

Perancangan Aplikasi Android untuk Penjualan Sayur dan Rekomendasi Resep Berbasis AI

Rafi Achmad Fachrudi¹, Alindro Septo Nugroho², Prita Ayu Maudea³

^{1,2,3}Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Nusantara PGRI Kediri

E-mail: ¹achmadfahrud153@gmail.com, ²alindrosepto9@gmail.com, ³pritaayu1053@gmail.com

Abstrak – Bagi pelanggan yang tidak dapat mengunjungi pasar tradisional pada pagi hari, penjualan sayur segar secara online menjadi solusi. Tujuan penelitian adalah untuk membuat aplikasi Android yang memiliki fitur penjualan sayur dan memanfaatkan kecerdasan buatan atau Artificial intelligence (AI) yaitu algoritma K-Nearest Neighbors (KNN) dengan metode Cosine similarity, untuk memberikan rekomendasi resep berdasarkan bahan yang diinputkan pengguna. Data produk diperoleh melalui penelitian pasar, dan data resep diperoleh dari dataset Kaggle. Prototipe aplikasi memiliki antarmuka pengguna (UI) yang ramah pengguna dan fitur navigasi yang mudah digunakan. Untuk memastikan bahwa prototipe berjalan sesuai desain, pengujian fungsional dilakukan menggunakan metode black-box. Hasil penelitian ini menghasilkan prototipe aplikasi yang dapat meningkatkan pengalaman belanja sayur secara digital dan siap untuk digunakan di masa mendatang.

Kata Kunci — android, kecerdasan buatan(AI), penjualan sayur, rekomendasi resep, Prototipe aplikasi.

1. PENDAHULUAN

Kehidupan sehari-hari kita memerlukan sayur dan buah segar, tetapi banyak orang yang menghadapi masalah berbelanja di pasar tradisional karena keterbatasan waktu, terutama bagi mereka yang sibuk bekerja di pagi hari. Oleh karena itu, aplikasi berbasis Android yang memungkinkan penjualan buah dan sayur segar secara online adalah pilihan yang tepat.[1] Selain itu, aplikasi ini memiliki Artificial intelligence (AI) yang memungkinkan pengguna mendapatkan ide masakan yang kreatif dan memberikan rekomendasi resep berdasarkan bahan yang mereka miliki. Dengan fitur ini, pengguna tidak hanya dapat membeli bahan yang mereka butuhkan, tetapi juga dapat memilih resep yang sesuai dengan bahan yang mereka miliki di rumah. Aplikasi ini diharapkan dapat membantu orang-orang yang tidak memiliki waktu untuk pergi ke pasar tradisional dan memasak dengan bahan-bahan yang mereka miliki di rumah, karena teknologi AI memainkan peran penting dalam membantu pengguna menemukan resep yang tepat dengan cepat.

2. METODE PENELITIAN

Pada penelitian ini, metode pengembangan perangkat lunak *Waterfall* digunakan untuk mendesain sistem. [2] Ini dipilih karena sifatnya yang berurutan dan sistematis, mulai dari tahap analisis kebutuhan, perancangan, implementasi, dan pengujian. Proses pengembangan aplikasi yang berfokus pada penjualan sayur dan buah segar serta rekomendasi resep berbasis AI menjadi lebih mudah, karena setiap tahapannya dilaksanakan dengan jelas dan terstruktur. Berikut tahapan- tahapannya :

2.1 Perancangan Sistem

1. Analisa Kebutuhan Sistem

Pada tahap ini, analisis dilakukan terhadap kebutuhan fungsional dan nonfungsional aplikasi. Kebutuhan fungsional termasuk fitur utama seperti penjualan sayur dan buah, rekomendasi resep, dan sistem *checkout*. Kebutuhan nonfungsional termasuk desain antarmuka pengguna yang mudah digunakan, responsif, dan efisien.

2. Desain Antarmuka Pengguna

Untuk membuat *wireframe* dan *prototype* awal aplikasi, desain antarmuka dibuat menggunakan *Figma*. Tujuan desain adalah pengalaman pengguna yang mudah dipahami dan menarik, dan penyesuaian desain didasarkan pada riset pasar untuk memastikan bahwa aplikasi akan diterima dengan baik oleh pengguna.

3. Desain Sistem AI untuk Rekomendasi Resep

Pada tahap ini, algoritma KNN (*K-Nearest Neighbors*), yang digunakan dengan metode *Cosine similarity*, digunakan untuk menyarankan resep yang bergantung pada bahan yang dimiliki pengguna. Data resep yang digunakan bersumber dari Kaggle (<https://www.kaggle.com/datasets/albertnathaniel12/food-recipes-dataset>) dan disaring sesuai dengan kategori tertentu, seperti sapi, ayam, dan udang. Resep yang akan disarankan kepada pengguna juga dipertimbangkan saat memproses data.

4. Desain *Application Programming Interface* dan *Backend*

Aplikasi *backend* menggunakan *framework server-side Django* untuk mengelola data produk (sayur dan buah) serta resep. *Application Programming Interface (API)* yang dibangun di *backend* akan menghubungkan aplikasi Android ke *database*. Ini memungkinkan aplikasi untuk mengambil dan menampilkan data produk dan resep kepada pengguna.

2.2 Alat dan Teknologi yang digunakan

Beberapa alat dan teknologi yang digunakan untuk membuat aplikasi ini adalah sebagai berikut:

1. *Android Studio* adalah alat yang digunakan untuk membuat aplikasi Android.
2. *Django* adalah alat yang digunakan untuk *backend* dan pengelolaan data melalui API. [3]
3. KNN dan *Cosine similarity* adalah alat yang digunakan untuk mengembangkan algoritma AI yang menawarkan rekomendasi resep.[4][5]

a. *Cosine similarity*

Cosine similarity digunakan untuk mengukur kesamaan antara dua vektor. Nilai *cosine similarity* berkisar antara -1 (terbalik) hingga 1 (sama). Dalam konteks KNN, *cosine similarity* digunakan untuk mengukur seberapa mirip dua titik data (vektor fitur) satu sama lain. Berikut rumus *cosine similarity*:

$$\text{Cosine Similarity} = \frac{A \cdot B}{||A|| \cdot ||B||}$$

Dimana :

1. A dan B adalah vektor fitur (misalnya, vektor representasi produk atau data pengguna).
2. $A \cdot B$ adalah hasil perkalian dot antara vektor A dan B.
3. $||A|| \cdot ||B||$ adalah panjang (norma) dari vektor A dan B, yang dihitung dengan rumus:

$$||A|| = \sqrt{A_1^2 + A_2^2 + \dots + A_n^2}$$
$$||B|| = \sqrt{B_1^2 + B_2^2 + \dots + B_n^2}$$

b. KNN (*K-Nearest Neighbors*)

Untuk menggunakan *Cosine similarity* dalam algoritma KNN, langkah-langkahnya adalah:

1. Tentukan jumlah tetangga terdekat k. Berikut adalah perhitungannya :

$$\text{Euclidean Distance } (x, y) = (x_1 - y_1)^2 + (x_2 - y_2)^2 \dots + (x_n - y_n)^2$$

2. Hitung *cosine similarity* antara data titik yang diuji dengan semua titik data dalam dataset.
3. Pilih k tetangga dengan nilai *cosine similarity* tertinggi.
4. Tentukan kelas (atau label) berdasarkan mayoritas kelas dari tetangga terdekat tersebut.

Biasanya, penghitungan *cosine similarity* dilakukan pada setiap pasangan data untuk menentukan kedekatan antar data, dan berdasarkan itu, KNN akan mengklasifikasikan data baru ke dalam kategori yang sesuai.

4. TF-IDF: Digunakan dalam pemrosesan teks untuk mengetahui seberapa penting kata dalam data resep [6]

2.3 Proses Pengujian

Meskipun aplikasi belum digunakan sepenuhnya, beberapa langkah pengujian telah dilakukan untuk memastikan bahwa semua fitur bekerja dengan benar. Metode *black-box testing* digunakan untuk menguji fungsi yang diharapkan tanpa melihat struktur internal sistem. Di antara tes yang dilakukan adalah:

1. Pengujian antarmuka pengguna dan antarmuka: memastikan aplikasi mudah digunakan dan antarmuka memenuhi harapan fungsionalitas.
2. Pengujian API: Uji API aplikasi yang terhubung ke server *Django* untuk memastikan bahwa data produk dan resep diterima dan dikirim dengan benar.
3. Pengujian Algoritma AI: Memastikan bahwa algoritma rekomendasi resep berdasarkan bahan yang dimiliki pengguna menghasilkan rekomendasi yang relevan dan akurat.
4. Pengujian Alur *Checkout*: Untuk memastikan bahwa pelanggan dapat menyelesaikan proses pemilihan produk dengan sukses, uji alur *checkout* meskipun fitur pembayaran belum digunakan.

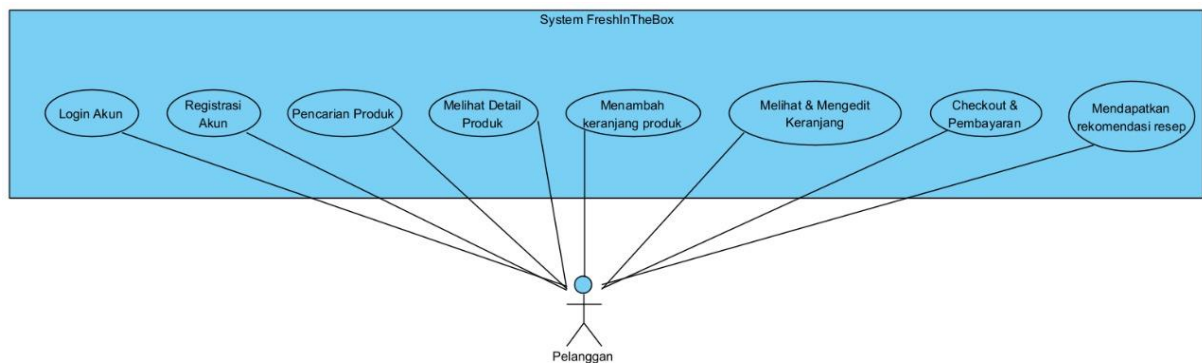
2.4 Evaluasi Sistem

Setelah tahap pengujian selesai, sistem akan dievaluasi berdasarkan hasilnya. Sebelum implementasi dan pengujian tambahan dilakukan, evaluasi ini akan menilai apakah aplikasi sudah memenuhi tujuan yang ditetapkan, seperti kemudahan penggunaan, relevansi rekomendasi resep, dan keandalan alur *checkout*. Hasil evaluasi akan digunakan untuk merencanakan perbaikan tambahan.

3. HASIL DAN PEMBAHASAN

3.1 Use Case Diagram

Aplikasi penjualan sayur dan rekomendasi resep berbasis kecerdasan buatan akan didukung oleh desain sistem ini. Proses perancangan melibatkan diagram dan tampilan antarmuka yang menunjukkan alur interaksi pengguna dengan fitur yang ada di aplikasi, sehingga pengguna dapat lebih mudah mengakses berbagai layanan. *Use Case Diagram* digunakan untuk menggambarkan fitur utama yang dapat diakses oleh pengguna, seperti login, registrasi, pencarian produk, hingga proses *checkout*. Berikut adalah gambaran diagramnya:



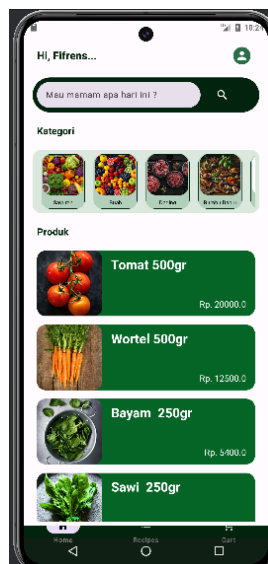
Gambar 1. Use Case Diagram

Diagram ini menunjukkan hubungan antara pengguna dengan fitur utama yang terdapat dalam aplikasi.

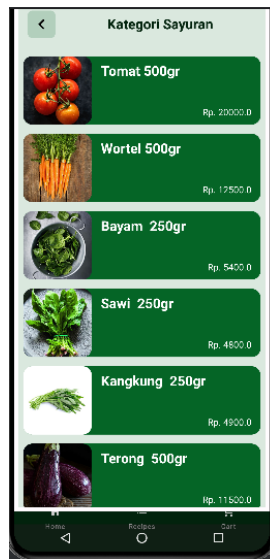
3.2 Desain Antarmuka

Aplikasi ini memiliki antarmuka yang dirancang dengan pendekatan yang sederhana dan mudah dipahami, sehingga pengguna dapat dengan mudah memahami cara kerja aplikasi. Berikut adalah beberapa tampilan antarmuka:

- Halaman Beranda: Menampilkan daftar produk dan kategori yang dilengkapi dengan pencarian produknya.



Gambar 2. Tampilan Beranda



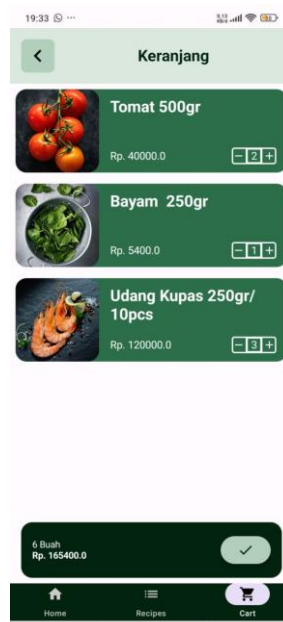
Gambar 3. Tampilan Produk Dalam Kategori Tertentu

- b. Halaman Produk : Menampilkan detail produk.



Gambar 4. Tampilan Detail Produk

- c. Halaman Keranjang : Mengelola produk yang dipilih untuk *checkout*.



Gambar 5. Tampilan Halaman Keranjang



Gambar 6. Tampilan *Checkout*

Pada halaman *checkout* maka ketika *icon* centang dipilih akan langsung terbaru halamannya menjadi keranjang kosong.

- d. Halaman Rekomendasi Resep: Memberikan rekomendasi resep berdasarkan bahan.



Gambar 7. Tampilan Rekomendasi Resep

Berikut adalah penjelasannya proses interaksi pengguna dalam aplikasi, mulai dari login hingga checkout : Pengguna dapat memulai menggunakan fitur *Login* Akun untuk masuk ke aplikasi. Pengguna dapat mencari produk sayur melalui fitur Pencarian Produk Sayur setelah berhasil masuk atau langsung memilih kategori yang tersedia. Pengguna dapat mengunjungi halaman Melihat Detail Produk jika mereka ingin informasi lebih lanjut. Kemudian, mereka dapat menambahkan produk ke keranjang dengan menggunakan fitur Menambahkan Produk ke Keranjang. Di halaman keranjang, mereka dapat melihat dan mengedit keranjang sebelum melanjutkan ke halaman *Checkout*.

API *Django* yang dibuat untuk mengintegrasikan sistem ke *backend* memungkinkan penampilan produk dan kategori di aplikasi Android dan menerima input pengguna untuk disimpan di *database*.

3.3 Pengujian Aplikasi

Fokus utama pengujian termasuk antarmuka pengguna (UI), alur kerja, integrasi API, dan akurasi rekomendasi resep berbasis AI untuk memastikan bahwa semua fitur aplikasi penjualan sayur dan rekomendasi resep dapat berjalan sesuai fungsinya.

3.4 Hasil Pengujian

Aplikasi ini menggunakan metode *Black Box Testing* untuk menguji. *Black Box Testing* memastikan bahwa setiap fitur berfungsi dengan baik tanpa memeriksa kode program.[7] Berikut adalah tabel hasil pengujian aplikasi:

Halaman/Fitur	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Catatan
Login	Memasukkan <i>username</i> dan <i>password</i> yang benar, klik <i>login</i> .	Pengguna berhasil masuk ke beranda.	Berhasil	
Registrasi	Masukkan data registrasi (nama, <i>email</i> , <i>password</i>), klik <i>submit</i> .	Akun baru berhasil dibuat, pengguna diarahkan ke halaman <i>login</i> .	Gagal	Tombol “ <i>submit</i> ” tidak berfungsi.
Pencarian Produk	Ketik kata kunci di <i>search bar</i> , tekan <i>icon</i> pencarian.	Produk yang sesuai kata kunci muncul di hasil pencarian.	Berhasil	<i>Search bar</i> bekerja dengan baik.
Tambah ke Keranjang	Pilih produk, klik tombol <i>icon</i> keranjang di samping total harga tertampil.	Produk berhasil ditambahkan ke keranjang.	Berhasil	

Checkout dan Pembayaran	Klik tombol " <i>Checkout</i> ", masukkan data pembayaran, konfirmasi.	Pesanan berhasil diproses, status pembayaran terbayar.	Gagal	Status pembayaran belum diperbarui setelah <i>checkout</i> .
Rekomendasi Resep	Masukkan bahan yang dimiliki ke input rekomendasi, klik <i>submit</i> .	Resep yang relevan muncul sesuai input bahan.	Berhasil	Algoritma KNN dan <i>Cosine similarity</i> bekerja sesuai harapan.
Halaman Keranjang	Klik "Lihat Keranjang", <i>edit</i> jumlah barang, hapus barang.	Perubahan jumlah dan penghapusan barang tercatat.	Berhasil	Tidak ada kendala.
Detail Rekomendasi Resep	Klik salah satu resep yang muncul dari rekomendasi	Menampilkan halaman detail resep yang didalamnya berisi bahan dan langkah – langkah pembuatan	Berhasil	

3.5 Analisis Hasil Pengujian

Berdasarkan hasil pengujian maka fitur yang berhasil yakni :

1. *Login*, pencarian produk, penambahan produk ke keranjang, dan rekomendasi resep telah berfungsi dengan baik.
2. Rekomendasi resep memberikan hasil yang relevan sesuai input bahan.

Namun juga ada fitur yang perlu diperbaiki yakni :

1. Registrasi: Tombol *submit* tidak berfungsi, sehingga pengguna tidak bisa membuat akun baru.
2. *Checkout*: Status pembayaran belum diperbarui meskipun proses *checkout* dapat dilakukan.

Performa sistemnya dianalisis dengan respons waktu untuk rekomendasi resep cukup cepat, dengan rata-rata waktu pengolahan di bawah 5 detik. UI aplikasi berjalan lancar, tetapi beberapa elemen perlu diperbaiki untuk pengalaman pengguna yang lebih baik, seperti tombol yang kurang responsif di halaman *checkout*.

3.6 Kendala dan Tantangan

Selama pengembangan dan pengujian aplikasi, beberapa kendala dan tantangan yang dihadapi adalah:

- a. Integrasi API nya karena API belum dilakukan *hosting* sehingga pengujian hanya dapat dilakukan di lingkungan lokal, membatasi fleksibilitas pengujian antar perangkat.
- b. Registrasi akun mengalami kegagalan pada tombol *submit*, yang kemungkinan disebabkan oleh validasi form di *backend*.
- c. *Checkout* dan pembayaran terletak pada status pembayaran yang tidak otomatis diperbarui setelah *checkout* perlu dilakukan *debugging* pada alur transaksi.
- d. Manajemen waktu karena pengembangan sistem dilakukan dalam waktu terbatas, sehingga beberapa fitur belum sempurna, seperti pengaturan status pembayaran dan *hosting* API.

4. SIMPULAN

Tujuan dari penelitian ini adalah untuk membuat dan mengembangkan aplikasi penjualan sayur berbasis Android yang memiliki fitur rekomendasi resep yang menggunakan algoritma KNN dan *Cosine similarity*. Aplikasi ini dimaksudkan untuk membantu pengguna mencari bahan masakan dan memberikan rekomendasi resep yang sesuai berdasarkan bahan yang mereka miliki. Pengujian yang telah dilakukan menunjukkan bahwa aplikasi dapat melaksanakan beberapa fitur utama dengan sukses, seperti pencarian produk, keranjang belanja, dan rekomendasi resep berbasis AI. Algoritma KNN dan *Cosine similarity* juga terbukti dapat memberikan rekomendasi yang relevan dengan bahan input pengguna. Selain itu, antarmuka pengguna (UI) halaman beranda, keranjang, dan resep berjalan dengan baik, tetapi fitur tertentu, seperti status pembayaran dan registrasi akun selama proses *checkout*, perlu diperbaiki. Meskipun aplikasi ini mempermudah proses belanja bahan makanan dan menawarkan pengalaman baru dengan fitur rekomendasi resep berbasis AI, ada beberapa keterbatasannya, seperti API yang belum diintegrasikan secara lokal dan status pembayaran yang belum otomatis diperbarui.

5. SARAN

Pengembangan aplikasi ini masih memiliki ruang untuk perbaikan dan pengembangan lebih lanjut. Disarankan agar API yang digunakan dapat diintegrasikan ke dalam *server hosting* sehingga aplikasi dapat digunakan secara *online* dengan lebih fleksibel. Selain itu, fitur *checkout* perlu ditingkatkan agar status pembayaran dapat diperbarui secara otomatis setelah transaksi berhasil. Kendala pada fitur registrasi akun juga perlu diselesaikan agar pengguna baru dapat melakukan pendaftaran tanpa hambatan. Selanjutnya, variasi rekomendasi resep dapat diperluas dengan mempertimbangkan preferensi pengguna atau bahan alternatif untuk meningkatkan relevansi hasil rekomendasi. Terakhir, disarankan untuk melakukan uji coba skala besar dengan lebih banyak pengguna untuk mengevaluasi performa aplikasi dan mendapatkan masukan yang lebih luas guna pengembangan di masa mendatang.

DAFTAR PUSTAKA

- [1] A. Saifudin, A. Wahid, I. U. Aranda, and S. Amruallah, “Pengembangan Aplikasi Mobile pada Studi Kasus Penjualan Sayur dan Buah Berbasis Android,” vol. 3, no. 6, pp. 1576–1582, 2024.
- [2] A. Hidayat, N. Q. Ridhaihi, M. F. A. Shiddiq, F. T. Ra’pak, and A. A. Khaerunnisa, “Pengembangan Aplikasi MySaku Menggunakan Metode Waterfall,” *Indones. Technol. Educ. J.*, vol. 01, no. 02, pp. 68–77, 2023.
- [3] A. Firdiansyah, I. Al Ikrom, M. Khamdanni, and W. C. Utomo, “Pemanfaatan Data Mining Untuk Memprediksi Kelulusan Mata Kuliah dan Referensi Strategi Pembelajaran,” *Semin. Nas. Teknol. Sains*, vol. 3, no. 1, pp. 338–344, 2024, doi: 10.29407/stains.v3i1.4338.
- [4] Luqyana Zakiya Almas, Yuliana Susanti, and Sri Sulistijowati Handajani, “Penerapan Algoritma K-Nearest Neighbors dalam Sistem Rekomendasi Makanan Berdasarkan Kebutuhan Nutrisi dengan Content-Based Filtering,” *Statistika*, vol. 24, no. 1, pp. 115–122, 2024, doi: 10.29313/statistika.v24i1.3558.
- [5] P. C. Siswipraptini, “Klasifikasi Pekerjaan Bidang Teknologi Informasi Menggunakan Algoritma *Cosine similarity*,” *Kilat*, vol. 12, no. 1, pp. 38–48, 2023, doi: 10.33322/kilat.v12i1.2001.
- [6] A. A. Huda, R. Fajarudin, and A. Hadinegoro, “Sistem Rekomendasi Content-based Filtering Menggunakan TF-IDF Vector Similarity Untuk Rekomendasi Artikel Berita,” *Build. Informatics, Technol. Sci.*, vol. 4, no. 3, pp. 1679–1686, 2022, doi: 10.47065/bits.v4i3.2511.
- [7] N. Ayuningsih, A. D. Rachmanto, and D. Hernawati, “Perancangan Aplikasi Belajar Mengaji Ilmu Tajwid Berbasis Android,” *J. Teknol. Inf. dan Komun.*, vol. 11, no. 1, 2021, doi: 10.56244/fiki.v11i1.419.