MAE yang lebih tinggi menunjukkan prediksi yang kurang akurat.

Rata-rata nilai MAE untuk semua *item* dalam pengujian ini adalah 0,19, yang mencerminkan akurasi keseluruhan sistem rekomendasi yang digunakan. Dengan MAE yang relatif rendah, sistem ini dianggap memberikan rekomendasi yang cukup akurat bagi pengguna, meskipun beberapa *item*, seperti *item* 3 dengan MAE sebesar 0,33, menunjukkan tingkat akurasi yang lebih rendah.

4.14. Sprint Planning

Pada tahap ini dilakukan perencanaan sprint berdasarkan product backlog yang telah disusun sebelumnya. Berikut adalah daftar tugas yang akan dikerjakan:

- Setup project dan fitur autentikasi, pada tahap ini membuat setup project seperti, menginstall *package*, mengatur struktur file, menyiapkan data

produk, mengatur konfigurasi untuk *project* dan tampilan dasar(navigasi dan *layout*).

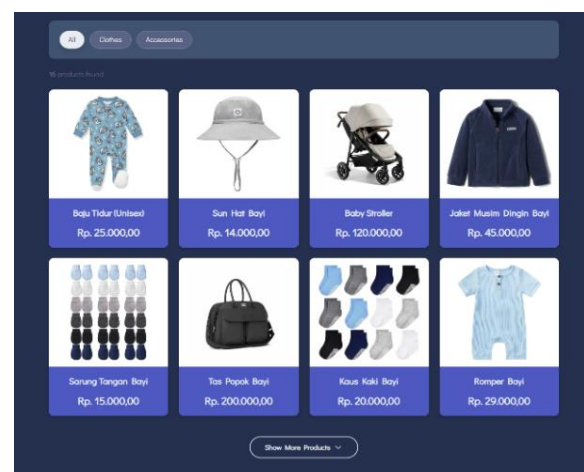
- Mengimplementasikan tampilan katalog dan detail produk, halaman katalog produk yang menampilkan daftar produk, dan detail informasi mengenai produk yang dipilih *user*
- Mengimplementasikan fitur *cart*, *user* dapat menambahkan, menghapus, mengubah produk yang ingin dibeli ke dalam *cart*.
- Mengimplementasikan fitur *management* data dan *sales report* untuk admin, fitur ini dibuat khusus untuk admin agar dapat mengelola data *user* dan produk, termasuk menambah, mengubah, dan menghapus data.
- Mengimplementasikan *Order history* dan *review*, *user* dapat melihat transaksi yang telah dilakukan seperti daftar transaksi, id transaksi, rincian setiap produk, total pembayaran dan *user* dapat menulis *review* dan melihat *review* dari pembeli lainnya mengenai produk.
- Rekomendasi fitur, membuat fitur rekomendasi produk berdasarkan produk yang telah *user* beli dan *review* menggunakan metode *item-based collaborative filtering*.

4.15. Setup project dan fitur autentikasi

Pada fase ini, aplikasi dibangun menggunakan Framework Next.js versi 14, dengan JavaScript sebagai bahasa pemrogramannya. Untuk penyimpanan data, digunakan MongoDB sebagai database.

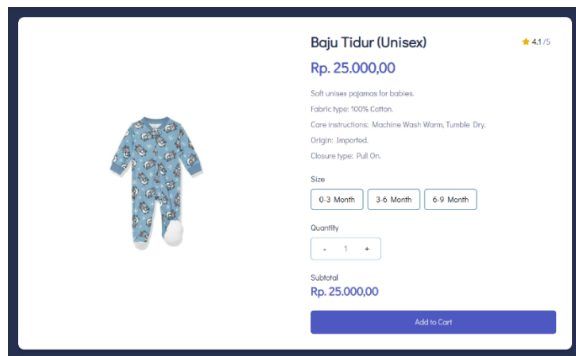
4.16. Implementasi katalog dan detail produk

Pada katalog produk berisi kategori produk, dan semua produk yang dengan nama produk, harga, dan jumlah stok yang tersedia. Berikut implementasinya.



Gambar 4. UI katalog produk

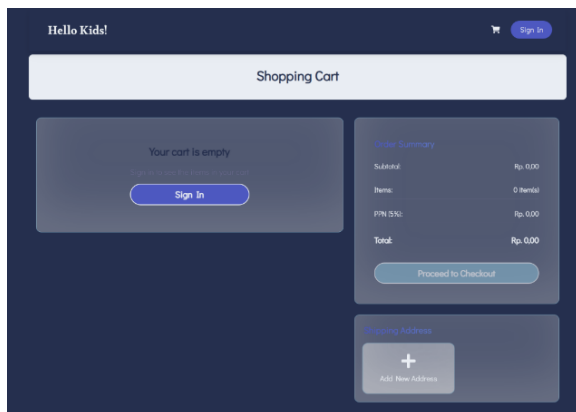
Pada Detail produk, berisi semua informasi yang diperlukan *user* untuk mengidentifikasi produk yang ingin dibeli seperti, nama produk, harga, stok, deskripsi produk, dan ukuran produk.



Gambar 5. UI Detail Produk

4.17. Implementasi fitur Cart

Dalam Halaman cart, terdapat banyak fitur seperti, mengubah jumlah barang yang dibeli memilih dan menambahkan Alamat tujuan. Berikut hasil implementasinya.



Gambar 1. UI Cart

4.18. Implementasi Data Management

Data management hanya dapat diakses oleh admin pada halaman admin. Admin dapat mengubah dan menghapus data user dan data produk. Berikut implementasinya. Pada halaman product management, admin dapat menambah, mengubah dan menghapus data produk.

4.19. Implementasi Sales Report



Gambar 7. Halaman Sales report

Pada halaman ini admin dapat melihat penjualan produk teratas, order produk terbaru, produk teratas

berdasarkan pendapatan, penjualan harian dan pendapatan dalam jangka waktu tertentu seperti, *all time*, setahun terakhir, sebulan kemarin, dan 7 hari terakhir.

4.20. Implementasi Order History

User dapat mengakses semua transaksi produk yang telah dibeli. User dapat melihat nomor resi pembelian, detail produk seperti tanggal pembelian, produk yang dibeli jumlah produk yang dibeli harga produk, subtotal dan Alamat pengiriman produk.

4.21. Implementasi Review

Fitur ini membantu user dalam melihat review dan rating yang telah dibuat oleh user lainnya, dan user hanya dapat menulis review dan memberikan rating setelah membeli produk.

4.22. Implementasi sistem rekomendasi

Pada halaman Detail produk terdapat bagian rekomendasi yang merekomendasikan produk lain yang sesuai dengan produk yang telah dibeli user.

4.23. Sprint Review

Setelah fitur selesai diimplementasikan, dilakukan *sprint review* untuk mengevaluasi apakah sudah sesuai dengan kebutuhan user dan *backlog* yang telah direncanakan sebelumnya. Evaluasi ini dilakukan untuk memastikan fungsionalitas, *User Interface (UI)*, dan performa aplikasi sistem secara keseluruhan. Proses evaluasi yang dilakukan sebagai berikut: Pengujian terhadap fitur yang telah dibangun seperti, autentikasi, katalog produk, manajemen data, *order history*, sistem review produk dan sistem rekomendasi.

4.24. Retrospective Process

Pada tahap ini dilakukan evaluasi terhadap proses pengembangan sistem yang telah dilaksanakan selama siklus sprint berlangsung. Hal-hal yang berjalan dengan baik, Penggunaan *framework Next.js* terbukti efektif dalam membangun aplikasi web yang terstruktur dan responsif. Fitur seperti *routing* dan *API handler* sangat membantu dalam pengelolaan antarmuka dan logika *backend* secara terpadu. Penggunaan *Tailwind CSS* mempercepat proses pengembangan antarmuka pengguna karena kemudahan dalam menerapkan gaya secara langsung melalui kelas-kelas *utility*.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil analisis dan implementasi dalam mengembangkan Progressive Web Apps (PWA) untuk toko baju bayi online dengan fitur rekomendasi berbasis item-based Collaborative Filtering, diperoleh nilai rata-rata Mean Absolute Error (MAE) sebesar 0.19, yang menunjukkan tingkat kesalahan prediksi yang rendah dan akurasi tinggi dalam memberikan rekomendasi produk sesuai preferensi pengguna. Sistem ini menghitung kesamaan

antar item menggunakan cosine similarity dan memprediksi rating dengan metode weighted sum, sehingga mampu mempersonalisasi pengalaman belanja secara efektif. Untuk pengembangan aplikasi selanjutnya, disarankan untuk meningkatkan aspek UI/UX agar aplikasi tidak hanya fungsional, tetapi juga menarik dan mudah digunakan, serta menambahkan fitur tambahan guna semakin meningkatkan pengalaman pengguna.

DAFTAR PUSTAKA

- [1] J. Guo, J. Deng, X. Ran, Y. Wang, and H. Jin, "An efficient and accurate recommendation strategy using degree classification criteria for item-based collaborative filtering," *Expert Syst Appl*, vol. 164, Feb. 2021, doi: 10.1016/j.eswa.2020.113756.
- [2] N. Nassar, A. Jafar, and Y. Rahhal, "A novel deep multi-criteria collaborative filtering model for recommendation system ☆," vol. 187, p. 104811, 2020, doi: 10.1016/j.knosys.
- [3] M. Srifi, A. Oussous, A. A. Lahcen, and S. Mouline, "Recommender systems based on collaborative filtering using review texts-A survey," Jun. 01, 2020, *MDPI AG*. doi: 10.3390/INFO11060317.
- [4] Mohamad Fahmi Hafidz and Sri Lestari, "Solution to Scalability and Sparsity Problems in Collaborative Filtering using K-Means Clustering and Weight Point Rank (WP-Rank)," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 4, pp. 743–750, Aug. 2023, doi: 10.29207/resti.v7i4.4543.
- [5] A. Yusmar, L. K. Wardhani, and H. B. Suseno, "RESTAURANT RECOMMENDER SYSTEM USING ITEM BASED COLLABORATIVE FILTERING AND ADJUSTED COSINE ALGORITHM SIMILARITY," *JURNAL TEKNIK INFORMATIKA*, vol. 14, no. 1, pp. 93–100, Sep. 2021, doi: 10.15408/jti.v14i1.21102.
- [6] S. Devi Nurhayati and W. Widayani, "Sistem Rekomendasi Wisata Kuliner di Yogyakarta dengan Metode Item-Based Collaborative Filtering Yogyakarta Culinary Recommendation System with Item-Based Collaborative Filtering Method," 2021. [Online]. Available: <https://manganenakyog.my.id/>,
- [7] C. Ajaegbu, "An optimized item-based collaborative filtering algorithm," *J Ambient Intell Humaniz Comput*, vol. 12, no. 12, pp. 10629–10636, Dec. 2021, doi: 10.1007/s12652-020-02876-1.
- [8] M. S. Kabul and E. B. Setiawan, "Recommender System with User-Based and Item-Based Collaborative Filtering on Twitter using K-Nearest Neighbors Classification," *Journal of Computer System and Informatics (JoSYC)*, vol. 3, no. 4, pp. 478–484, Sep. 2022, doi: 10.47065/josyc.v3i4.2204.
- [9] P. Wlodarczak, "Agile Software Development," 2020.
- [10] M. C. Layton and D. Morrow, "Scrum 2nd Edition," 2021.