

**IMPLEMENTASI MACHINE LEARNING DALAM SISTEM
REKOMENDASI PEMESANAN SAYUR MENGGUNAKAN
METODE *CONTENT-BASED FILTERING***



Oleh :

RAHMAT HIDAYAT | 192581
MUHAMMAD FAHREZY AL DZUHRI | 192336

**UNIVERSITAS DIPA MAKASSAR
2024**

IMPLEMENTASI *MACHINE LEARNING* DALAM SISTEM
REKOMENDASI PEMESANAN SAYUR MENGGUNAKAN
METODE *CONTENT-BASED FILTERING*

Oleh:

RAHMAT HIDAYAT (192581)
MUHAMMAD FAHREZY AL DZUHRI (192336)

Skripsi Diserahkan ke Universitas Dipa Makassar untuk Memenuhi
Persyaratan Sarjana Program Studi Teknik Informatika
September 2024

**IMPLEMENTASI *MACHINE LEARNING* DALAM SISTEM
REKOMENDASI PEMESANAN SAYUR MENGGUNAKAN
METODE *CONTENT-BASED FILTERING***

SKRIPSI

Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Strata 1
Program Studi Teknik Informatika

RAHMAT HIDAYAT (192581)
MUHAMMAD FAHREZY AL DZUHRI (192336)

Telah Disetujui Untuk Dipertahankan

Pembimbing I

Erni Marlina, S.Kom., M.I.Kom
NIDN : 0914037501

Pembimbing II

Sunardi, S.Kom., M.T
NIDN : 0931108803

PENGESAHAN NASKAH SKRIPSI

IMPLEMENTASI *MACHINE LEARNING* DALAM SISTEM REKOMENDASI PEMESANAN SAYUR MENGGUNAKAN METODE *CONTENT-BASED FILTERING*

RAHMAT HIDAYAT (192581)
MUHAMMAD FAHREZY AL DZUHRI (192336)

Tanggal Ujian - September 2024

Pembimbing I, Pembimbing II,

Erni Marlina, S.Kom., M.I.Kom Sunardi, S.Kom., M.T
NIDN : 0914037501 NIDN : 0931108803

Penguji I, Penguji II,

Dr. Cucut Susanto, S.Kom., M.Si Sri Wahyuni, S.Kom., M.T
NIDN : 0914037501 NIDN : 0931108803

Mengetahui,
Ketua Program Studi Teknik Informatika
Universitas Dipa Makassar

Ir. Irsal, MT.
NIDK : 9990216745

PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi yang diserahkan untuk memperoleh gelar Sarjana Komputer adalah hasil karya saya sendiri dan semua sumbangan dari orang atau sumber lain dikutip dengan baik dan benar. Saya selanjutnya menyatakan bahwa materi dalam skripsi ini belum pernah diserahkan baik seluruhnya atau sebagian, untuk mendapatkan gelar di universitas ini atau universitas lainnya. Dalam membuat pernyataan ini, saya memahami dan mengakui setiap pelanggaran dalam pernyataan ini merupakan pelanggaran akademis, yang dapat mengakibatkan saya dikeluarkan dari Universitas dan/atau pencabutan dari gelar yang telah diberikan.

Demikian pernyataan ini dibuat dengan sangat sadar dan tanpa paksaan dari pihak manapun.

Makassar, Septemberi 2024

Penulis I,

Penulis II,

Rahmat Hidayat
Stambuk: 192581

Muhammad Fahrezy Al Dzuhrí
Stambuk: 192336

KATA PENGANTAR

Segala puji dan syukur kepada Tuhan Yang Maha Esa yang telah melimpahkan rahmat dan karunia-Nya kepada penulis, sehingga penulis bisa menyelesaikan skripsi ini dengan judul “IMPLEMENTASI *MACHINE LEARNING DALAM SISTEM REKOMENDASI PEMESANAN SAYUR MENGGUNAKAN METODE CONTENT-BASED FILTERING*”. Shalawat dan juga salam senantiasa tercurah kepada Junjungan kita Nabi Muhammad *Shalallahu Alaihi Wassalam* yang mengantarkan manusia dari kegelapan ke zaman yang terang benderang.

Dalam penyelesaian skripsi ini, penulis menyadari bahwa penulisan ini tidak dapat terselesaikan tanpa pihak yang mendukung baik dukungan secara moral maupun dukungan secara material. Maka penulis mengucapkan banyak terima kasih kepada pihak yang membantu dalam penulisan skripsi ini, terutama kepada:

1. Dr. Y. Johny W. Soetikno, S.E., M.M., selaku Rektor Universitas Dipa Makassar.
2. Ir. Irsal, MT. selaku Ketua Jurusan Teknik Informasi yang senantiasa memberikan arahan dan petunjuk pada penyelesaian skripsi ini.
3. Erni Marlina, S.Kom., M.I.Kom. selaku pembimbing I yang selalu sabar memberikan arahan dan petunjuk pada penyelesaian skripsi ini.
4. Sunardi, S.Kom., MT. selaku pembimbing II yang selalu sabar memberikan bimbingan dan arahan penulis dalam penyelesaian skripsi ini

5. Kedua orang tua penulis yang senantiasa selalu mendukung dalam bentuk apapun dan tak bosan-bosannya juga memberikan nasihat kepada penulis,
6. Semoga Tuhan selalu senantiasa melimpahkan kesehatan dan kesejahteraan bagi beliau, Amin.
7. Kepada seluruh rekan yang telah membantu sampai penulis mampu menyelesaikan skripsi ini tepat waktu yang tak sempat penulis sebutkan namanya satu per satu.

Penulis menyadari bahwa penyusunan skripsi ini masih banyak kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran konstruktif demi perkembangan positif bagi penulis. Seperti yang dikatakan oleh salah satu pemanah asal Rusia “*If you’re not a good shot today, don’t worry. There are other ways to be useful*”.

Demikian skripsi ini penulis susun, semoga dapat bermanfaat bagi semua pihak dan penulis sendiri, akhir kata penulis ucapkan terima kasih.

Makassar, September 2024

Penulis

ABSTRAK

Penelitian ini mengembangkan aplikasi *mobile* berbasis *Android* untuk memfasilitasi pemesanan sayur secara online dengan antarmuka yang *user-friendly*. Aplikasi ini menggunakan teknologi *machine learning* untuk mengenali pola gambar sayuran yang diinput admin, lalu mengelolanya sebagai *hashtag*. Metode *Content-Based Filtering* diterapkan untuk memberikan rekomendasi sayuran yang relevan berdasarkan riwayat pemesanan pengguna yang memiliki data *hashtag*. Data yang digunakan meliputi gambar sayuran yang diinput admin dan riwayat pemesanan pengguna. Algoritma *Convolutional Neural Network (CNN)* menunjukkan tingkat akurasi yang memadai dalam mengenali dan mengklasifikasikan jenis sayuran.

Hasil penelitian menunjukkan bahwa aplikasi ini tidak hanya memenuhi kebutuhan dasar pemesanan sayur *online*, tetapi juga menawarkan nilai tambah melalui pengenalan gambar dan sistem rekomendasi terintegrasi. Pengembangan lebih lanjut disarankan, seperti fitur kustomisasi rekomendasi dan ekspansi ke *platform iOS*.

Kata kunci: Aplikasi *Mobile*, *Machine Learning*, *Content-Based Filtering*, Pemesanan Sayur, Rekomendasi Produk

ABSTRACT

This research developed an Android-based mobile application to facilitate online vegetable ordering with a user-friendly interface. The application utilizes machine learning technology to recognize patterns in vegetable images input by admins, which are then managed as hashtags. The Content-Based Filtering method is applied to provide relevant vegetable recommendations based on the user's ordering history that includes hashtag data. The data used includes vegetable images input by admins and the order history of users. The Convolutional Neural Network (CNN) algorithm used demonstrated adequate accuracy in recognizing and classifying vegetable types.

The research findings indicate that this application not only meets the basic needs of online vegetable ordering but also offers added value through integrated image recognition and recommendation systems. Further development is suggested, such as customization of recommendation features and expansion to the iOS platform.

Keywords: *Mobile Application*, *Machine Learning*, *Content-Based Filtering*, *Vegetable Ordering*, *Product Recommendation*

DAFTAR ISI

Halaman

HALAMAN JUDUL.....	1
PERSETUJUAN NASKAH SKRIPSI.....	3
PENGESAHAN NASKAH SKRIPSI.....	4
PERNYATAAN.....	5
KATA PENGANTAR.....	6
ABSTRAK.....	8
DAFTAR ISI.....	9
DAFTAR TABEL.....	11
DAFTAR GAMBAR.....	13
BAB I PENDAHULUAN.....	15
1.1 Latar Belakang.....	15
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	3
1.5 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1. Kerangka Pikir.....	5
2.2. Kerangka Teori.....	6
2.2.1. Sistem Rekomendasi.....	7
2.2.2. <i>Content-Based Filtering</i>	8

2.2.5. <i>Machine Learning</i>	11
2.2.6. Pemesanan.....	14
2.2.7. <i>Android</i>	14
2.2.8. <i>Flutter</i>	15
2.2.9. <i>Supabase</i>	16
2.2.10. <i>Use Cases</i>	16
2.2.11. <i>Activity Diagram</i>	17
2.2.12. <i>Class Diagram</i>	18
2.2.13. <i>Visual Studio Code</i>	19
2.2.14. <i>Black box</i>	20
2.3. Penelitian Terkait.....	20
BAB III METODE PENELITIAN.....	23
3.1. Waktu dan Tempat Penelitian.....	23
3.2. Alat dan Bahan.....	23
3.2.1. Alat Penelitian.....	23
3.2.2. Bahan Penelitian.....	24
3.3. Jenis Penelitian.....	24
3.4. Metode Pengumpulan Data.....	24
3.5. Tahapan Penelitian.....	25
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	26
4.1. Perancangan Solusi.....	26

4.1.1. <i>Use Case Diagram</i>	27
4.1.2. Arsitektur Sistem.....	28
4.1.3. <i>Class Diagram</i>	28
4.1.4. <i>Activity Diagram</i>	29
4.1.5. Rancangan Aplikasi Admin.....	41
4.1.6. Rancangan Aplikasi <i>Mobile</i>	43
4.1.7. Implementasi <i>Machine Learning</i>	48
4.1.8. Implementasi Algoritma <i>Content-Based Filtering</i> pada <i>Back-End</i> . 48	48
4.1.9. Implementasi Algoritma <i>Content-Based Filtering</i> pada <i>Android</i>	48
4.2. Analisa dan Validasi Data.....	48
4.3. Hasil dan Pengujian.....	49
4.3.1. Proses <i>Machine Learning</i>	49
4.3.2. Proses Metode <i>Content-Based Filtering</i>	54
4.3.3. Aplikasi Admin.....	60
4.3.4. Aplikasi <i>Mobile</i>	66
BAB V KESIMPULAN.....	78
5.1. Kesimpulan.....	78
5.2. Saran.....	79

DAFTAR PUSTAKA

LAMPIRAN

RIWAYAT HIDUP

DAFTAR TABEL

	Halaman
Tabel 2. 1 <i>Use Case</i>	16
Tabel 2. 2 Simbol <i>Activity Diagram</i>	17
Tabel 2. 3 Simbol <i>Class Diagram</i>	18
Tabel 2. 4 Penelitian Terkait	20
Tabel 3. 1 Tahapan Penelitian	24
Tabel 4. 1 Algoritma <i>Convolutional Neural Network</i>	49
Tabel 4. 2 Tabel Hasil Uji Coba Deteksi Objek	53
Tabel 4. 3 Hasil Proses <i>Clean Data</i>	54
Tabel 4. 4 Hasil Proses <i>Tokenisasi</i>	54
Tabel 4. 5 Hasil Proses <i>Filtering</i>	55
Tabel 4. 6 Perhitungan <i>Term Frequency</i>	56
Tabel 4. 7 Perhitungan <i>Inverse Document Frequency</i>	57
Tabel 4. 8 Perhitungan <i>TF-IDF</i>	58
Tabel 4. 9 Pengujian <i>Login Admin</i>	59
Tabel 4. 10 Pengujian Tambah <i>Product Admin</i>	60
Tabel 4. 11 Pengujian Edit <i>Product Admin</i>	61
Tabel 4. 12 Pengujian Hapus <i>Product Admin</i>	61
Tabel 4. 13 Pengujian Identifikasi <i>Image</i> dengan <i>Machine Learning</i> Admin ...	62
Tabel 4. 14 Pengujian <i>Transaction Admin</i>	63
Tabel 4. 15 Pengujian <i>Customer Admin</i>	63
Tabel 4. 16 Pengujian <i>Log-Out Admin</i>	64

Tabel 4. 17 Pengujian <i>Black Box</i> Aplikasi Admin	65
Tabel 4. 18 Pengujian <i>Register Customer</i>	66
Tabel 4. 19 Pengujian <i>Login Customer</i>	67
Tabel 4. 20 Pengujian Rekomendasi <i>Customer</i>	68
Tabel 4. 21 Pengujian <i>Search</i> Rekomendasi <i>Customer</i>	69
Tabel 4. 22 Pengujian <i>Filter</i> Kategori <i>Customer</i>	70
Tabel 4. 23 Pengujian <i>Search</i> Kategori <i>Customer</i>	70
Tabel 4. 24 Pengujian Riwayat Pesanan <i>Customer</i>	71
Tabel 4. 25 Pengujian Keranjang <i>Customer</i>	72
Tabel 4. 26 Pengujian <i>Checkout Customer</i>	73
Tabel 4. 27 Pengujian <i>Update Profile Customer</i>	74
Tabel 4. 28 Pengujian <i>Log-Out Customer</i>	74
Tabel 4. 29 Pengujian <i>Black Box</i> Aplikasi <i>Android</i>	75

DAFTAR GAMBAR

Halaman

Gambar 2. 1 Kerangka Pikir	5
Gambar 2. 2 <i>Architecture Content-Based Filtering</i>	7
Gambar 2. 3 <i>Architecture Umum Tensorflow</i>	13
Gambar 2. 4 Kerangka Penelitian Terkait	21
Gambar 4. 1 <i>Use case Diagram</i>	26
Gambar 4. 2 <i>Arsitektur Sistem</i>	27
Gambar 4. 3 <i>Class Diagram</i>	28
Gambar 4. 4 <i>Activity Diagram Login Admin</i>	29
Gambar 4. 5 <i>Activity Diagram Tambah data sayur</i>	30
Gambar 4. 6 <i>Activity Diagram Edit Data Sayur</i>	31
Gambar 4. 7 <i>Activity Diagram Detail Data Sayur</i>	32
Gambar 4. 8 <i>Activity Diagram Hapus Data Sayur</i>	33
Gambar 4. 9 <i>Activity Diagram Kelola Data Pemesanan</i>	34
Gambar 4. 10 <i>Activity Diagram Kelola Data Customer</i>	35
Gambar 4. 11 <i>Activity Diagram Login Customer</i>	36
Gambar 4. 12 <i>Activity Diagram Edit Profil Customer</i>	37
Gambar 4. 13 <i>Activity Diagram Keranjang Customer</i>	38
Gambar 4. 14 <i>Activity Diagram Pemesanan Sayur</i>	39
Gambar 4. 15 Rancangan Tampilan <i>Login Admin</i>	40
Gambar 4. 16 Rancangan Tampilan <i>Dashboard Admin</i>	41
Gambar 4. 17 Rancangan Tampilan Tabel <i>Product Admin</i>	41

Gambar 4. 18 Rancangan Tampilan Tabel <i>Transaction Admin</i>	42
Gambar 4. 19 Rancangan Tampilan Tabel <i>Customer Admin</i>	42
Gambar 4. 20 Rancangan Tampilan <i>Register Customer</i>	43
Gambar 4. 21 Rancangan Tampilan <i>Login Customer</i>	43
Gambar 4. 22 Rancangan Tampilan <i>Home Customer</i>	44
Gambar 4. 23 Rancangan Tampilan Kategori <i>Customer</i>	45
Gambar 4. 24 Rancangan Tampilan Riwayat Pesanan <i>Customer</i>	45
Gambar 4. 25 Rancangan Tampilan <i>Profil Customer</i>	46
Gambar 4. 26 Rancangan Tampilan Keranjang <i>Customer</i>	46
Gambar 4. 27 Gambaran Operasi <i>Konvolusi</i>	50
Gambar 4. 28 <i>Max-Pooling</i>	51
Gambar 4. 29 <i>Fully Connected Layer</i>	52

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pertumbuhan pesat teknologi *mobile* dan aplikasi berbasis *android* telah mengubah cara konsumen berbelanja produk sehari-hari, termasuk sayur-sayuran. Sayur merupakan kelompok bahan makanan yang berasal dari bahan nabati. yaitu dari berbagai bagian tumbuhan, seperti daun, akar, batang dan bunga.

Namun, dalam lingkungan yang semakin sibuk, mencari dan memesan sayur yang sesuai dengan preferensi dan kebutuhan individu bisa menjadi tugas yang rumit dan memakan waktu. Oleh karena itu, pengembangan aplikasi *mobile android* untuk pemesanan sayur menjadi relevan untuk memberikan solusi yang nyaman bagi konsumen.

Content-Based Filtering adalah salah satu metode dari sistem rekomendasi yang didasarkan pada atribut, karakteristik, atau fitur yang dimiliki oleh sebuah item (P. Nastiti., 2019). Sistem rekomendasi dengan metode *content-based filtering* merekomendasikan item yang mirip dengan item sebelumnya yang disukai atau dipilih oleh pengguna. Kemiripan item dihitung berdasarkan pada fitur-fitur yang ada pada item yang dibandingkan. Metode ini bersifat *pengguna independence*, tidak bergantung pada situasi apakah item tersebut merupakan item baru (yang belum pernah dipilih oleh pengguna manapun) maupun bukan item baru. Jika seorang pengguna telah memesan suatu menu hidangan pada kategori tertentu maka sistem akan mencoba merekomendasikan sayur dengan kategori

serupa yang juga tersedia yang mungkin akan disukai juga oleh pengguna tersebut.

Penelitian akan dilakukan dengan menerapkan *Machine Learning* untuk mengenali pola pada gambar serta menerapkan metode *Content-Based Filtering* dalam pengembangan sistem rekomendasi pada aplikasi pemesanan sayur. Rekomendasi yang dihasilkan akan mencakup beberapa jenis sayur yang memiliki kemiripan konten dengan sayur acuan. Konten tersebut diperoleh melalui proses pengenalan pola oleh algoritma Machine Learning.

Dengan pembuatan aplikasi *mobile android*, penelitian ini bertujuan untuk mengimplementasikan metode *Content-Based Filtering* dalam sistem rekomendasi pemesanan sayur. Dimana dalam penelitian ini diharapkan aplikasi akan memberikan kemudahan kepada pengguna dalam proses pemesanan sayur khususnya di pemukiman Bukit Baruga.

1.2 Rumusan Masalah

Berdasarkan uraian diatas, maka rumusan masalah untuk proyek akhir ini adalah sebagai berikut :

1. Bagaimana cara merancang dan mengembangkan aplikasi *mobile android*?
2. Bagaimana cara mengenali pola pada suatu gambar?
3. Bagaimana cara merekomendasikan sayur berdasarkan riwayat pemesanan?

1.3 Tujuan Penelitian

Tujuan dari pembuatan alat ini antara lain :

1. Menyediakan Aplikasi yang memudahkan bagi pengguna perangkat *mobile android*.

2. Mengimplementasikan *Machine Learning* dalam mengenali pola pada suatu gambar.
3. Mengimplementasikan cara merekomendasikan sayur secara efisien dalam aplikasi dengan menerapkan metode *Content-Based Filtering*.

1.4 Manfaat Penelitian

Adapun manfaat pada penelitian ini adalah sebagai berikut :

1. Bagi penulis, sebagai wadah dalam mengimplementasikan ilmu yang didapatkan selama masa kuliah dan sebagai syarat untuk mendapatkan gelar sarjana pada jurusan Teknik Informatika Universitas Dipa Makassar.
2. Bagi Universitas Dipa Makassar, Penelitian yang menghasilkan temuan atau inovasi yang dapat dikomersialisasikan dapat mendorong pengembangan kewirausahaan di kampus. Ini dapat menciptakan peluang baru untuk penciptaan lapangan kerja dan meningkatkan kontribusi ekonomi kampus.
3. Bagi masyarakat Baruga, meningkatkan aksesibilitas masyarakat Baruga terhadap sayur-sayuran. Dengan adanya aplikasi, masyarakat dapat dengan mudah menjelajahi dan memesan sayur-sayuran secara online, memudahkan mereka untuk mendapatkan bahan makanan yang segar dan berkualitas.

1.5 Sistematika Penulisan

BAB I : PENDAHULUAN

Bab ini membahas latar belakang, rumusan masalah, tujuan dan manfaat penelitian serta sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Bab ini membahas kerangka pikir, landasan teori dan penelitian terkait.

BAB III : METODE PENELITIAN

Bab ini membahas gambaran umum mengenai penelitian yang dilakukan antara lain lokasi dan waktu penelitian, jenis penelitian, sumber data, teknik pengumpulan data, alat dan bahan, metode pengujian, urutan kegiatan, jadwal kegiatan.

BAB IV : HASIL PENELITIAN DAN PEMBAHASAN

Membahas tentang sistem yang digunakan dan proses berjalannya suatu aplikasi yang dibuat.

BAB V : PENUTUP

Membahas tentang tentang kesimpulan dan saran yang di buat oleh penulis.

DAFTAR PUSTAKA

LAMPIRAN

RIWAYAT HIDUP

BAB II

TINJAUAN PUSTAKA

2.1. Kerangka Pikir

Secara garis besar alur gambaran penelitian yang penulis akan lakukan dapat dilihat pada kerangka pikir berikut :

Dalam penelitian ini didorong oleh transformasi perilaku konsumen dalam berbelanja, terutama sayuran, yang beralih ke *platform mobile*. Kehadiran aplikasi *android* untuk memesan sayuran menjadi penting karena memberikan solusi praktis dalam memenuhi kebutuhan konsumen yang sibuk.



Sistem rekomendasi menggunakan metode *Content-Based Filtering* menjadi fokus penelitian untuk memberikan saran sayuran yang sesuai dengan preferensi dan kebutuhan pengguna.



Metode ini didasarkan pada atribut dan karakteristik sayuran serta pola pembelian sebelumnya, dengan tujuan meningkatkan kemudahan dalam proses pemesanan. Melalui penerapan *Machine Learning* untuk mengenali pola pada gambar dan algoritma *Content-Based Filtering*.



Penelitian ini bertujuan untuk membuat aplikasi yang dapat meningkatkan aksesibilitas pengguna dan memberikan kemudahan dalam melakukan pemesanan sayuran, khususnya di wilayah pemukiman Bukit Baruga.

Gambar 2.1 Kerangka Pikir

2.2. Kerangka Teori

Dalam pembuatan suatu sistem tidak bisa terlepas dari beberapa teori yang menjadi dasar dalam pembuatan sistem tersebut, hal ini diperlukan agar proses pembangunan sistem dapat dibuat dengan mudah dan sesuai dengan yang diinginkan. Adapun teori yang dijadikan landasan dalam membangun sistem ini adalah sebagai berikut:

2.2.1. Sistem Rekomendasi

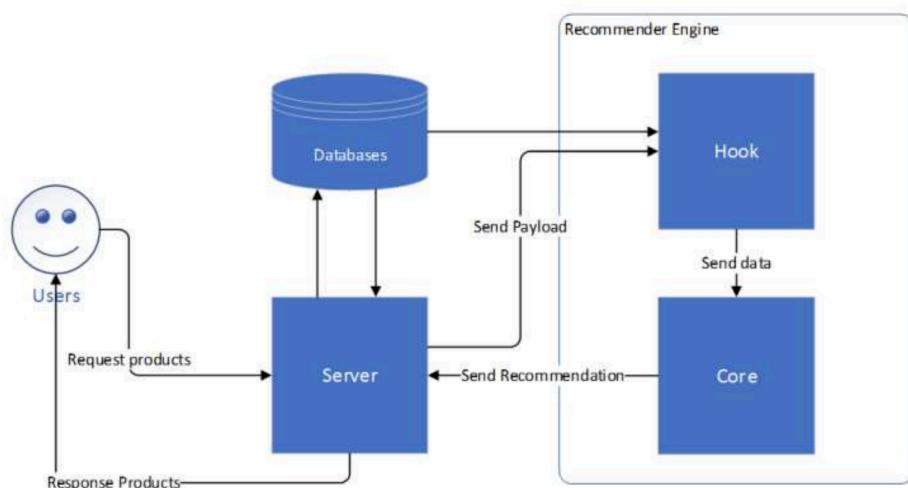
Sistem Rekomendasi, yang juga disebut mesin rekomendasi, merupakan aplikasi yang membantu pengguna dalam membuat keputusan dengan memberikan rekomendasi terhadap item berdasarkan analisis data yang tersedia. Faktor-faktor seperti perilaku pengguna, deskripsi item, dan preferensi kelompok pengguna yang serupa dapat mempengaruhi keputusan sistem dalam menyajikan rekomendasi. Tujuan utama sistem rekomendasi adalah menyaring informasi dengan menekankan pada penyaringan produk atau item yang mungkin menarik bagi pengguna.

Banyak perusahaan, seperti *Amazon*, *Netflix*, dan *YouTube*, telah mengadopsi sistem rekomendasi ini untuk membantu pengguna menemukan item dari beragam koleksi. Sistem ini menawarkan serangkaian item yang sesuai dengan kebutuhan dan preferensi pengguna dari berbagai deskripsi item yang tersedia di *website* atau sumber informasi lainnya.

Sistem rekomendasi berkembang terus dan umumnya dapat dikelompokkan menjadi tiga metode utama: *content-based*, *collaborative filtering*, dan *hybrid recommender system*. *Content-based filtering* memberikan

rekomendasi dengan mempertimbangkan perbandingan antara item dan informasi tambahan seperti teks, gambar, atau video. *Collaborative filtering* menggunakan interaksi historis pengguna terhadap item, baik dalam bentuk peringkat atau riwayat penelusuran. Sedangkan model *hybrid* merupakan kombinasi dari kedua pendekatan tersebut.

Dalam penelitian kali ini akan menggunakan metode *content-based filtering* dalam memberikan rekomendasi sayur kepada pengguna. Yang dimana sistem rekomendasi dengan metode *content-based filtering* merekomendasikan item yang mirip dengan item sebelumnya yang disukai atau dipilih oleh pengguna. Adapun *architecture system* dari *content-based filtering*, sebagai berikut.



Gambar 2.2 *Architecture Content-Based Filtering*

2.2.2. *Content-Based Filtering*

Content Algoritma berbasis konten merekomendasikan item yang kontennya mirip dengan konten dari item yang pernah dinilai positif oleh pengguna di masa lalu. Misalnya, dalam domain film, konten seperti judul film, para aktor yang berperan, sutradara, *genre*, dan ringkasan cerita dapat menjadi

faktor konten tersebut. Meskipun pendekatan dasar untuk rekomendasi berbasis konten didasarkan pada analisis kejadian kata-kata per item, dan mengabaikan struktur semantik dari konten item, teknik yang lebih canggih mencoba untuk memanfaatkan fitur semantik tersebut.

Content-Based Filtering pada Sistem rekomendasi adalah metode yang mempertimbangkan perilaku dari pengguna dari masa lalu yang kemudian diidentifikasi pola perilakunya untuk merekomendasikan barang yang sesuai dengan pola perilaku tersebut (Reddy et al., 2019). Metode *content-based filtering* menganalisis preferensi dari perilaku pengguna dimasa lalu untuk membuat model. Model tersebut akan dicocokkan dengan serangkaian karakteristik atribut dari barang yang akan direkomendasikan. Barang dengan tingkat kecocokan tertinggi akan menjadi rekomendasi untuk pengguna. (Muhammad Fajriansyah, Putra Pandu Adikara, & Agus Wahyu Widodo. 2021).

Kelemahan dari metode content-based filtering adalah terbatasnya rekomendasi hanya pada item-item yang mirip sehingga tidak ada kesempatan untuk mendapatkan item yang tidak terduga. Akan tetapi, Pazzani dan Bilsus dalam jurnalnya menjelaskan memang benar bahwa tidak menutup kemungkinan pengguna yang melihat atau membeli boneka juga telah membeli film dewasa, oleh sebab itu hal-hal seperti ini sebaiknya tidak direkomendasikan.

Salah satu metode yang digunakan pada *Content Based Filtering* adalah *TF-IDF (Term Frequency Inverse Document Frequency)* dan *Cosine Similarity* (Alkaff, 2020)

2.2.3. Term Frequency Inverse Document Frequency

Term Frequency Inverse Document Frequency (TF-IDF) adalah nilai statistik yang mencerminkan atau menunjukkan kemiripan suatu term dengan beberapa data dalam dokumen yang membuat term tersebut bisa menjadi kata kunci dari dokumen tersebut. Nilai *TF-IDF* diambil dari perhitungan *Term Frequency (TF)* yang menggambarkan frekuensi kemunculan istilah atau kata yang muncul dari dokumen yang ada, dan *Inverse Document Frequency (IDF)* yang menggambarkan frekuensi dokumen yang memiliki istilah dalam koleksi (Arfisko & Wibowo, 2022). Berikut adalah rumus dari *Term Frequency*.

$$tf(d, t) = 1 + \log(f_{dt})$$

Keterangan :

$tf(d,t)$ = Nilai TF untuk kata ke-t dalam dokumen ke-d

f_{dt} = Jumlah kata ke-t dalam dokumen ke-d

Berikut adalah rumus perhitungan dari *Inverse Document Frequency* :

$$IDF(t) = \log \frac{D}{df}$$

Keterangan :

$IDF(t)$ = Nilai IDF untuk kata ke-t pada item

Df = Jumlah dokumen yang mengandung kata ke-t

D = Total jumlah dokumen pada item

Berikut adalah rumus yang digunakan dalam proses TF-IDF :

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

Keterangan :

d = dokumen ke-d

t = kata ke-t dari kata kunci

W = bobot dokumen d terhadap kata ke-t

2.2.4. Cosine Similarity

Cosine similarity adalah metode pengukuran kesamaan yang biasa digunakan pada data atau dokumen berupa teks yang mengukur dua vektor dan ruang multidimensional yang mendekati arah yang sama. *Cosine Similarity* akan menghitung kemiripan antara dua data dalam bentuk vektor berdasarkan sudut antara vektor-vektor tersebut. Semakin kecil sudut antara vektor-vektor, maka data-data tersebut memiliki tingkat kemiripan yang tinggi (Lindang dkk., 2023).

Berikut adalah rumus dari *Cosine Similarity* :

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Keterangan :

A = vektor

B = vektor

A_i = bobot term i dalam blok A_i

B_i = bobot term i dalam blok B_i

I = jumlah term dalam kalimat

N = jumlah vektor

2.2.5. *Machine Learning*

Menurut Arthur Samuel, *machine learning* adalah bidang ilmu komputer yang memberikan kemampuan kepada komputer untuk mempelajari hal-hal tanpa instruksi pemrograman yang eksplisit. Menurut Mohri dan rekan-rekannya, *machine learning* adalah metode komputasi yang menggunakan pengalaman sebelumnya untuk meningkatkan kinerja dan membuat prediksi yang tepat. Pengalaman tersebut mencakup informasi sebelumnya yang dapat dijadikan data pembelajaran.

Dalam *machine learning*, terdapat beberapa skenario utama:

1. ***Supervised Learning***: Pembelajaran menggunakan data yang telah diberi label, memungkinkan komputer membuat prediksi berdasarkan data tersebut.
2. ***Unsupervised Learning***: Menggunakan data yang tidak memiliki label, komputer mencoba mengelompokkan data berdasarkan karakteristik yang ditemui.
3. ***Reinforcement Learning***: Fase pembelajaran dan tes terjadi secara bersamaan. Komputer belajar dari interaksi aktif dengan lingkungan untuk mendapatkan umpan balik atas setiap tindakan yang diambil.

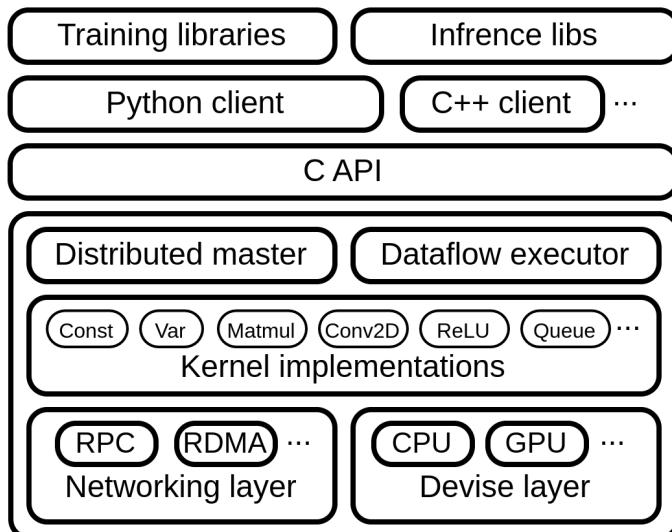
Saat ini, terdapat beragam pendekatan *machine learning* yang digunakan dalam banyak aplikasi, seperti deteksi spam, *Optical Character Recognition (OCR)*, pengenalan wajah, deteksi penipuan *online*, *Named Entity Recognition (NER)*, dan *Part-of-Speech Tagger*.

Adapun *library machine learning* yang biasa digunakan yaitu *Tensorflow Js*. *Tensorflow Js* adalah *library JavaScript* yang dikembangkan oleh *Google* untuk memudahkan pengembangan aplikasi *machine learning* di web. *Tensorflow Js* memungkinkan manusia untuk membangun dan melatih model *machine learning* di browser, sehingga manusia dapat membuat aplikasi *machine learning* yang dapat dijalankan di platform apa saja yang mendukung *browser*. Saat ini *tensorflow* sangat berperan penting dalam pada penelitian dalam *machine learning* dan penelitian *deep neural network* yang memerlukan komputasi paralel secara masif (Sutjiadi, Raymond & Pattiasina, Timothi Jhon. 2020). *Tensorflow* dapat melatih dan menjalankan *neural network* untuk keperluan mengklasifikasikan tulisan tangan, pengenalan gambar/objek, serta menggabungkan suatu kata (Prisky Ratna Aningtiyas, Agus Sumin, & Setia Wirawan. 2020). Beberapa fitur utama yang dimiliki *tensorflow* adalah:

1. Definisi, optimalisasi dan pemrosesan perhitungan dilakukan secara efisien juga dalam bentuk ekspresi matematis berisi *array multidimensi* dari bentuk atau disebut juga *tensors*.
2. Karakteristik dan sifat pembelajaran mesin pemrograman yang mendukung perhitungan jaringan saraf dalam. *Tensorflow* bisa memberikan informasi yang digunakan dalam proses melakukan tugas perhitungan ekspresi matematika.
3. *Graphics Processing Unit (GPU)* yang transparan serta memiliki fungsi optimalisasi dan otomatisasi manajemen memori yang sama dengan data yang digunakan. Dimana *tensorflow* memiliki kemampuan menulis kode

dan dapat bekerja dengannya secara maksimal di unit pemrosesan pusat atau *GPU*. Lagi lebih khusus lagi, *Tensorflow* dapat mengetahui bagian mana saja dari perhitungan akan dan harus diteruskan ke *GPU*.

4. Kumpulan data besar dan skalabilitas pemrosesan data yang tinggi di semua bagiannya, sehingga memberikan akurasi yang berjalan baik. *Arsitektur umum Tensorflow* dilihat pada gambar dibawah ini:



Gambar 2.3 *Architecture Umum Tensorflow*

2.2.6. Pemesanan

Pemesanan adalah suatu aktivitas yang dilakukan oleh konsumen sebelum membeli proses, pembuatan, dan cara memesan (tempat, barang, jasa) kepada orang lain.

2.2.7. *Android*

Menurut Wiwin Susanty (2019) *Android* adalah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi, bersama dengan lebih dari 34 perusahaan besar di dunia bersatu membentuk

aliansi bernama *OHA* (*Open Handset Alliance*) yang berguna untuk menyempurnakan sistem operasi baru ini.

Sedangkan menurut (Hartati et al. 2017) dalam jurnal Muhamad Muslihudin, dkk (2018) dikatakan bahwa *android* adalah sebuah sistem operasi pada *handphone* yang bersifat terbuka dan berbasis *Linux*. *Android* menyediakan *platform* terbuka (*open source*) sehingga memudahkan bagi para pengembang untuk menciptakan aplikasi mereka sendiri *android* awalnya dikembangkan oleh *Android, Inc.*, dengan dukungan *Google*, yang kemudian *android* dibeli oleh *Google* pada tahun 2005.

2.2.8. *Flutter*

Flutter adalah sebuah SDK atau *framework open source* yang dikembangkan oleh Google untuk membuat atau mengembangkan aplikasi yang dapat berjalan dalam sistem operasi *Android* dan *iOS*. *Flutter* menggunakan bahasa pemrograman *Dart* dalam pengkodean. Perbedaan *framework Flutter* dengan yang lainnya yaitu dalam build aplikasi, pada *framework* ini semua kodennya di compile dalam kode *native*-nya tanpa ada interpreter pada prosesnya sehingga proses compile-nya menjadi lebih cepat (Enggar Krisnada & Tanone, 2020).

Dart language merupakan salah satu bahasa pemrograman oleh *Google* yang merupakan bahasa *general-purpose* yang dapat digunakan untuk mengembangkan berbagai *platform* termasuk *web*, *mobile*. Bahasa ini juga merupakan bahasa standar yang digunakan dari *Flutter* (Hanif & Sinambela, 2020).

Flutter hanya memiliki satu mode rendering yang disebut *skia*. *Skia* adalah pustaka grafis sumber terbuka yang diakuisisi *Google* pada tahun 2005. Ini berjalan diberbagai *platform* yang berbeda karena menyediakan seperangkat *API* yang sama. Ini berfungsi sebagai mesin grafis untuk produk seperti *google chrome* dan *chrome OS*, dan *android*. Ini menjadikannya alat rendering grafis yang cocok untuk digunakan *Flutter*. *Skia* digunakan untuk *translate* kode yang dibuat menggunakan *dart* ke semua *platform* untuk di *render*.

2.2.9. *Supabase*

Supabase adalah *platform open-source* yang menyediakan solusi lengkap untuk pengembangan aplikasi *website* dan *mobile* dengan menggunakan *PostgreSQL* sebagai basis datanya. Ini menggabungkan beberapa layanan seperti database *PostgreSQL*, otentikasi pengguna, penyimpanan *file*, dan *real-time subscriptions* dalam satu *platform*.

Supabase telah menarik perhatian pengembang dengan menyediakan serangkaian layanan yang umumnya diperlukan dalam pengembangan aplikasi modern dalam satu tempat. Kombinasi kekuatan *PostgreSQL* dengan kemudahan penggunaan layanan terintegrasi membuat *Supabase* menarik bagi pengembang yang mencari platform yang kuat dan efisien untuk aplikasi mereka.

2.2.10. *Use Cases*

Use cases dinyatakan secara sederhana di bawah deskripsi urutan peristiwa yang jika digabungkan menyebabkan sistem melakukan sesuatu yang berguna. sesederhana kedengarannya ini penting. Ketika dihadapkan hanya dengan

setumpuk persyaratan seringkali tidak mungkin untuk memahami apa yang benar-benar diinginkan oleh penulis persyaratan untuk dilakukan oleh sistem.

Tabel 2.1. *Use Case*

No	Simbol	Keterangan
1		Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
2		<i>Use Case</i> : Abstraksi dan interaksi antara <i>system</i> dan <i>actor</i>
3		<i>Assocation</i> : Abstraksi dari penghubung antara <i>actor</i>
4		<i>Generalisasi</i> : Menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>use case</i>
5		Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
6		Menunjukkan bahwa suatu <i>use case</i> merupakan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

2.2.11. *Activity Diagram*

Activity Diagram adalah diagram yang menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.2. Simbol *Activity Diagram*

Simbol	Nama	Keterangan
1	2	3
	<i>Start Point</i>	Mempresentasikan dimulainya alur kerja atau sistem dalam <i>activity diagram</i> . Disimbolkan dengan lingkaran solid
	<i>End Point</i>	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	<i>Activity</i>	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
	<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
	<i>Fork</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel
	<i>Join</i>	Digunakan untuk menentukan kegiatan yang dibutuhkan

2.2.12. *Class Diagram*

Class diagram digunakan untuk menggambarkan struktur dalam objek sistem. Dengan ini menyatakan *Class Object* yang menyusun sistem dan juga hubungan antara *Class Object*. Kelas memiliki atribut dan metode atau operasi.

Tabel 2.3. Simbol *Class Diagram*.

Simbol	Nama	Keterangan
1	2	3
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.2.13. Visual Studio Code

Visual Studio Code (VSCode) adalah editor teks yang ringan dan kuat yang dibuat oleh *Microsoft* untuk sistem multi-pengguna, yang artinya juga tersedia

untuk *Linux*, *Mac*, dan *Windows*. Teks editor ini mendukung langsung bahasa pemrograman *JavaScript*, *TypeScript* dan *Node.js*, serta bahasa pemrograman lainnya melalui plugin yang dapat diinstal melalui *marketplace Visual Studio Code* (misal *C*, *C#*, *Python*, *Go*, *Java*, dll.). *Visual Studio Code* menawarkan banyak fitur, termasuk *Intellisense*, *integrasi Git*, *debugging*, dan fungsi ekstensi yang melengkapi kemampuan teks editor. Fitur-fitur ini meningkat dengan meningkatnya versi *Visual Studio Code*. *Visual Studio Code* diperbarui secara berkala setiap bulan, yang membedakan *VSCodium* dari editor teks lainnya. Editor teks *VSCodium* juga *open source*, di mana Anda dapat melihat kode sumber dan berkontribusi untuk pengembangannya. Kode sumber *VSCodium* juga dapat dilihat di tautan *Github*. Ini juga membuat *VSCodium* menjadi favorit di kalangan pengembang perangkat lunak, karena pengembang perangkat lunak dapat berpartisipasi dalam proses pengembangan *VSCodium* di masa mendatang.

2.2.14. *Black box*

Metode pengujian pemesanan sayur adalah pengujian *black box* dengan teknik partisi yang setara. Pengujian dilakukan secara manual dan otomatis menggunakan bantuan *tools Katalon*. Pengujian berguna untuk menentukan kinerja antara pengujian manual dan otomatis dari membandingkan waktu berjalan dengan fungsinya yang sudah berjalan dengan yang diharapkan. Hasil pengujian manual dan otomatis menunjukkan fungsi pada pemesanan sayur.

2.3. Penelitian Terkait

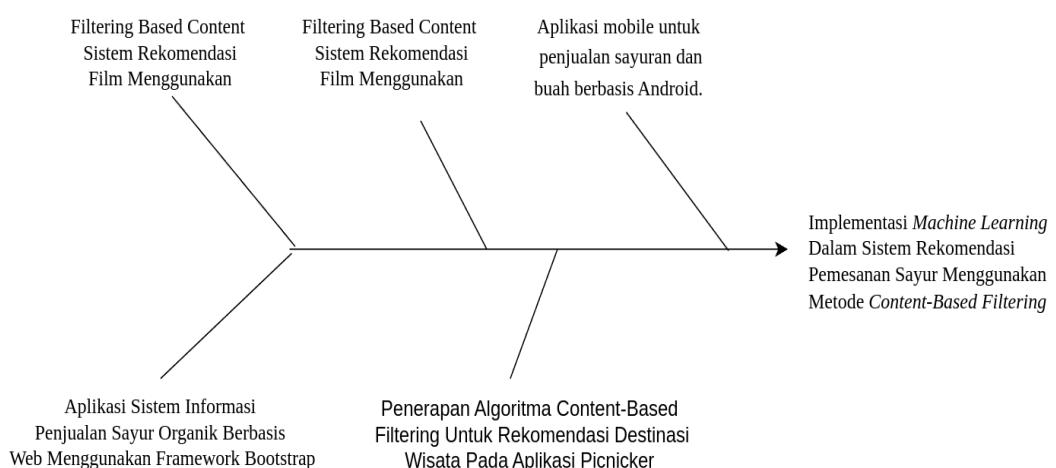
Dalam penyusunan penelitian ini, penulis sedikit banyak terinspirasi dan mereferensi dari penelitian-penelitian sebelumnya yang berkaitan dengan

penelitian ini. Berikut ini adalah penelitian terdahulu yang berhubungan dengan penelitian dari penulis antara lain :

Tabel 2.4. Penelitian Terkait

No	Judul Penelitian	Metode Penelitian	Hasil Penelitian	Perbedaan
1	Aplikasi pedagang sayur untuk pengiriman bahan sayuran dengan menggunakan <i>entity relationship</i> berbasis <i>Android</i> .	<i>entity relationship</i>	Aplikasi menjadi media penghubung antara pembeli dan penjual sehingga memudahkan dalam melakukan kegiatan jual beli.	penelitian ini menggunakan metode <i>entity relationship</i> , Sedangkan penelitian sekarang menggunakan metode <i>Content - Based Filtering</i>
2	Aplikasi <i>mobile</i> untuk penjualan sayuran dan buah berbasis <i>Android</i> .	<i>Scrum</i>	Aplikasi yang dibuat ini telah memiliki fitur yang membantu masyarakat dalam pemenuhan kebutuhan buah, sayur dan kebutuhan dapur lainnya, tanpa harus pergi ke pasar atau swalayan,	penelitian ini menggunakan metode <i>Scrum</i> , Sedangkan penelitian sekarang menggunakan metode <i>Content - Based Filtering</i>
3	Penerapan Algoritma <i>Content-Based Filtering</i> Untuk Rekomendasi Destinasi Wisata Pada Aplikasi <i>Picnicker</i>	<i>Content - Based Filtering</i>	Penerapan algoritma <i>Content-Based Filtering</i> Untuk Rekomendasi Destinasi Wisata Pada Aplikasi <i>Picnicker</i>	penelitian ini untuk rekomendasi objek wisata dan penelitian sekarang untuk pemesanan sayur.

No	Judul Penelitian	Metode Penelitian	Hasil Penelitian	Perbedaan
4	Aplikasi Sistem Informasi Penjualan Sayur Organik Berbasis Web Menggunakan <i>Framework Bootstrap</i>	<i>prototype</i>	terdapat 83.3% anak sekolah kekurangan konsumsi sayur dan buah di UPT Puskesmas Sawangan khususnya di MI Al Khoiriyah	penelitian ini menggunakan metode <i>prototype</i> , Sedangkan penelitian sekarang menggunakan metode <i>Content - Based</i>
5	Sistem Rekomendasi Film Menggunakan <i>Content-Based Filtering</i>	<i>Content - Based Filtering</i>	Jenis kueri multiple seeds kueri cukup berpengaruh terhadap hasil rekomendasi film menggunakan <i>content-based filtering</i> dengan fitur judul, genre, dan sinopsis, pembobotan <i>TF-IDF</i> , dan <i>cosine similarity</i> .	penelitian ini untuk rekomendasi film dan penelitian yang sekarang untuk pemesanan sayur.



Gambar 2.4. Kerangka penelitian terkait

BAB III

METODE PENELITIAN

3.1. Waktu dan Tempat Penelitian

Penelitian ini direncanakan akan dilaksanakan selama 3 bulan mulai bulan November 2023 s.d. Januari 2023 di Bukit Baruga, Desa Antang, Kec. manggala Kota Makassar, Sulawesi Selatan.

3.2. Alat dan Bahan

Dalam penelitian dibutuhkan beberapa alat dan bahan untuk merancang dan membuat perangkat lunak atau program dari hasil perancangan. Adapun alat penelitian yang akan digunakan adalah sebagai berikut :

3.2.1. Alat Penelitian

Hardware :

Satu unit laptop dengan spesifikasi antara lain:

1. Intel® Core™ i5-1035G1 @1.00 – 1.2 GHz
2. NVIDIA® GeForce MX™ 330
3. Memori 8 GB DDR4 RAM
4. 512 GB SSD

Software :

1. Sistem Operasi Ios
2. *Supabase*
3. *Visual Studio Code*
4. *Google Docs*

3.2.2. Bahan Penelitian

Adapun bahan penelitian yang digunakan adalah *VSCode* dalam pengembangan aplikasi.

3.3. Jenis Penelitian

Dalam menyelesaikan skripsi ini, jenis penelitian yang dilakukan adalah sebagai berikut :

1. Penelitian Kepustakaan (*Library Research*)

Penelitian ini dilakukan dengan menggunakan beberapa buku,jurnal dan hasil dari *internet* sebagai referensi untuk penulis, kemudian menyesuaikan dengan kemungkinan-kemungkinan yang akan terjadi dalam penyelesaian masalah.

2. Penelitian Lapangan (*Field Research*)

Penelitian ini akan dilakukan secara langsung terhadap objek yang akan diteliti dengan mengamati aktivitas-aktivitas pengolahan data pada objek yang akan diteliti.

3.4. Metode Pengumpulan Data

Adapun metode penelitian yang digunakan dalam pengumpulan data adalah sebagai berikut :

1. Metode Observasi

Melakukan pengamatan langsung terhadap objek yang akan diteliti untuk memperoleh data.

2. Metode Wawancara

Melakukan tanya jawab langsung terhadap responden yang akan diteliti untuk memperoleh data seperti pemimpin, staff produksi.

3. Study Literature

Dasar teori dikumpulkan melalui penelusuran *literatur* yang bersumber dari media *internet* dan hasil jurnal penelitian penulis lain.

3.5. Tahapan Penelitian

Tabel 3. 1 Tahapan Penelitian

No	Jenis Kegiatan	Waktu Penelitian												Waktu (Minggu)	
		Desember 2023				Januari 2024				Februari 2024					
		1	2	3	4	1	2	3	4	1	2	3	4		
1	Pengumpulan Data	Green	Red											2 Minggu	
2	Analisis dan Desain Objek			Red	Red									2 Minggu	
3	Perancangan Arsitektur dan Penulisan Kode Program					Red	Red	Red	Red	Red				6 Minggu	
4	Pengujian Sistem								Red	Red				3 Minggu	
5	Implementasi									Red				1 Minggu	

Keterangan Warna :

- : Belum Terlaksana
- : Terlaksana

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Perancangan Solusi

Pada tahapan ini merancang sebuah solusi dengan beberapa tahapan :

1. Mengidentifikasi kebutuhan dan tujuan

Tahap pertama adalah mengidentifikasi kebutuhan dan tujuan sistem aplikasi yang akan dirancang. Hal ini akan membantu dalam menentukan fitur-fitur yang diperlukan, target pengguna, tampilan dan fungsionalitas, serta mengukur keberhasilan sistem *website* nantinya.

2. Perencanaan dan desain

Setelah menentukan kebutuhan dan tujuan, tahap selanjutnya adalah membuat perencanaan dan desain sistem aplikasi. Hal ini meliputi pemilihan platform aplikasi, desain *layout* dan tampilan *website*, serta penentuan navigasi dan struktur situs. Perencanaan dan desain yang matang akan membantu dalam memastikan sistem *website* yang dihasilkan sesuai dengan kebutuhan dan tujuan.

3. Implementasi dan testing

Setelah desain sistem aplikasi sudah selesai, tahap berikutnya adalah implementasi dan testing sistem aplikasi. Pada tahap ini, pengembang aplikasi dapat menggunakan bahasa pemrograman seperti *HTML*, *CSS*, dan *JavaScript* untuk aplikasi web admin dan dart dari *framework flutter* untuk membuat aplikasi mobile pengguna. Selain itu, tahap ini juga mencakup

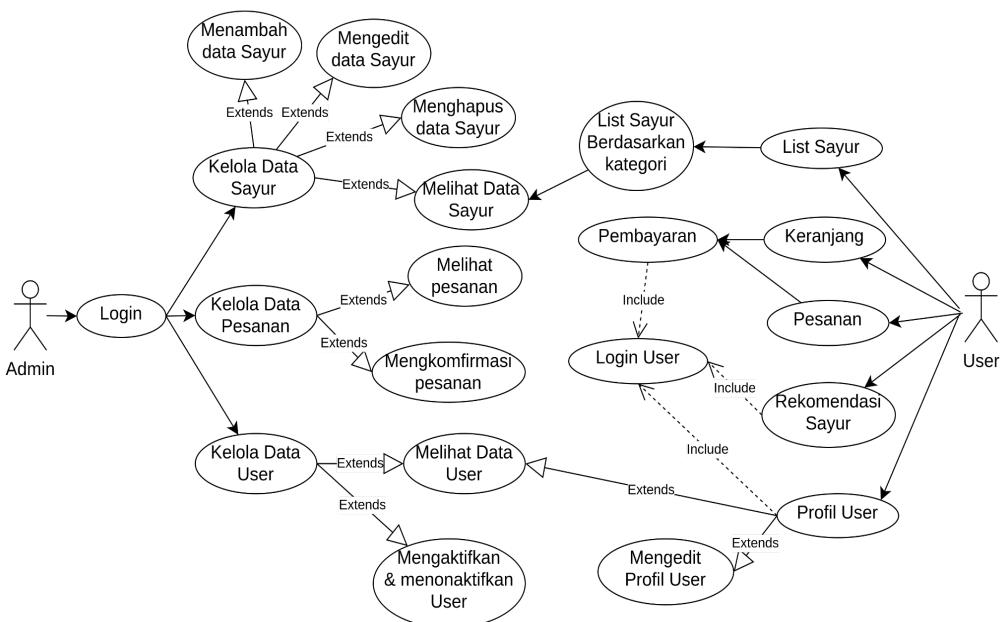
pengujian dan *debug* sistem aplikasi untuk memastikan bahwa semua fitur berjalan dengan baik.

4. Peluncuran

Setelah tahap pengembangan dan implementasi selesai, sistem aplikasi siap untuk diluncurkan. Tahap ini meliputi pemasangan sistem *website* pada server dan optimisasi performa sistem *website* agar bisa diakses dengan baik oleh pengguna. Selain itu, sistem *website* juga memerlukan pemeliharaan yang teratur, termasuk pembaruan keamanan dan peningkatan performa sistem.

4.1.1. *Use Case Diagram*

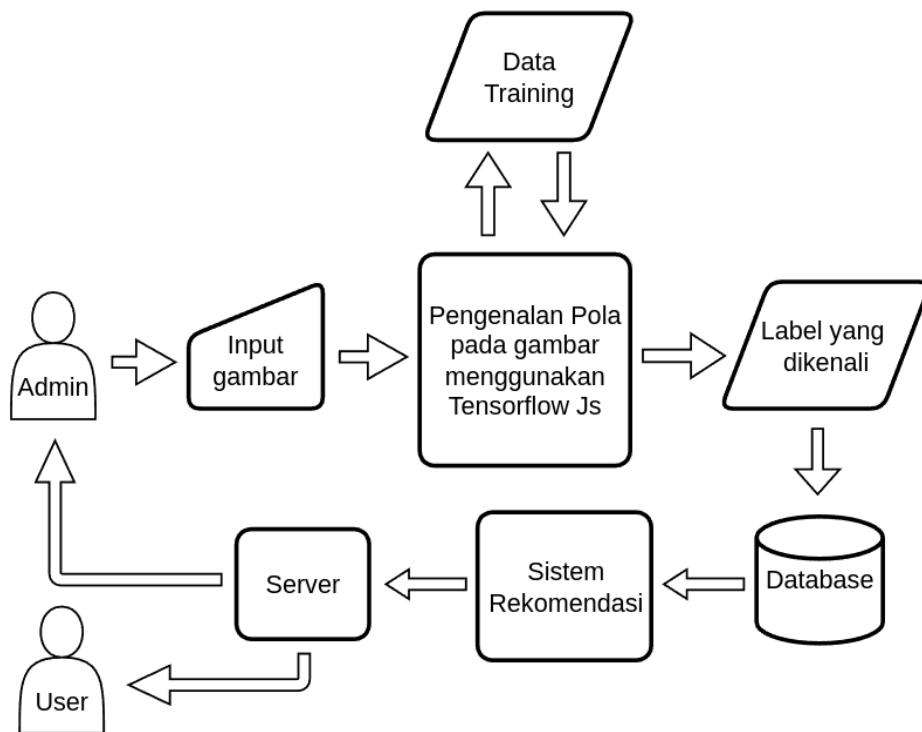
Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Adapun gambaran *use case* dari perancangan aplikasi pemesanan aplikasi sayur sebagai berikut ini :



Gambar 4.1 *Use case Diagram*

4.1.2. *Arsitektur Sistem*

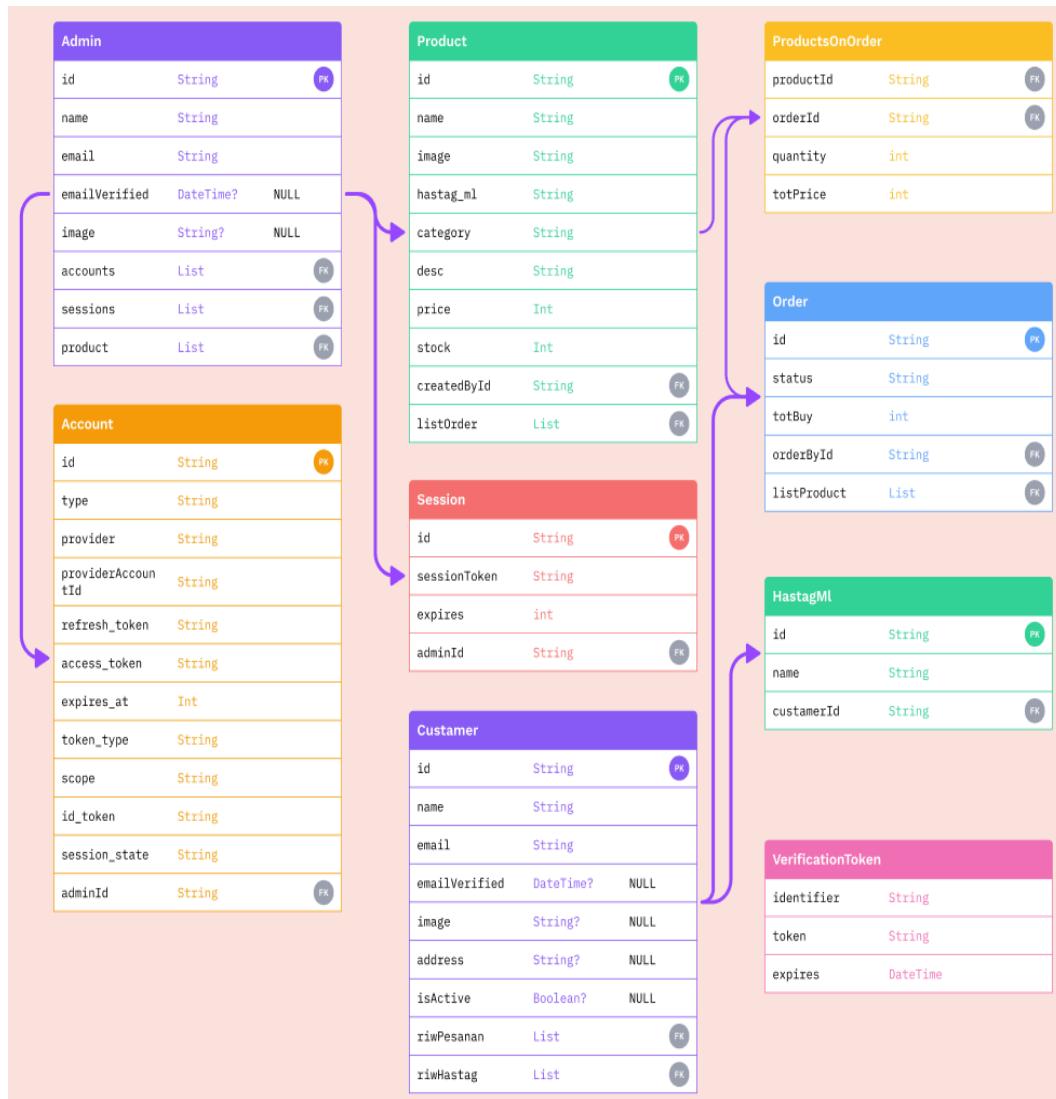
Arsitektur Sistem menggambarkan alur kerja sistem yang akan diimplementasikan dalam fitur utama pada aplikasi. Adapun gambaran *Arsitektur Sistem* dari perancangan aplikasi pemesanan sayur sebagai berikut ini :



Gambar 4.2 *Arsitektur Sistem*

4.1.3. *Class Diagram*

Class Diagram menjelaskan desain perancangan dari *database* yang akan dibuat, dimana terdapat sembilan entitas yang memiliki relasi satu sama lain dan entitas tersebut masing-masing mempunyai *primary key* sebagai atribut unik/ciri khas dari entitas tersebut dan *foreign key* yang menjadi patokan relasi dengan entitas yang lain. Adapun gambaran *Class Diagram* dari perancangan aplikasi pemesanan sayur sebagai berikut ini :

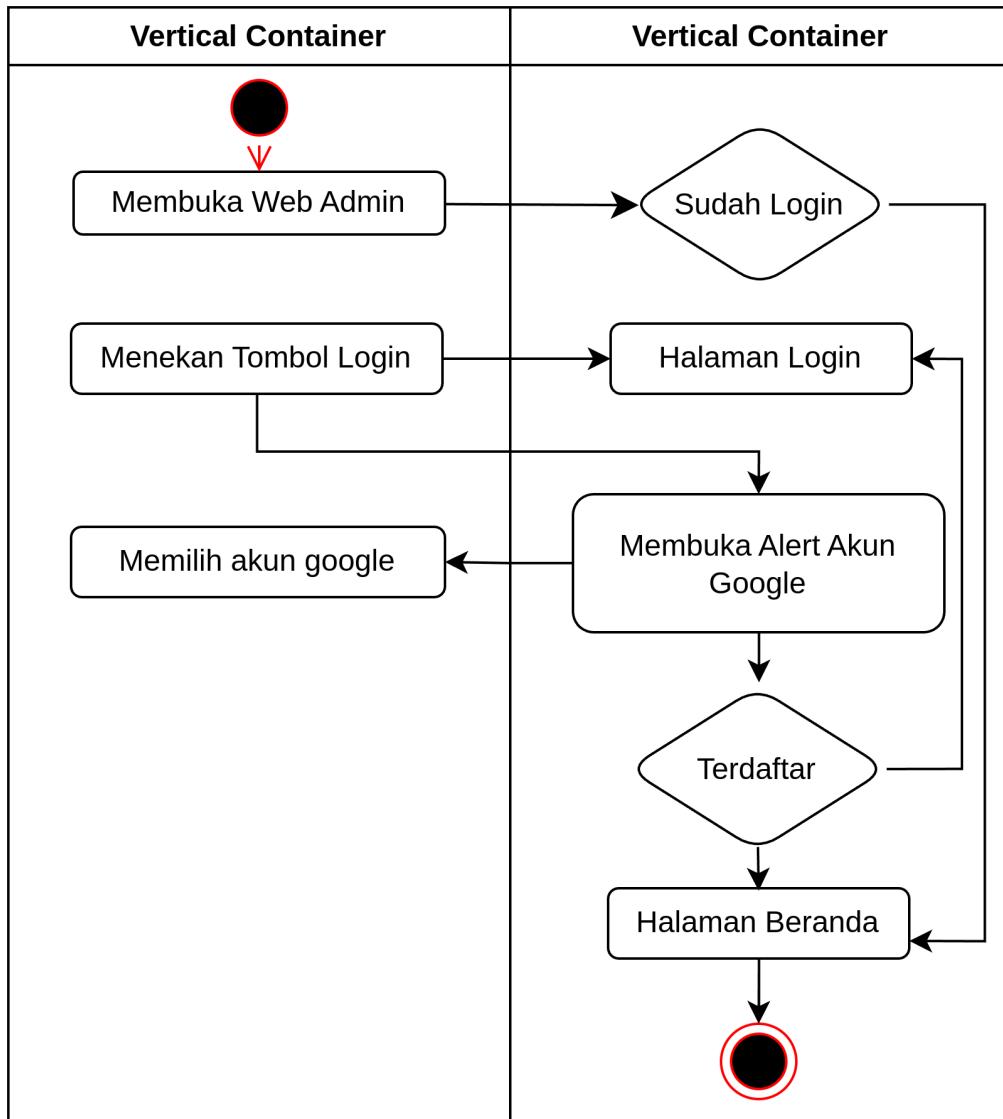


Gambar 4.3 *Class Diagram*

4.1.4. *Activity Diagram*

Activity diagram digunakan untuk menggambarkan proses bisnis yang kompleks, seperti alur kerja atau alur transaksi pada sistem informasi atau *software*. Diagram ini memudahkan kita dalam memahami alur kerja secara visual, sehingga mempermudah dalam analisis dan pengembangan sistem. Berikut *activity diagram* sistem yang akan dibuat:

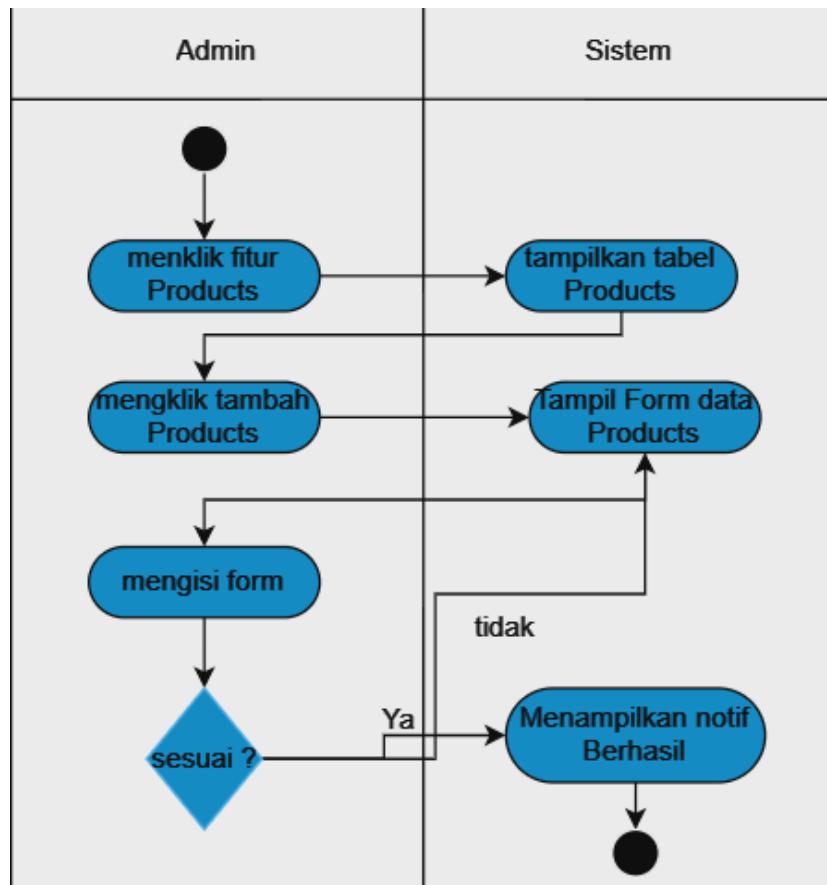
1. *Activity Diagram Login Admin*



Gambar 4.4 *Activity Diagram Login Admin*

Pada *activity diagram* login admin di atas dapat dilihat bahwa admin memberikan aksi yaitu menginput pengguna dan password lalu sistem merespon dengan memverifikasi pengguna dan *password* dan jika tidak sesuai maka tampilan beranda tidak tampil dan jika sesuai maka tampilan akan tampil.

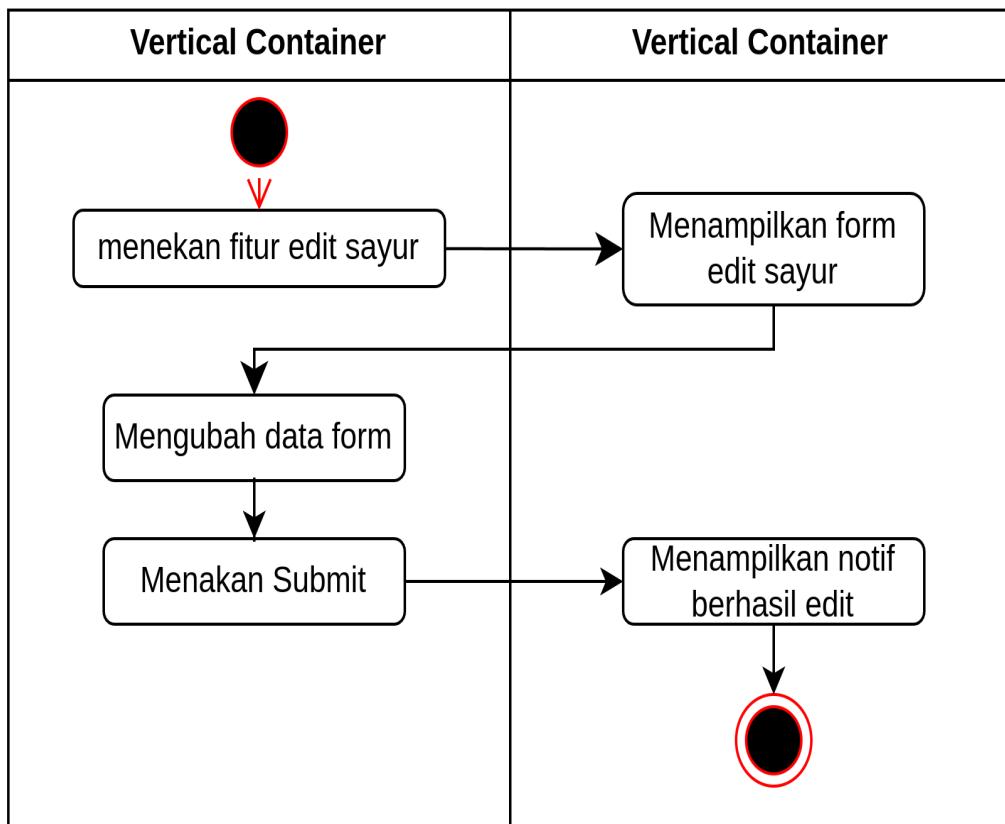
2. Activity Diagram Tambah Data Sayur



Gambar 4.5 Activity Diagram Tambah data sayur

Pada *activity diagram* tambah sayur diatas dapat dilihat bahwa admin memberikan aksi dengan mengklik fitur *products* dan sistem merespon dengan menampilkan tabel *products*. kemudian aksi berikutnya yang dilakukan admin adalah mengklik button tambah *products* dan sistem merespon dengan menampilkan form data *products* lalu admin memberikan aksi inputan ke dalam *form* dan sistem memverifikasi jika tidak sesuai maka sistem akan menampilkan *form* kembali dan sebaliknya jika sesuai akan menampilkan notif berhasil.

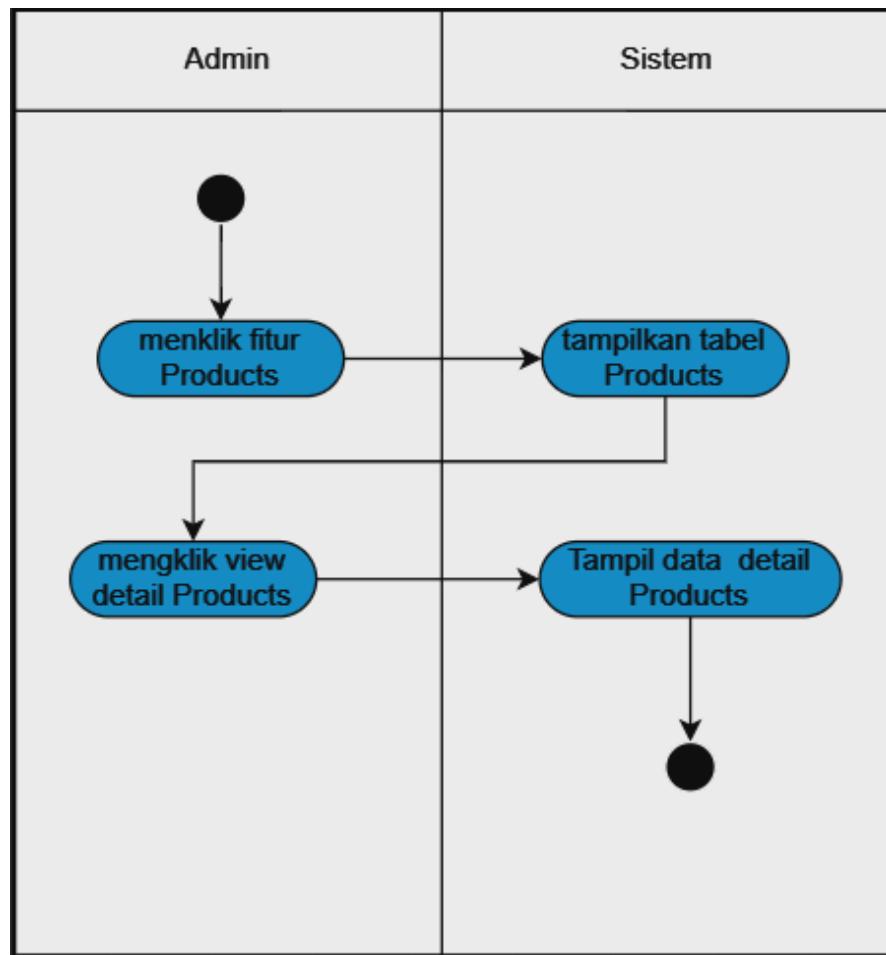
3. Activity Diagram Edit Data Sayur



Gambar 4.6 *Activity Diagram* Edit Data Sayur

Pada *activity diagram* edit data sayur, admin melakukan aksi membuka fitur edit *products* dan sistem menampilkan form edit setelah itu admin dapat melakukan aksi input kedalam form sehingga dapat mengubah data setelah itu aksi berikutnya adalah mengklik button agar sistem dapat memberikan respon dengan menampilkan notif berhasil.

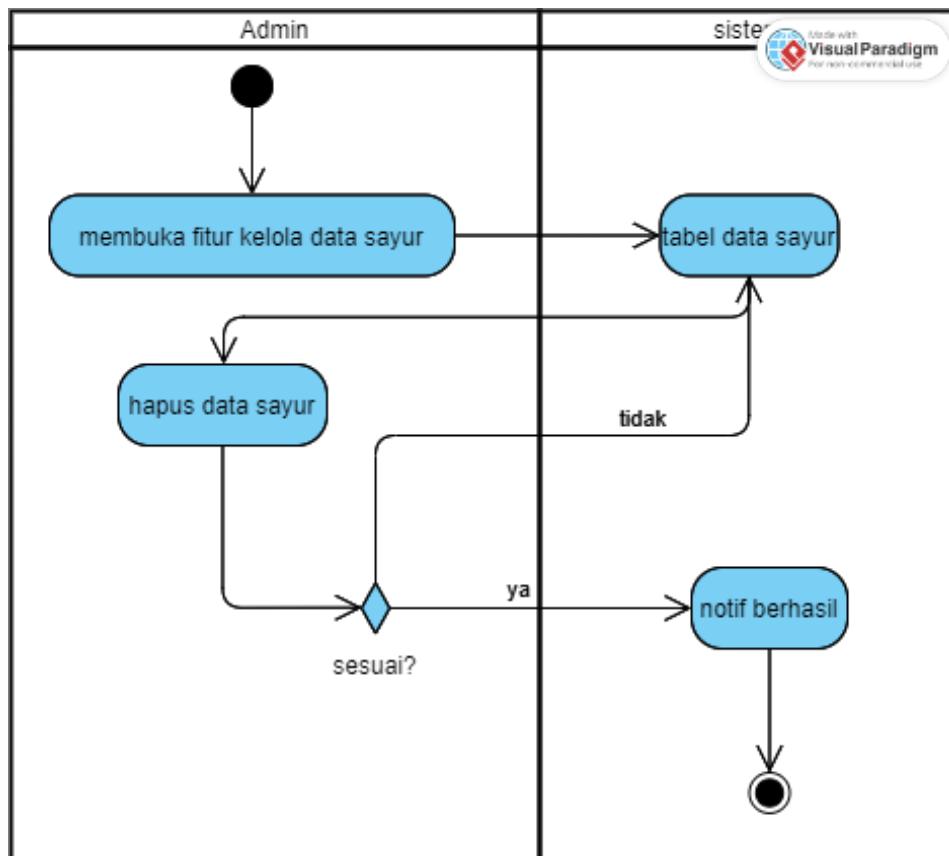
4. Activity Diagram Views Detail Data Sayur



Gambar 4.7 Activity Diagram Detail Data Sayur

Pada *activity diagram* detail data sayur, aksi admin adalah membuka fitur *products* dan sistem akan menampilkan tabel *products* lalu aksi admin berikutnya adalah membuka *views detail products* dan sistem akan memberikan tampilan data detail *products*.

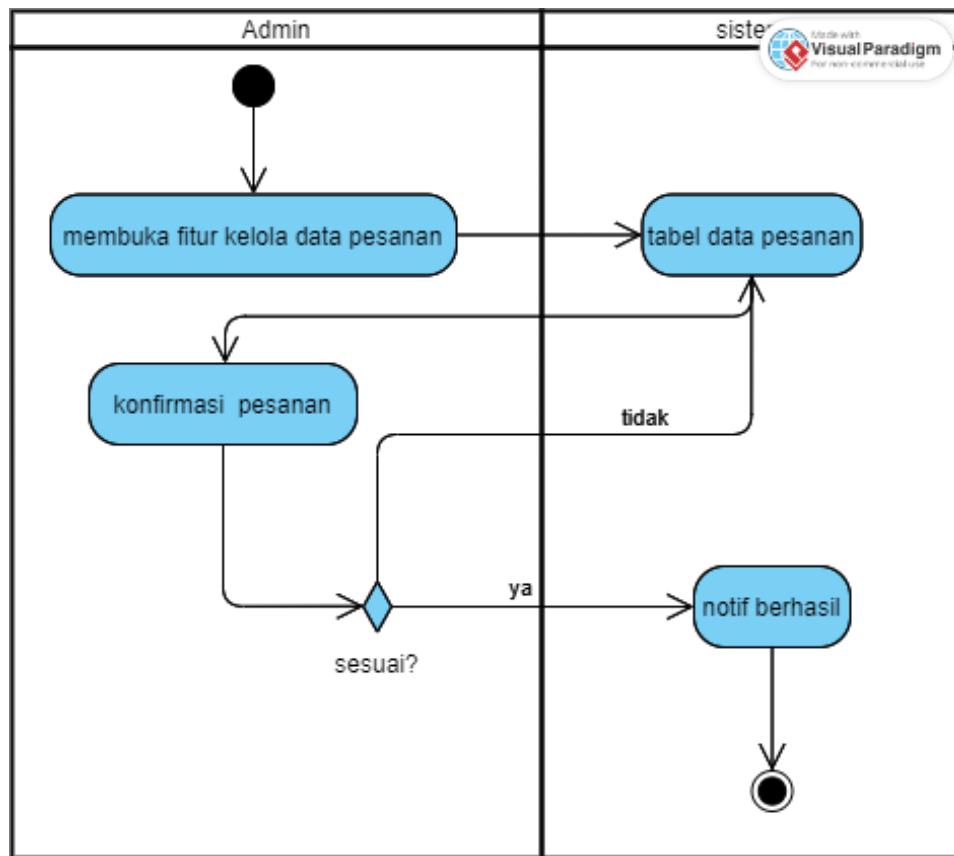
5. Activity Diagram Hapus Data Sayur



Gambar 4.8 Activity Diagram Hapus Data Sayur

Pada *activity diagram* hapus data *products*, aksi admin adalah membuka fitur kelola data sayur lalu sistem akan menampilkan data *products* dan admin dapat melakukan aksi hapus data *products* sistem akan memverifikasi dan ketika tidak sesuai maka akan tampil tabel data *products* dan jika sesuai maka notif berhasil yang akan tampil.

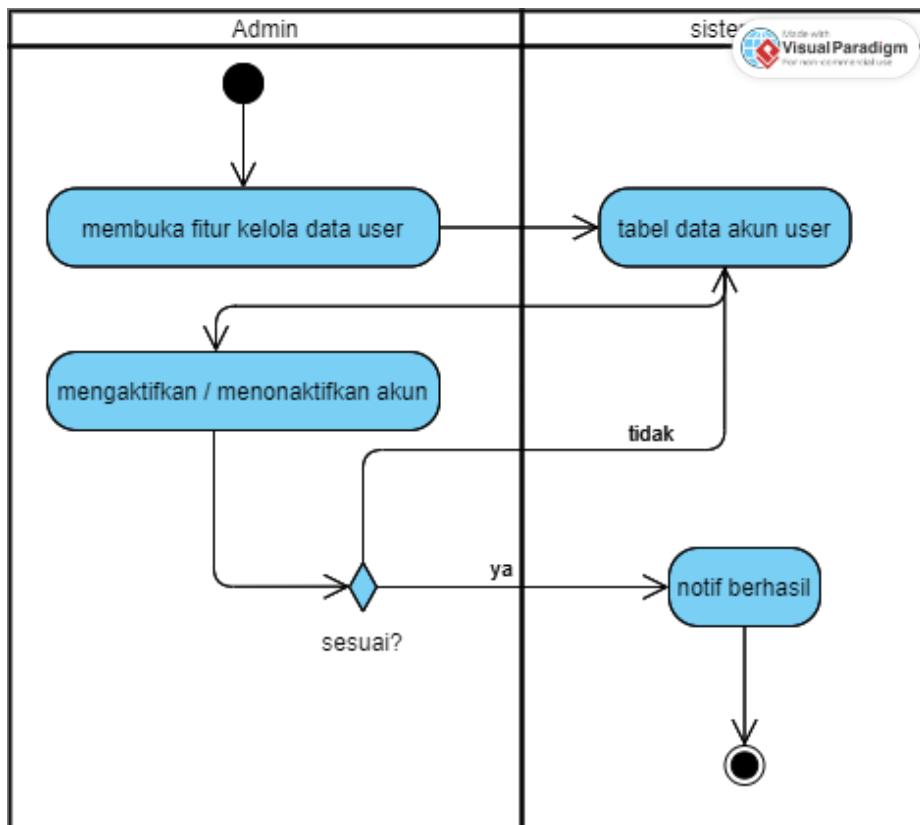
6. Activity Diagram Kelola Data Pemesanan



Gambar 4.9 *Activity Diagram* Kelola Data Pemesanan

Pada *activity diagram* kelola data pemesanan, aksi admin adalah membuka fitur kelola data pemesanan dan sistem akan menampilkan tabel data pemesanan lalu admin dapat melakukan aksi untuk konfirmasi pesanan dan jika sesuai maka sistem akan menampilkan notif berhasil dan jika tidak maka admin dapat membatalkan aksi tersebut lalu kembali ke halaman data pemesanan.

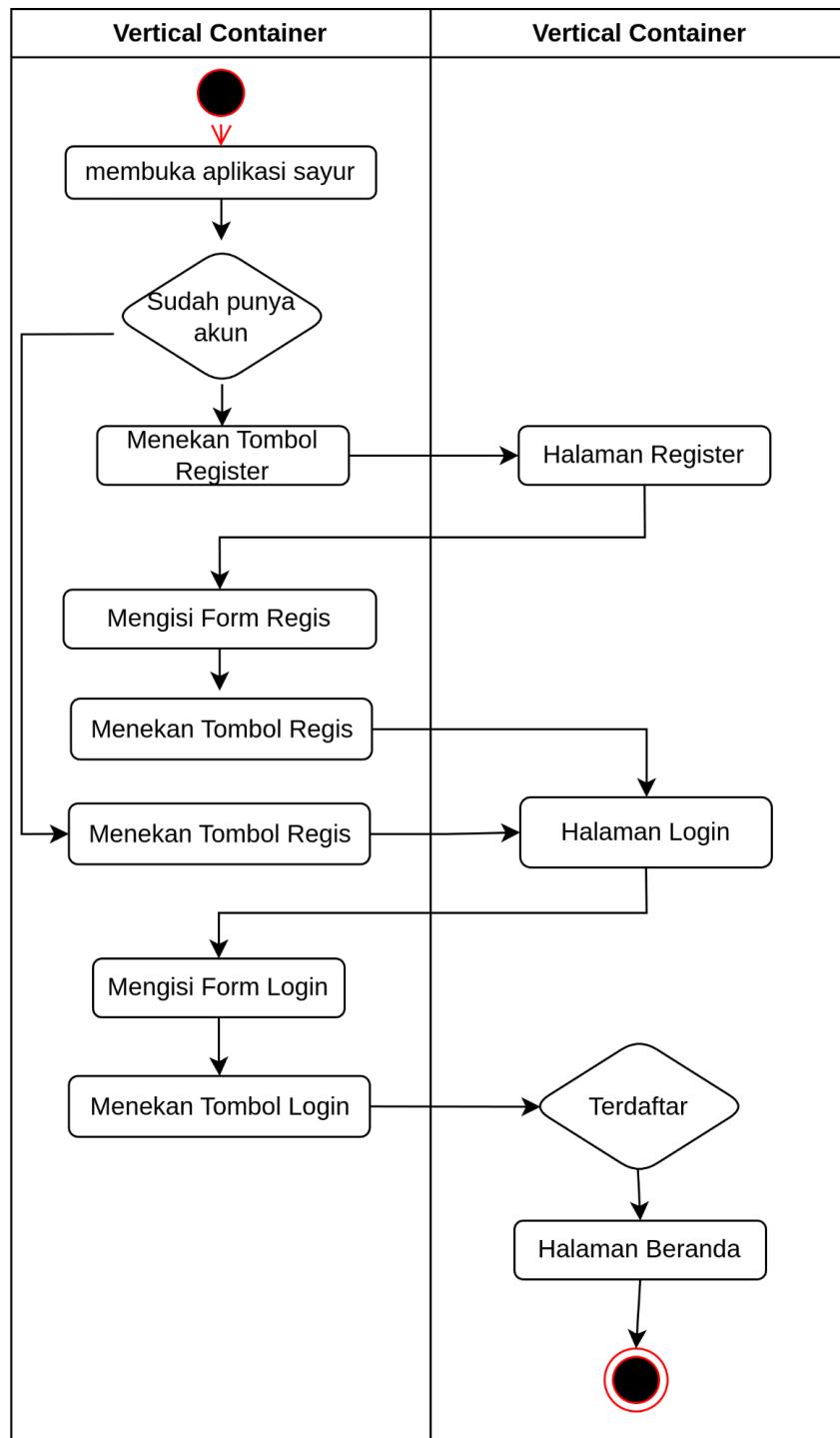
7. Activity Diagram Kelola Data Customer



Gambar 4.10 *Activity Diagram Kelola Data Customer*

Pada *activity diagram* kelola data *customer*, aksi admin adalah membuka fitur kelola data pengguna dan sistem akan menampilkan data pengguna setelah itu admin dapat melakukan aksi untuk mengaktifkan atau menonaktifkan akun pengguna setelah itu dapat konfirmasi iya atau tidak dan jika di konfirmasi tidak maka sistem akan menampilkan data pengguna dan jika konfirmasi iya maka sistem menampilkan notif berhasil.

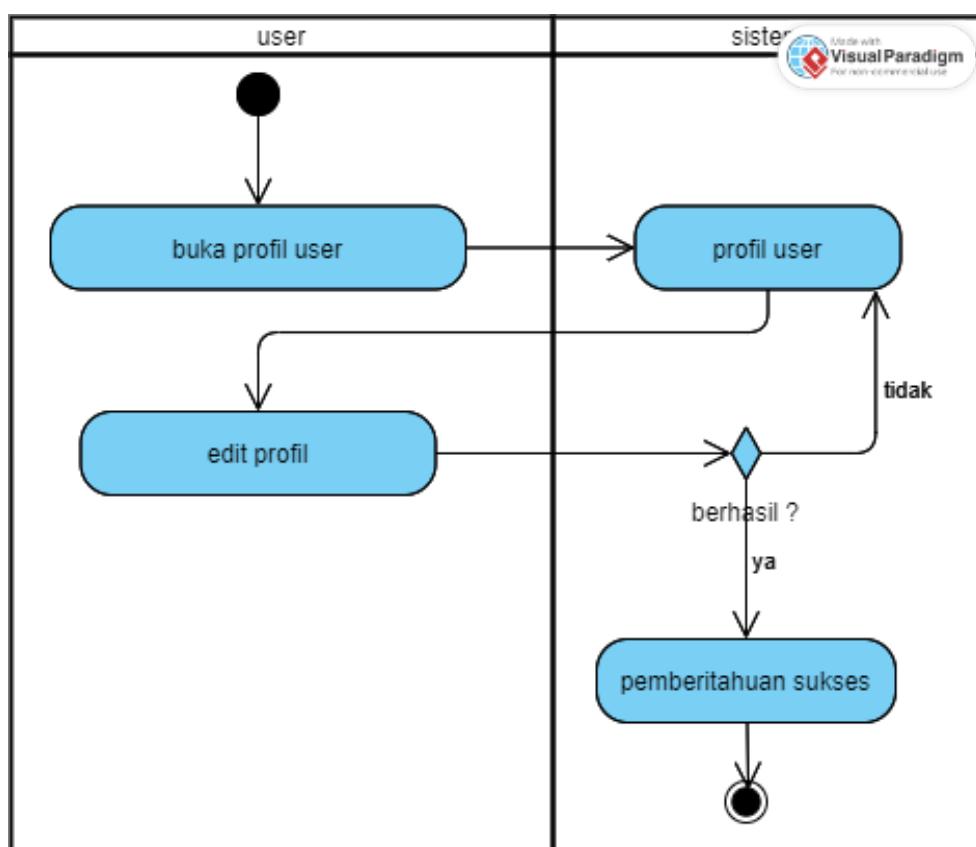
8. Activity Diagram Login Customer



Gambar 4.11 Activity Diagram Login Customer

Pada *activity diagram* login customer di atas dapat dilihat bahwa *customer* memberikan aksi yaitu menginput nama dan *email* pada *form register* lalu sistem merespon dengan memverifikasi nama dan *email* dan jika berhasil maka akan berpindah ke tampilan *login*, kemudian *customer* dapat *login* dengan *email* yang telah didaftarkan, dan jika berhasil akan kembali halaman beranda.

9. Activity Diagram Edit Profil Customer

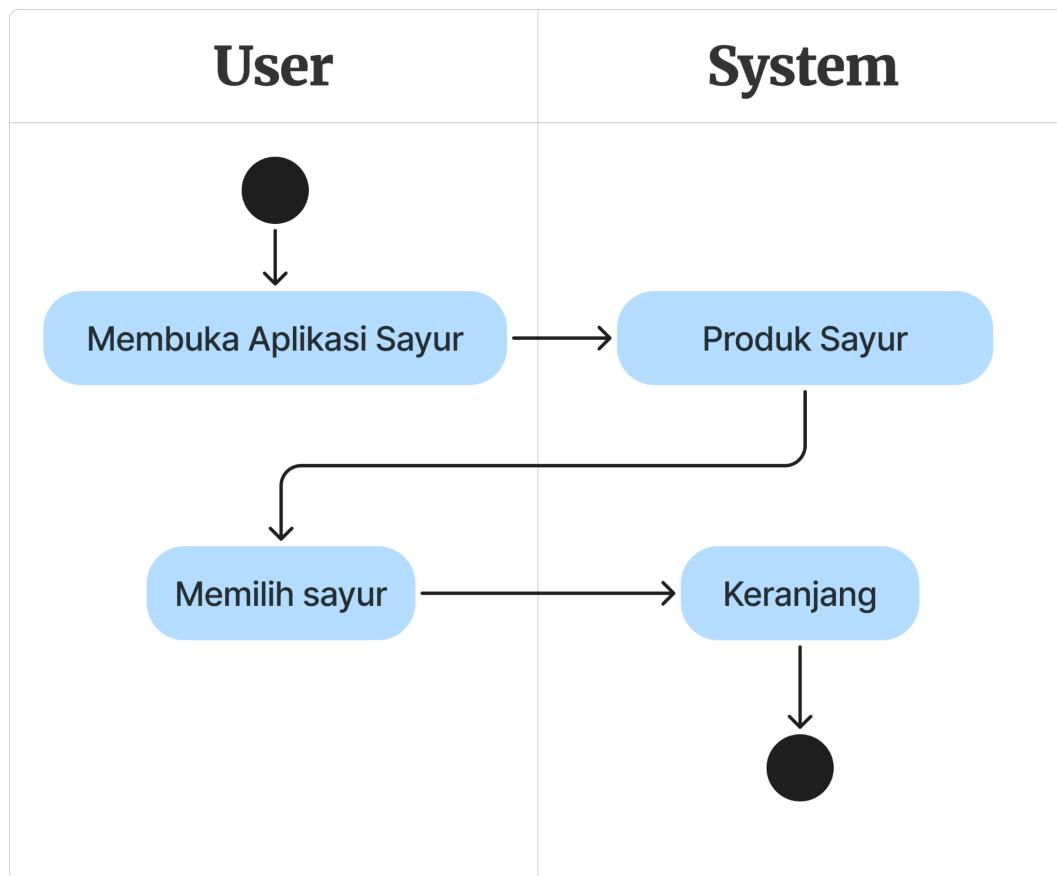


Gambar 4.12 *Activity Diagram Edit Profil Customer*

Pada *activity diagram* edit data profil, *Customer* melakukan aksi membuka profil, kemudian memilih fitur edit profil dan sistem menampilkan form edit setelah itu *Customer* dapat melakukan aksi input kedalam form sehingga dapat

mengubah data setelah itu aksi berikutnya adalah mengklik *button* agar sistem dapat memberikan respon dengan menampilkan notif berhasil.

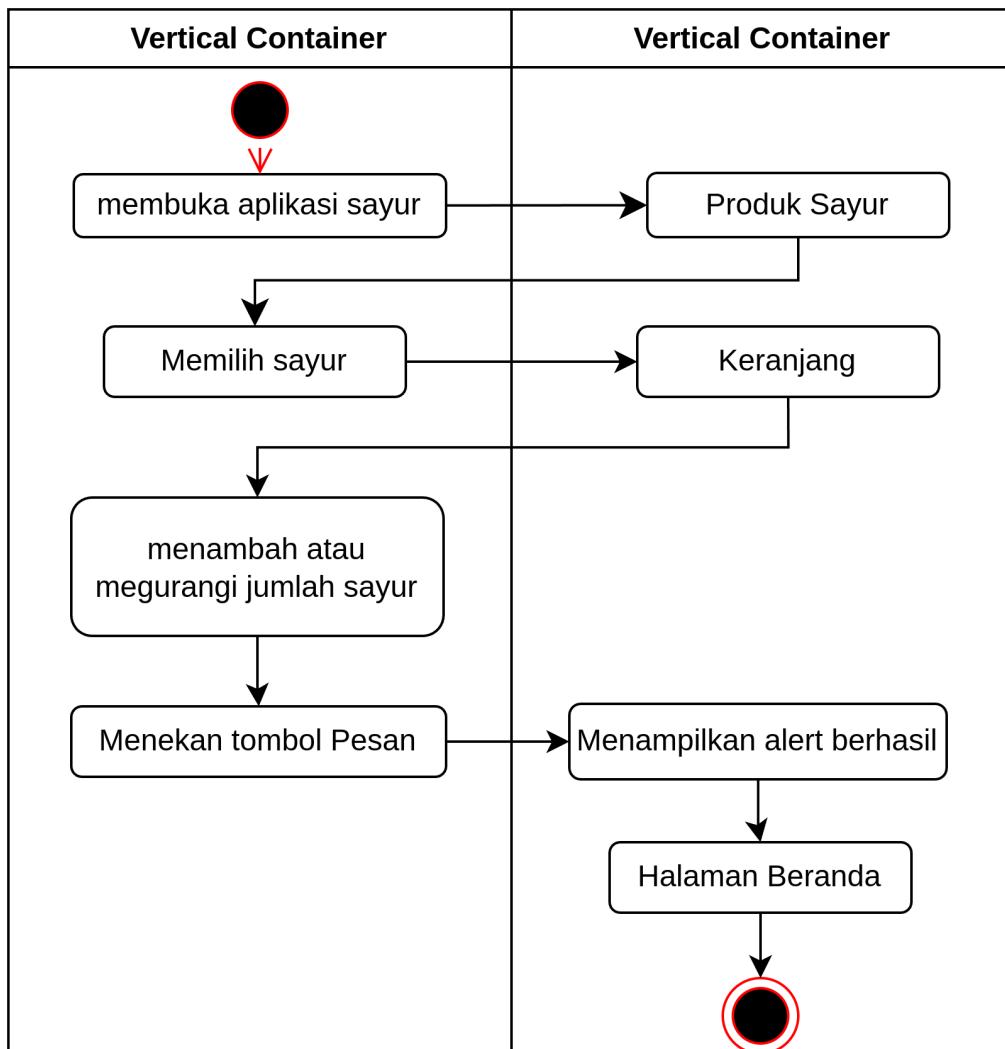
10. *Activity Diagram* Keranjang Customer



Gambar 4.13 *Activity Diagram* Keranjang Customer

Pada *activity diagram* keranjang, halaman ini dapat diakses oleh konsumen dengan cara memilih produk pada halaman *home*, kategori dan riwayat pesanan, setelah memilih produk yang ingin dibeli otomatis produk akan masuk ke keranjang belanja.

11. *Activity Diagram* Pemesanan Sayur



Gambar 4.14 *Activity Diagram* Pemesanan Sayur

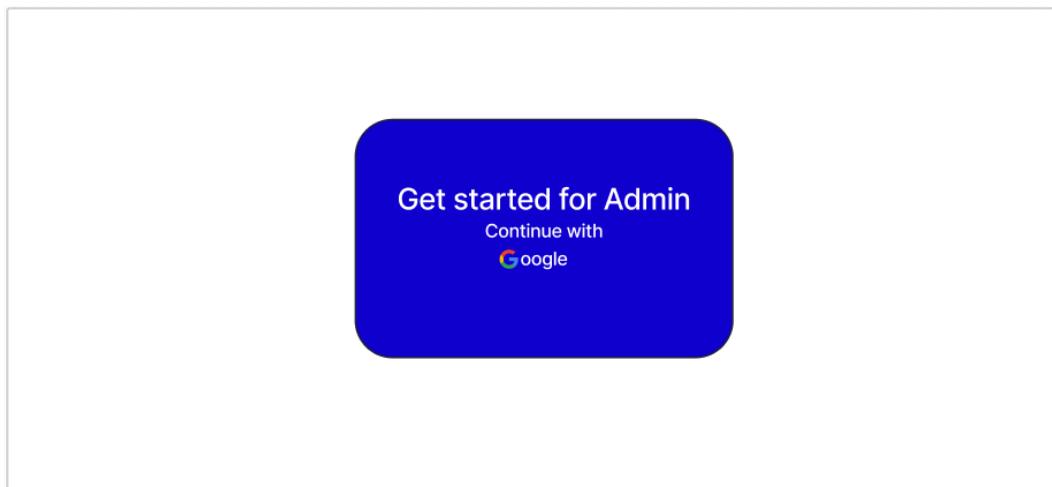
Pada *activity diagram* Pemesanan Sayur. Halaman ini dapat diakses oleh konsumen setelah halaman keranjang, setelah itu *customer* akan menekan tombol pesan untuk melakukan pemesanan.

4.1.5. Rancangan Aplikasi Admin

Pembahasan desain *interface* yang akan diimplementasikan pada aplikasi admin

1. Tampilan *Login* Admin

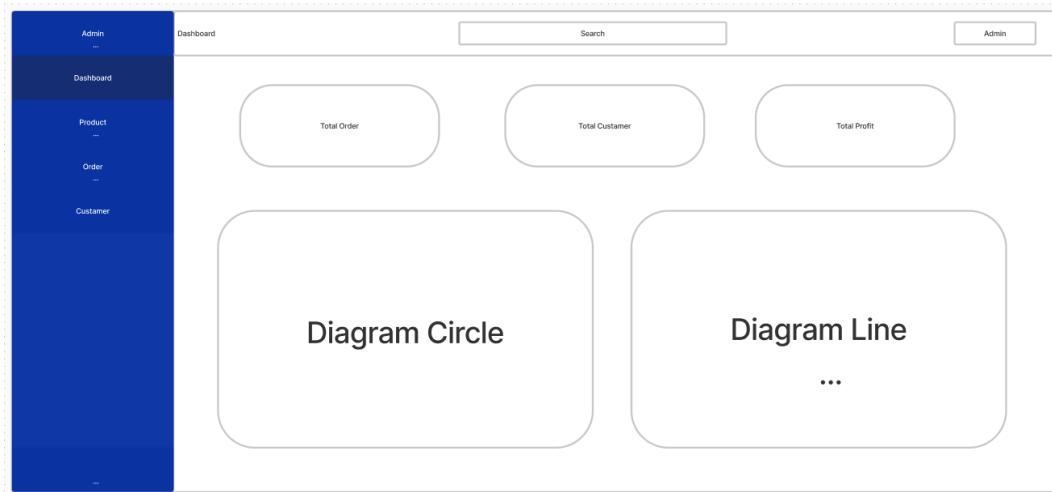
Tampilan *login* admin adalah tampilan pertama yang akan muncul ketika web admin di buka, yang dimana admin harus *login* terlebih dahulu untuk mengakses *dashboard* berikut rancangan tampilan *login* admin.



Gambar 4.15 Rancangan Tampilan *Login* Admin

2. Tampilan *Dashboard* Admin

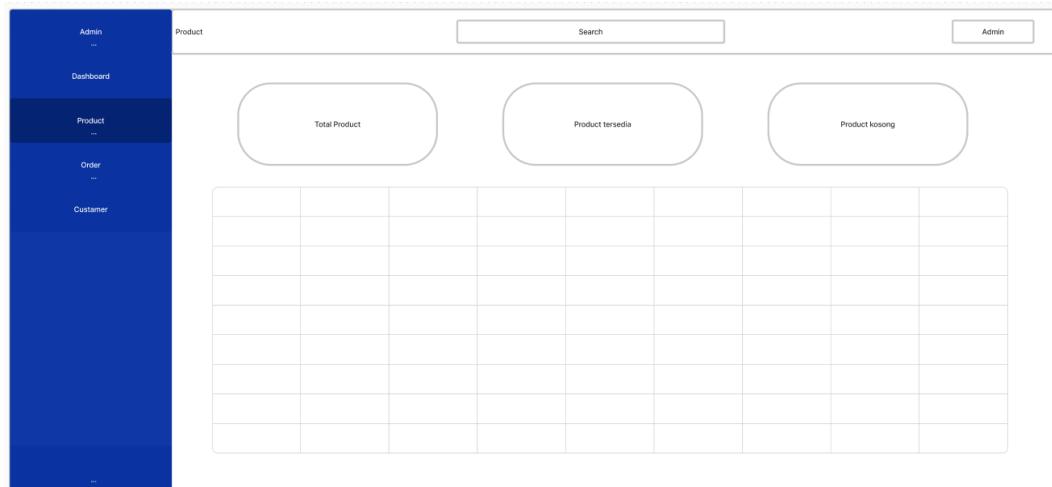
Dashboard admin adalah halaman utama yang tampil ketika admin selesai melakukan *login*, dalam *dashboard* menampilkan jumlah *customer*, transaksi dan pendapatan, adapun beberapa diagram yakni *diagram circle* untuk mempresentasikan jenis sayur, dan *diagram line* untuk mempresentasikan total transaksi dalam seminggu belakangan, berikut rancangan tampilan *dashboard* admin.



Gambar 4.16 Rancangan tampilan *dashboard* Admin

3. Tampilan Tabel *Product*

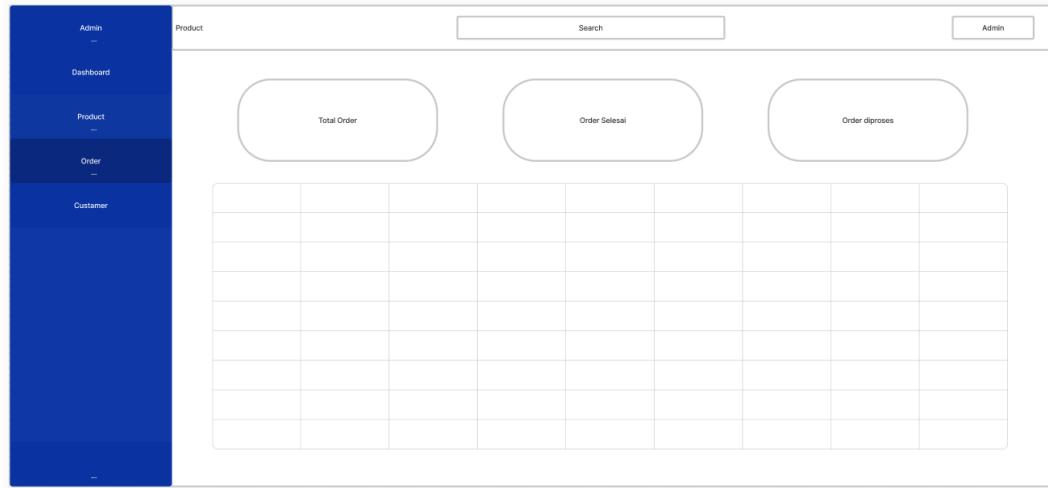
Tampilan tabel *product* adalah tampilan untuk melakukan aktivitas *CRUD* pada *product* sayur, berikut rancangan tampilan tabel *product*.



Gambar 4.17 Rancangan tampilan tabel *product* Admin

4. Tampilan Tabel Transaksi

Tampilan tabel *product* adalah tampilan untuk melakukan aktivitas *CRUD* pada transaksi yg telah dilakukan oleh customer. berikut rancangan tampilan tabel transaksi.



Gambar 4.18 Rancangan tampilan tabel transaksi Admin

5. Tampilan Tabel *Customer*

Tampilan tabel *product* adalah tampilan untuk melakukan aktivitas *CRUD* pada *customer*, berikut rancangan tampilan tabel *customer*.



Gambar 4.19 Rancangan tampilan tabel *customer* Admin

4.1.6. Rancangan Aplikasi *Mobile*

1. Tampilan Register

Tampilan register adalah Tampilan registrasi untuk *customer*, berfungsi untuk masuk kedalam sistem dengan melakukan registrasi dan akan

diberikan hak akses dan disimpan di *database*, berikut gambar rancangan tampilan register untuk *customer*.



Gambar 4.20 Rancangan tampilan *register customer*

2. Tampilan *Login*

Tampilan input di bawah ini berfungsi untuk pengguna dapat mengakses sistem dengan melakukan *login* menggunakan *email*, berikut gambar rancangan tampilan *login* untuk *customer*.



Gambar 4.21 Rancangan tampilan *login customer*

3. Tampilan *Home*

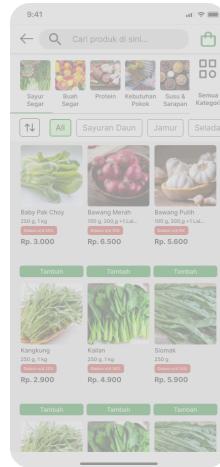
Tampilan *home* adalah tampilan utama pada aplikasi yang memiliki fitur rekomendasi sayur, dan fitur form *search* untuk mencari sayur yang diinginkan pengguna. berikut gambar rancangan tampilan *home* untuk *customer*.



Gambar 4.22 Rancangan tampilan *Home customer*

4. Tampilan Kategori

Tampilan kategori adalah tampilan yang menampilkan semua sayur beserta kategori sayur tersebut, dalam tampilan ini, *customer* melakukan filter berdasarkan kategori dan form *search* untuk mencari sayur yang diinginkan. berikut gambar rancangan tampilan kategori untuk *customer*.



Gambar 4.23 Rancangan tampilan kategori *customer*

5. Tampilan Riwayat Pesanan

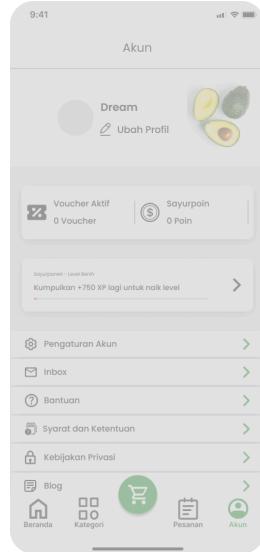
Tampilan riwayat pesanan adalah tampilan untuk melihat riwayat pesanan yang sudah selesai dan yang sementara diproses. berikut gambar rancangan tampilan riwayat pesanan untuk *customer*.



Gambar 4.24 Rancangan tampilan riwayat pesanan *customer*

6. Tampilan Profil

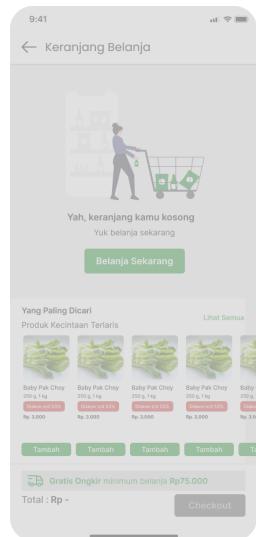
Tampilan Profil adalah tampilan untuk melihat info pribadi *customer* dan juga dapat mengedit info tersebut berikut gambar rancangan tampilan profil untuk *customer*.



Gambar 4.25 Rancangan tampilan profil *customer*

7. Tampilan Keranjang

Tampilan keranjang adalah tampilan untuk melihat total sayur yang ingin dipesan. dalam tampilan ini, *customer* dapat memilih beberapa sayur sekaligus dalam satu kali pemesanan, berikut gambar rancangan tampilan keranjang untuk *customer*.



Gambar 4.26 Rancangan tampilan keranjang *customer*

4.1.7. Implementasi *Machine Learning*

Pengimplementasian *Machine Learning* dalam aplikasi web admin digunakan dalam menambah atau melakukan *update* data sayur. Adapun kode lengkap dan dapat dilihat pada lampiran point pertama.

4.1.8. Implementasi *Algoritma Content-based filtering* pada *Back-End*

Dari gambar diatas, penerapan *Algoritma Content-based filtering* pada pembuatan *API*, yang dimana *API* hanya menerima dua parameter hastag machine learning, kemudian *API* akan digunakan di aplikasi android. Adapun kode lengkap dan dapat dilihat pada lampiran point kedua.

4.1.9. Implementasi *Algoritma Content-based filtering* pada *Android*

Pengimplementasian kode sistem rekomendasi yang menggunakan *algoritma Content-based filtering* yang berfokus pada karakteristik item. yang dimana karakteristik item (produk) direpresentasikan oleh *hastag machine learning*. Adapun kode lengkap dan dapat dilihat pada lampiran point ketiga.

4.2. Analisa dan Validasi Data

Perancangan sebuah sistem atau aplikasi dibutuhkan adanya analisis yang bertujuan untuk mendapatkan data-data yang akan dibutuhkan sistem dalam melakukan perancangan. Analisis sistem berupa analisis sistem baru, analisis kebutuhan perangkat keras, kebutuhan perangkat lunak dan analisis kebutuhan informasi. Fungsi yang akan dicapai merupakan analisa fungsional dari sistem yang akan dirancang adalah sebagai berikut :

1. Pengguna dapat mengakses halaman utama.

2. Pengguna mendapatkan rekomendasi sayur dari algoritma *Content-Based Filtering*.
3. Pengguna dapat melihat sayur berdasarkan kategori
4. Pengguna dapat melihat riwayat pesanan yang telah selesai diantar atau yg sementara diproses.
5. Pengguna dapat memesan sayur untuk diantarkan ke alamat pengguna.
6. Pengguna dapat melihat informasi profil.
7. Pengguna dapat mengubah informasi profil.

Sedangkan data-data yang dibutuhkan sistem yaitu berupa sayur akan dikelola di aplikasi admin.

4.3. Hasil dan Pengujian

Hasil dan pengujian terbagi menjadi dua, yakni Admin dan pengguna, aplikasi Admin berupa *website*, sedangkan untuk pengguna berupa aplikasi *mobile*.

4.3.1. Proses *Machine Learning*

Pada proses ini penulis menggunakan model *MobileNet* yang telah dilatih dengan *dataset ImageNet* untuk klasifikasi dan deteksi sayur pada gambar. Pada kasus ini penulis menggunakan *library TensorFlow.js* untuk dapat menggunakan model data dari *MobileNet* untuk dipakai dalam sistem *website*. *MobileNet* sendiri merupakan arsitektur *Convolutional Neural Networks (CNN)* sehingga membuat sistem dapat mengklasifikasi sebuah objek.. Adapun bagian dari lapisan arsitektur *CNN* dijelaskan pada tabel 1 :

Tabel 4.1 Algoritma *Convolutional Neural Network*

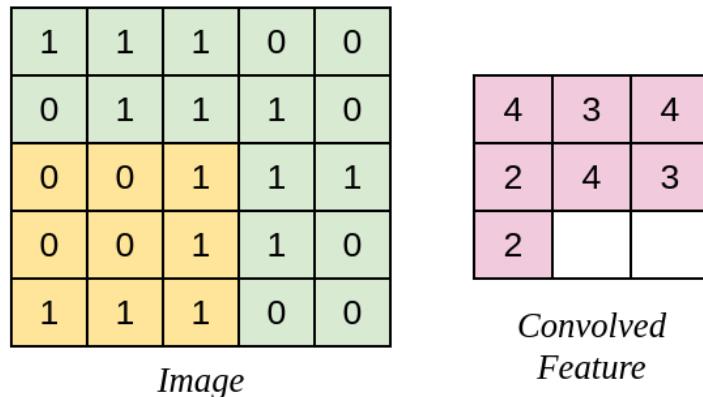
Algoritma Proses <i>Convolutional Neural Network</i>	
<i>Feature learning</i>	<i>Input Layer</i>
	<i>Convolution Layer</i>
	<i>Activation Layer</i>
	<i>Pooling Layer</i>
<i>Classification</i>	<i>Fully Connected Layer</i>
	<i>Output Layer</i>

1. *Input Layer*

Lapisan ini berguna untuk menampung *pixel value* dari citra yang diinputkan. Citra sayir yang telah diinputkan memiliki 3 channel warna *RGB (Red, Green, Blue)*.

2. *Convolution Layer*

Convolutional layer terdiri dari *neuron* yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (*pixels*). Proses konvolusi menggunakan *kernel* dan *stride*, proses konvolusi ini adalah proses kombinasi antara dua buah *matriks* yang berbeda untuk menghasilkan suatu nilai *matriks* yang baru. Dalam pengolahan citra, *konvolusi* berarti mengaplikasikan sebuah kernel (kotak kuning) pada citra di semua *offset* yang memungkinkan seperti ilustrasi pada gambar berikut,



Gambar 4.27 Gambaran Operasi *Konvolusi*

Kotak hijau secara keseluruhan adalah gambar yang akan dilakukan *konvolusi*. *Kernel* bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil *konvolusi* dari gambar dapat dilihat dari gambar di sebelah kanan. Tujuannya dilakukan *konvolusi* pada data gambar yaitu untuk mengekstraksi fitur dari gambar *input*.

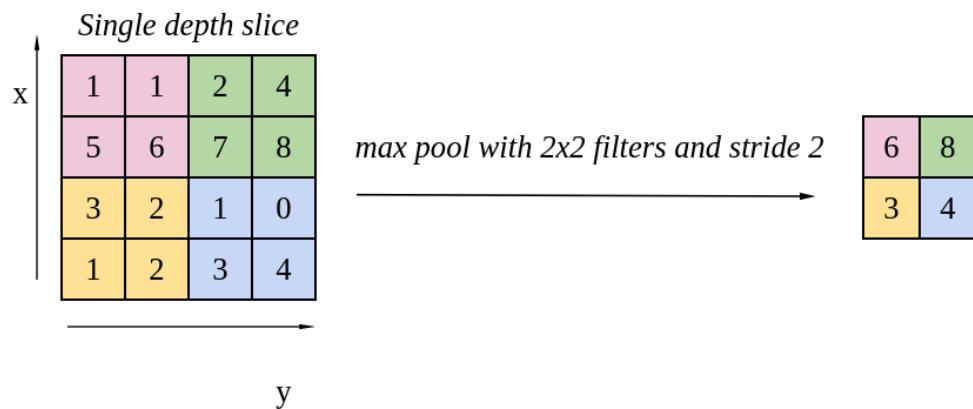
3. Activation layer

Pada lapisan ini, terjadi proses pengubahan nilai nilai *feature map* pada jarak tertentu tergantung pada fungsi aktivasi yang dipakai, yaitu fungsi aktifasi *ReLU*. Pada dasarnya fungsi *ReLU* (*Rectified Linear Unit*) melakukan “*threshold*” dari 0 hingga *infinity*. Fungsi ini menjadi salah satu fungsi yang populer saat ini.

4. Pooling Layer

Lapisan *pooling* bekerja di setiap tumpukan *feature map* dan melakukan pengurangan pada ukurannya. Bentuk lapisan *pooling* umumnya dengan menggunakan filter dengan ukuran 2x2 yang diaplikasikan dengan langkah

sebanyak dua dan beroperasi pada setiap irisan dari inputnya. Berikut ini adalah contoh gambar operasi *max-pooling*:

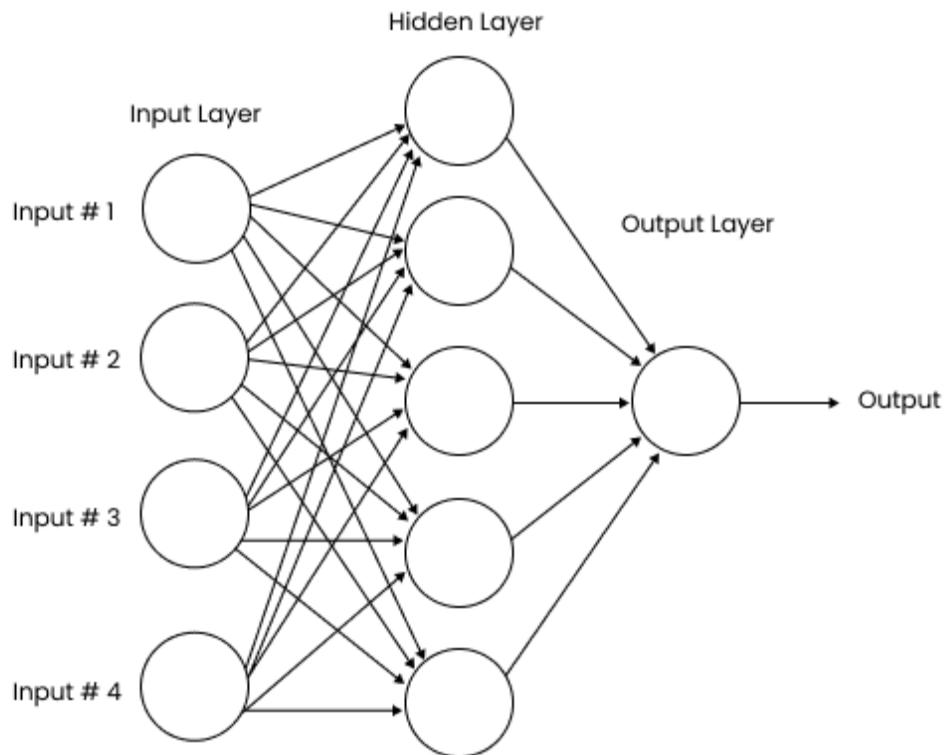


Gambar 4.28 *Max-Pooling*

Kotak yang berwarna merah, hijau, kuning dan biru pada sisi kiri merupakan kelompok kotak yang akan dipilih nilai maksimumnya. Sehingga hasil dari proses tersebut dapat dilihat pada kumpulan kotak disebelah kanannya.

5. *Fully connected layer*

Lapisan *Fully-connected* atau lebih dikenal dengan *dense layer* adalah lapisan dimana semua *neuron* aktivitas dari lapisan sebelumnya terhubung semua dengan *neuron* di lapisan selanjutnya seperti halnya jaringan syaraf tiruan.



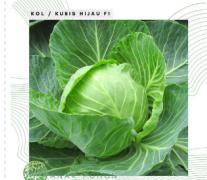
Gambar 4.29 *Fully Connected Layer*

6. *Output Layer*

Setelah itu dilanjutkan pada proses klasifikasi, yang mana dengan bantuan aktivasi *softmax* akan diklasifikasi input sesuai dengan target kategorinya yakni pada 1 jenis sayur yakni kubis.

Dari proses Algoritma *CNN* tersebut, didapatkan hasil pengklasifikasian objek sebagai berikut:

Tabel 4.2 Tabel Hasil Uji Coba Deteksi Objek

Objek	Harapan	Akurasi	Hasil
	Sistem berhasil melakukan prediksi pada gambar yang di input admin	99.94%	Berhasil menampilkan hasil prediksi
	Sistem berhasil melakukan prediksi pada gambar yang di input admin	99.97%	Berhasil menampilkan hasil prediksi
	Sistem berhasil melakukan prediksi pada gambar yang di input admin	99.73%	Berhasil menampilkan hasil prediksi

4.3.2. Proses Metode *Content-Based Filtering*

Metode *Content-Based Filtering* memiliki beberapa tahapan yang harus dilalui agar dapat mendapatkan hasil rekomendasi. Pada pembuatan suatu sistem rekomendasi tahapan yang harus dilalui yaitu *pre-processing*, pembobotan kata dan normalisasi. Berikut adalah penjelasan pada tahapan *Content-Based Filtering*.

1. *Pre-Processing*

Ada beberapa tahap dalam *Pre-Processing*, yakni *clean data*, *tokenisasi* dan *filtering*. Berikut ini adalah hasil dari tahapan *Pre-Processing*.

a) *Clean Data*

Tahapan ini memiliki tujuan untuk membersihkan data dari *noise* seperti mengubah data menjadi huruf kecil, menghilangkan tanda baca, dan menghilangkan angka pada data. Tahapan clean data pada pembuatan penelitian ini dapat dilihat pada tabel 1.

Tabel 4.3 Hasil Proses *Clean Data*

Sebelum	Sesudah
<i>Head Cabbage Head Cabbage</i> <i>Head Cabbage Head Cabbage</i> <i>Cauliflower</i>	<i>head cabbage head cabbage head</i> <i>cabbage head cabbage cauliflower</i>

b) *Tokenisasi*

Tokenisasi bertujuan untuk mengubah data teks menjadi kata-kata yang terpisah atau memisahkan kata-kata. Kata yang akan dibuat terpisah menjadi kata per kata. Proses *tokenisasi* ini juga memiliki tujuan agar mempermudah data teks untuk dihitung term dalam data. Tahapan *tokenisasi* pada penelitian ini dapat dilihat pada tabel 2.

Tabel 4.4 Hasil Proses *Tokenisasi*

Sebelum	Sesudah
<i>head cabbage head cabbage head</i> <i>cabbage head cabbage cauliflower</i>	<i>head, cabbage, head, cabbage,</i> <i>head, cabbage, head, cabbage,</i> <i>cauliflower</i>

c) *Filtering*

Tahapan *filtering* pada penelitian ini dapat dilihat pada tabel 3.

Tabel 4.5 Hasil Proses *Filtering*

Sebelum	Sesudah
<code>head, cabbage, head, cabbage,</code> <code>head, cabbage, head, cabbage,</code> <code>cauliflower</code>	<code>head, cabbage, cauliflower</code>

Pada tabel diatas menunjukkan proses memfilter atau menghilangkan kata-kata yang tidak dibutuhkan dalam proses analisis dokumen.

2. Pembobotan Kata

Setelah data telah melalui tahapan *pre-processing* selanjutnya data tersebut akan dihitung bobot kata pada data dengan menggunakan metode *TF-IDF*. Perhitungan ini mempertimbangkan frekuensi kata pada sebuah data dan frekuensi kata semua data. Berikut proses perhitungan dari *TF-IDF*.

a) *Term Frequency (TF)*

Rumus:

$$TF = \frac{\text{Jumlah kemunculan kata dalam dokumen}}{\text{Jumlah total kata dalam dokumen}}$$

Perhitungan:

kita memiliki sebuah dokumen dengan kata-kata berikut: `["head", "cabbage", "head", "cauliflower", "cauliflower", "cauliflower", "cauliflower"]`.

- 1). Kata "head" muncul 2 kali dari total 4 kata unik (["head", "cabbage", "cauliflower"]).
 $TF(\text{head}) = 2/4 = 0.5 = 0.5$

- 2). Kata "cabbage" muncul 1 kali dari total 4 kata unik.
 $TF(\text{cabbage}) = 1/4 = 0.5$

- 3). Kata "cauliflower" muncul 4 kali dari total 8 kata.
 $TF(\text{cauliflower}) = 4/8 = 0.125$

Hasil dari perhitungan Term Frequency pada penelitian ini dapat dilihat pada tabel 4.

Tabel 4.6 Perhitungan *Term Frequency*

Kata	TF
head	0.5
cabbage	0.5
cauliflower	0.125

b) *Inverse Document Frequency (IDF)*

Setelah mendapatkan nilai *Term Frequency* selanjutnya adalah menghitung *Inverse Document Frequency (IDF)* yaitu seberapa sering kata tersebut muncul pada data.

Rumus:

$$IDF = \log \left(\frac{\text{Jumlah total dokumen}}{1 + \text{Jumlah dokumen yang mengandung kata tersebut}} \right)$$

Perhitungan:

Misalnya, kita memiliki 3 dokumen:

- 1). Dokumen 1: `["head", "cabbage", "cauliflower", "cucumber", "cuke"]`
- 2). Dokumen 2: `["head", "cabbage", "artichoke", "globe", "artichoke"]`
- 3). Dokumen 3: `["cauliflower", "head", "cabbage", "broccoli"]`
- 4). Kata `"head"` muncul dalam 3 dari 3 dokumen.

$$IDF(\text{head}) = \log(3/1+3) = \log(3/4) = \log(1) = -0.12494$$

- 5). Kata `"cabbage"` juga muncul dalam 3 dari 3 dokumen.

$$IDF(\text{cabbage}) = \log(3/1+3) = \log(1) = -0.12494$$

- 6). Kata `"cauliflower"` muncul dalam 2 dari 3 dokumen.

$$IDF(\text{cauliflower}) = \log(3/1+2) = \log(3/3) = 0$$

- 7). Kata `"cucumber"` muncul dalam 1 dari 3 dokumen.

$$IDF(\text{cucumber}) = \log(3/1+1) = \log(3/2) = 0.17609$$

- 8). Kata `"cuke"` muncul dalam 1 dari 3 dokumen.

$$IDF(\text{cuke}) = \log(3/1+1) = \log(3/2) = 0.17609$$

- 9). Kata `"artichoke"` muncul dalam 1 dari 3 dokumen.

$$IDF(\text{artichoke}) = \log(3/1+1) = \log(3/2) = 0.17609$$

- 10). Kata `"globe"` muncul dalam 1 dari 3 dokumen.

$$IDF(\text{globe}) = \log(3/1+1) = \log(3/2) = 0.17609$$

- 11). Kata `"artichoke"` muncul dalam 1 dari 3 dokumen.

$$IDF(\text{artichoke}) = \log(3/1+1) = \log(3/2) = 0.17609$$

12). Kata "*broccoli*" muncul dalam 1 dari 3 dokumen.

$$IDF(\text{cauliflower}) = \log (3/1+1) = \log (3/2) = 0.17609$$

Hasil dari perhitungan *Inverse Document Frequency* pada penelitian ini dapat dilihat pada tabel 5.

Tabel 4.7 Perhitungan *Inverse Document Frequency*

Kata	IDF
<i>head</i>	-0.12494
<i>cabbage</i>	-0.12494
<i>cauliflower</i>	0
<i>cucumber</i>	0.17609
<i>cuke</i>	0.17609
<i>artichoke</i>	0.17609
<i>globe</i>	0.17609
<i>broccoli</i>	0.17609

c) *Inverse Document Frequency (IDF)*

Setelah mendapatkan nilai dari *TF* dan *IDF*, maka selanjutnya melakukan perhitungan untuk *TF-IDF* untuk setiap kata dalam setiap dokumen.

Rumus:

$$TF-IDF = TF \times IDF$$

Perhitungan:

1. Dokumen 1: ["head", "cabbage", "cauliflower", "cucumber", "cuke"]

a) Kata "head" :

- **TF:** Misalkan $TF = 0.5$ (sesuai contoh).
- **IDF:** -0.12494 (hasil perhitungan sebelumnya).

$$\begin{aligned} \text{TF-IDF(head)} &= 0.5 \times -0.12494 = -0.06247 \\ \text{\textbackslash times} &\quad -0.12494 &= \\ -0.06247 & \end{aligned}$$

b) Kata "cabbage" :

- **TF:** 0.5 (sesuai contoh).
- **IDF:** -0.12494 .

$$\begin{aligned} \text{TF-IDF(cabbage)} &= 0.5 \times -0.12494 = -0.06247 \\ 0.5 & \text{\textbackslash times} -0.12494 &= \\ -0.06247 & \end{aligned}$$

c) Kata "cauliflower" :

- **TF:** 0.125 (sesuai contoh).
- **IDF:** 0 .

$$\begin{aligned} \text{TF-IDF(cauliflower)} &= 0.125 \times 0 = 0 \\ \text{\textbackslash times } 0 &= 0 \end{aligned}$$

d) Kata "cucumber" :

- **TF:** 0.125 (asumsi nilai TF yang sama).
- **IDF:** 0.17609.

$$\text{TF-IDF}(\text{cucumber}) = 0.125 \times 0.17609 = 0.02201$$

$$\} = 0.125 \times 0.17609 = 0.02201$$

$$0.02201 \text{TF-IDF}(\text{cucumber}) = 0.125 \times 0.17609 = 0.02201$$

e) Kata "cuke" :

- **TF:** 0.125 (asumsi nilai TF yang sama).
- **IDF:** 0.17609.

$$\text{TF-IDF}(\text{cuke}) = 0.125 \times 0.17609 = 0.02201$$

$$\times 0.17609 = 0.02201$$

2. Dokumen 2: ["head", "cabbage", "artichoke", "globe", "artichoke"]

a) Kata "head"

- **TF:** 0.5 (misal nilai TF tetap 0.5).
- **IDF:** -0.12494.

$$\text{TF-IDF}(\text{head}) = 0.5 \times -0.12494 = -0.06247$$

$$\times -0.12494 = -0.06247$$

$$-0.06247 \text{TF-IDF}(\text{head}) = 0.5 \times -0.12494 = -0.06247$$

b) Kata "cabbage"

- **TF:** 0.5.
- **IDF:** -0.12494.

$$\text{TF-IDF(cabbage)} = 0.5 \times -0.12494 = -0.06247$$

$$0.5 \quad \times \quad -0.12494 \quad =$$

$$-0.06247$$

c) Kata "artichoke"

- **TF:** 0.4 (karena muncul 2 kali dalam dokumen, misal TF 0.4).
- **IDF:** 0.17609.

$$\text{TF-IDF(artichoke)} = 0.4 \times 0.17609 = 0.07044$$

$$0.4 \quad \times \quad 0.17609 \quad =$$

$$0.07044$$

d) Kata "globe"

- **TF:** 0.2 (misalkan TF untuk kata ini).
- **IDF:** 0.17609.

$$\text{TF-IDF(globe)} = 0.2 \times 0.17609 = 0.03522$$

$$\times 0.17609 = 0.03522$$

3. Dokumen 3: ["cauliflower", "head", "cabbage", "broccoli"]

a) Kata "head"

- **TF:** 0.333 (misal karena muncul 1 dari 3 kata di dokumen).
- **IDF:** -0.12494.

$$\text{TF-IDF(head)} = 0.333 \times -0.12494 = -0.04164$$

$$0.333 \quad \times \quad -0.12494 \quad =$$

$$-0.04164$$

b) Kata "cabbage"

- **TF:** 0.333.
- **IDF:** -0.12494.

$$\text{TF-IDF}(\text{cabbage}) = 0.333 \times -0.12494 = -0.04164 \text{ text}\{\text{TF-IDF}(\text{cabbage})\}$$

$$= \quad \quad \quad 0.333 \quad \quad \quad \backslash \text{times} \quad \quad \quad -0.12494 \quad \quad \quad = \\ -0.04164 \text{TF-IDF}(\text{cabbage}) = 0.333 \times -0.12494 = -0.04164$$

c) Kata "cauliflower"

- **TF:** 0.333.
- **IDF:** 0.

$$\text{TF-IDF}(\text{cauliflower}) = 0.333 \times 0 = 0 \text{ text}\{\text{TF-IDF}(\text{cauliflower})\} = 0.333$$

$$\backslash \text{times} 0 = 0 \text{TF-IDF}(\text{cauliflower}) = 0.333 \times 0 = 0$$

d) Kata "broccoli"

- **TF:** 0.333.
- **IDF:** 0.17609.

$$\text{TF-IDF}(\text{broccoli}) = 0.333 \times 0.17609 = 0.05866 \text{ text}\{\text{TF-IDF}(\text{broccoli})\} =$$

$$0.333 \quad \quad \quad \backslash \text{times} \quad \quad \quad 0.17609 \quad \quad \quad = \\ 0.05866 \text{TF-IDF}(\text{broccoli}) = 0.333 \times 0.17609 = 0.05866$$

Hasil dari perhitungan *TF-IDF* pada penelitian ini dapat dilihat pada tabel 6.

Tabel 4.8 Perhitungan *TF-IDF*

Kata	<i>TF-IDF(Dok 1)</i>	<i>TF-IDF(Dok 2)</i>	<i>TF-IDF(Dok 3)</i>
head	-0.06247	-0.06247	-0.04164
cabbage	-0.06247	-0.06247	-0.04164
cauliflower	0	-	0
cucumber	0.02201	-	-
cuke	0.02201	-	-
artichoke	-	0.07044	-
globe	-	0.03522	-
broccoli	-	-	0.05866

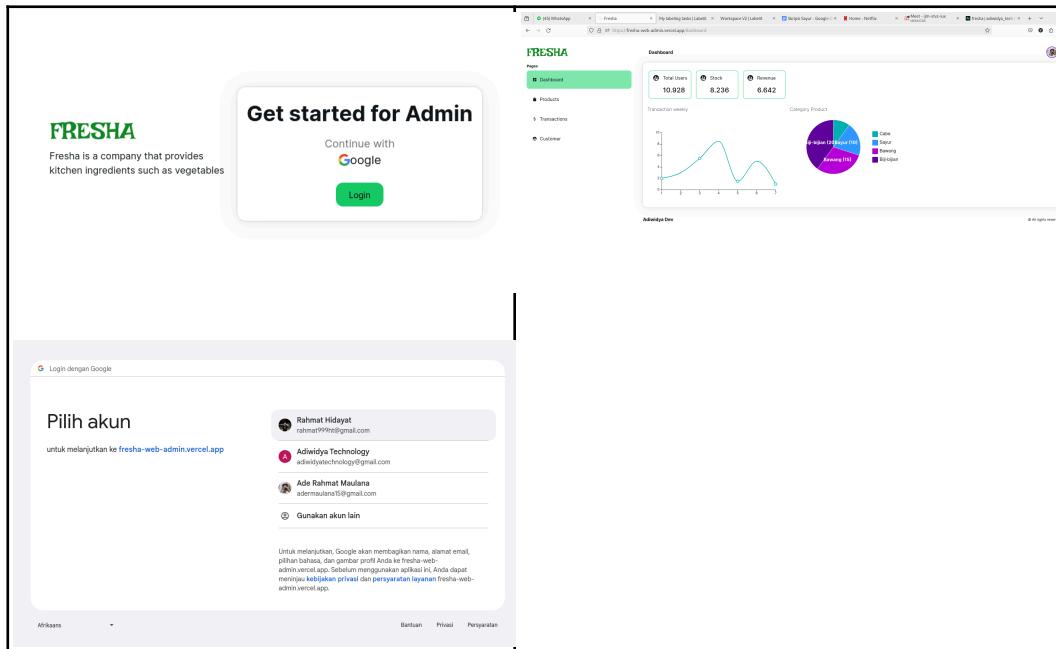
4.3.3. Aplikasi Admin

Pengujian Sistem yang dilakukan dengan menggunakan metode pengujian langsung berdasarkan teknik *Black Box* dengan menguji fungsionalitas dari aplikasi, tombol dan kesesuaian hasil aplikasi, berdasarkan pada pembahasan bab- bab sebelumnya hasil penelitian aplikasi admin sebagai berikut:

1. Pengujian *Login Admin*

Tabel 4.9 Pengujian *Login Admin*

<i>Test Factor</i>	<i>Hasil</i>	<i>Keterangan</i>
<i>Login Menggunakan Akun Google</i>	✓	Berhasil masuk halaman utama
<i>Screenshot</i>		
Sebelum		Sesudah

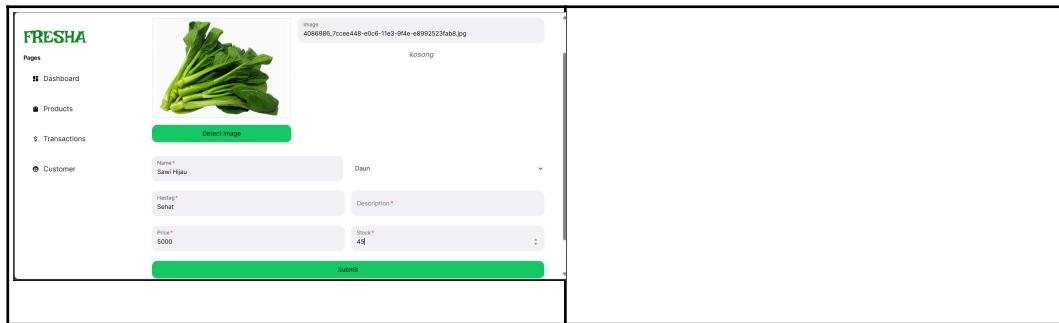


Pengujian login admin menggunakan akun *Google* berhasil, dan diarahkan ke halaman utama.

2. Pengujian Tambah *Product*

Tabel 4.10 Pengujian Tambah *Product* Admin

Test Factor	Hasil	Keterangan																											
Mengisi form dan menekan submit	✓	Berhasil menambah <i>product</i>																											
Screenshot																													
Sebelum		Sesudah																											
<p>FRESHA Pages Dashboard Products Transactions Customer</p> <p>Total Product: 4 Ready: 4 Empty: 0</p> <table border="1"> <thead> <tr> <th>NAME</th> <th>STOCK</th> <th>PRICE</th> <th>ACTIONS</th> </tr> </thead> <tbody> <tr> <td>Kubis Buncu</td> <td>122 Kg</td> <td>Rp. 20000</td> <td>⊕ ⊖ 🗑</td> </tr> <tr> <td>Kubis Unti</td> <td>100 Kg</td> <td>Rp. 20000</td> <td>⊕ ⊖ 🗑</td> </tr> <tr> <td>Kubis Njoo</td> <td>216 Kg</td> <td>Rp. 30000</td> <td>⊕ ⊖ 🗑</td> </tr> <tr> <td>Kubis Ungu</td> <td>225 Kg</td> <td>Rp. 20000</td> <td>⊕ ⊖ 🗑</td> </tr> </tbody> </table>		NAME	STOCK	PRICE	ACTIONS	Kubis Buncu	122 Kg	Rp. 20000	⊕ ⊖ 🗑	Kubis Unti	100 Kg	Rp. 20000	⊕ ⊖ 🗑	Kubis Njoo	216 Kg	Rp. 30000	⊕ ⊖ 🗑	Kubis Ungu	225 Kg	Rp. 20000	⊕ ⊖ 🗑								
NAME	STOCK	PRICE	ACTIONS																										
Kubis Buncu	122 Kg	Rp. 20000	⊕ ⊖ 🗑																										
Kubis Unti	100 Kg	Rp. 20000	⊕ ⊖ 🗑																										
Kubis Njoo	216 Kg	Rp. 30000	⊕ ⊖ 🗑																										
Kubis Ungu	225 Kg	Rp. 20000	⊕ ⊖ 🗑																										
<p>FRESHA Pages Dashboard Products Transactions Customer</p> <p>Total Product: 6 Ready: 6 Empty: 0</p> <table border="1"> <thead> <tr> <th>NAME</th> <th>STOCK</th> <th>PRICE</th> <th>ACTIONS</th> </tr> </thead> <tbody> <tr> <td>Gawii Hijau</td> <td>45 Kg</td> <td>Rp. 5000</td> <td>⊕ ⊖ 🗑</td> </tr> <tr> <td>Kubis Tunel</td> <td>120 Kg</td> <td>Rp. 20000</td> <td>⊕ ⊖ 🗑</td> </tr> <tr> <td>Kubis Buncu</td> <td>122 Kg</td> <td>Rp. 20000</td> <td>⊕ ⊖ 🗑</td> </tr> <tr> <td>Kubis Unti</td> <td>100 Kg</td> <td>Rp. 20000</td> <td>⊕ ⊖ 🗑</td> </tr> <tr> <td>Kubis Njoo</td> <td>216 Kg</td> <td>Rp. 30000</td> <td>⊕ ⊖ 🗑</td> </tr> <tr> <td>Kubis Ungu</td> <td>225 Kg</td> <td>Rp. 20000</td> <td>⊕ ⊖ 🗑</td> </tr> </tbody> </table>		NAME	STOCK	PRICE	ACTIONS	Gawii Hijau	45 Kg	Rp. 5000	⊕ ⊖ 🗑	Kubis Tunel	120 Kg	Rp. 20000	⊕ ⊖ 🗑	Kubis Buncu	122 Kg	Rp. 20000	⊕ ⊖ 🗑	Kubis Unti	100 Kg	Rp. 20000	⊕ ⊖ 🗑	Kubis Njoo	216 Kg	Rp. 30000	⊕ ⊖ 🗑	Kubis Ungu	225 Kg	Rp. 20000	⊕ ⊖ 🗑
NAME	STOCK	PRICE	ACTIONS																										
Gawii Hijau	45 Kg	Rp. 5000	⊕ ⊖ 🗑																										
Kubis Tunel	120 Kg	Rp. 20000	⊕ ⊖ 🗑																										
Kubis Buncu	122 Kg	Rp. 20000	⊕ ⊖ 🗑																										
Kubis Unti	100 Kg	Rp. 20000	⊕ ⊖ 🗑																										
Kubis Njoo	216 Kg	Rp. 30000	⊕ ⊖ 🗑																										
Kubis Ungu	225 Kg	Rp. 20000	⊕ ⊖ 🗑																										



Pengujian penambahan produk oleh admin berhasil, setelah mengisi form dan menekan tombol submit, produk berhasil ditambahkan.

3. Pengujian Edit Product

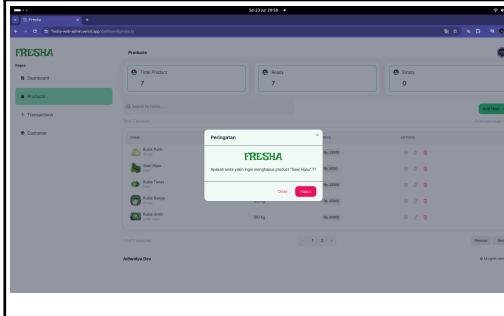
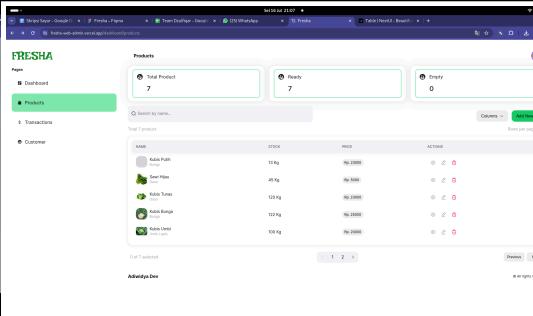
Tabel 4.11 Pengujian Edit Product Admin

Test Factor	Hasil	Keterangan
Mengisi form dan menekan <i>submit</i>	✓	Berhasil mengedit <i>product</i>
Screenshot		
Sebelum		
Sesudah		

Pengujian edit produk oleh admin berhasil, setelah mengisi form dan menekan tombol submit, produk berhasil diedit.

4. Pengujian Hapus Product

Tabel 4.12 Pengujian Hapus Product Admin

Test Factor	Hasil	Keterangan
Mengisi form dan menekan <i>submit</i>	✓	Berhasil mengedit <i>product</i>
Screenshot		
Sebelum		Sesudah
		

Pengujian hapus produk oleh admin berhasil, setelah mengisi form dan menekan tombol submit, produk berhasil dihapus.

5. Pengujian Identifikasi Image dengan Machine Learning

Tabel 4.13 Pengujian Identifikasi Image dengan Machine Learning Admin

Test Factor	Hasil	Keterangan
Memilih <i>image</i> dan menekan tombol <i>identify</i>	✓	Berhasil menampilkan hasil identifikasi
Screenshot		
Sebelum		Sesudah



Pengujian identifikasi *image* menggunakan *machine learning* berhasil, setelah memilih *image* dan menekan tombol *identify*, hasil identifikasi berhasil ditampilkan.

6. Pengujian *Transaction*

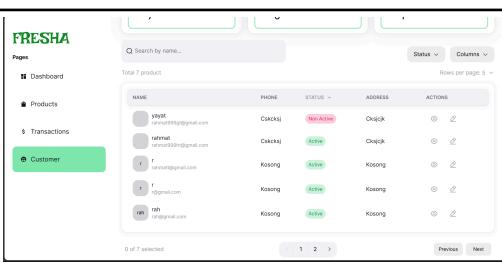
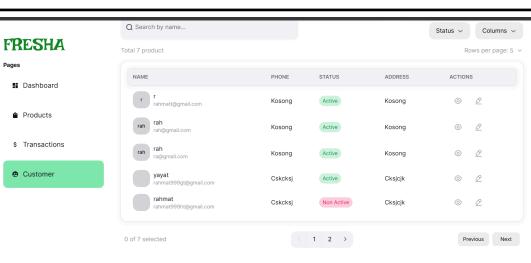
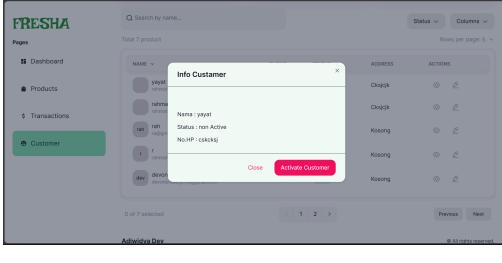
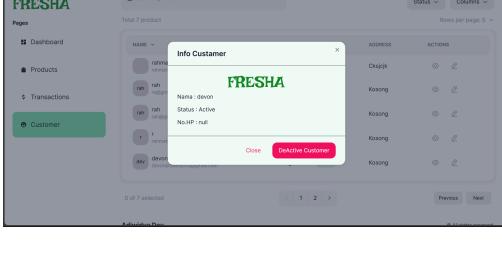
Tabel 4.14 Pengujian *Transaction* Admin

Test Factor	Hasil	Keterangan																														
Mengubah status pesanan menjadi selesai	✓	Berhasil mengubah status pesanan																														
Screenshot																																
Sebelum		Sesudah																														
<table border="1"> <thead> <tr> <th>CUSTOMER</th> <th>STATUS</th> <th>TOTAL BUY</th> <th>TOTAL PRODUCT</th> <th>ACTIONS</th> </tr> </thead> <tbody> <tr> <td>rahnut rahnut@gmail.com</td> <td>Done</td> <td>Rp. 3000</td> <td>2</td> <td>⋮ ⚡</td> </tr> <tr> <td>devon devonprasya@gmail.com</td> <td>Processed</td> <td>Rp. 3000</td> <td>2</td> <td>⋮ ⚡</td> </tr> </tbody> </table>		CUSTOMER	STATUS	TOTAL BUY	TOTAL PRODUCT	ACTIONS	rahnut rahnut@gmail.com	Done	Rp. 3000	2	⋮ ⚡	devon devonprasya@gmail.com	Processed	Rp. 3000	2	⋮ ⚡	<table border="1"> <thead> <tr> <th>CUSTOMER</th> <th>STATUS</th> <th>TOTAL BUY</th> <th>TOTAL PRODUCT</th> <th>ACTIONS</th> </tr> </thead> <tbody> <tr> <td>rahnut rahnut@gmail.com</td> <td>Done</td> <td>Rp. 3000</td> <td>2</td> <td>⋮ ⚡</td> </tr> <tr> <td>devon devonprasya@gmail.com</td> <td>Done</td> <td>Rp. 3000</td> <td>2</td> <td>⋮ ⚡</td> </tr> </tbody> </table>	CUSTOMER	STATUS	TOTAL BUY	TOTAL PRODUCT	ACTIONS	rahnut rahnut@gmail.com	Done	Rp. 3000	2	⋮ ⚡	devon devonprasya@gmail.com	Done	Rp. 3000	2	⋮ ⚡
CUSTOMER	STATUS	TOTAL BUY	TOTAL PRODUCT	ACTIONS																												
rahnut rahnut@gmail.com	Done	Rp. 3000	2	⋮ ⚡																												
devon devonprasya@gmail.com	Processed	Rp. 3000	2	⋮ ⚡																												
CUSTOMER	STATUS	TOTAL BUY	TOTAL PRODUCT	ACTIONS																												
rahnut rahnut@gmail.com	Done	Rp. 3000	2	⋮ ⚡																												
devon devonprasya@gmail.com	Done	Rp. 3000	2	⋮ ⚡																												

Pengujian *transaction* berhasil, setelah mengubah status pesanan menjadi selesai, status pesanan berhasil diperbarui

7. Pengujian *Customer*

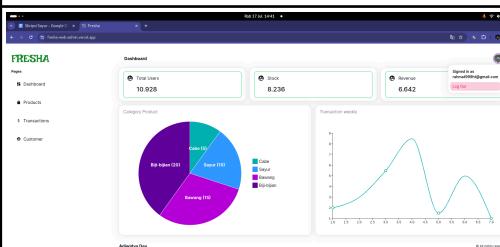
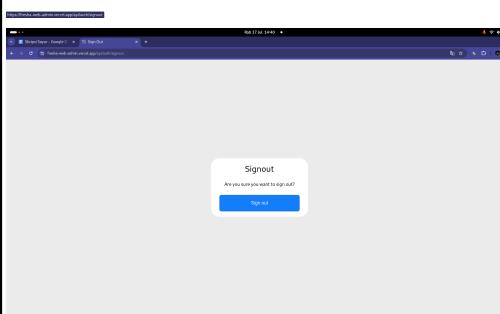
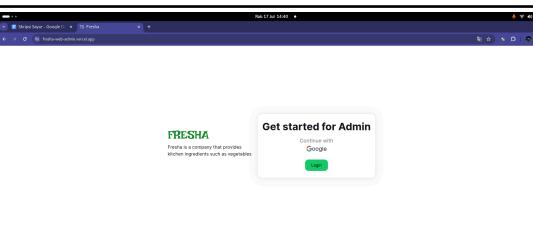
Tabel 4.15 Pengujian *Customer* Admin

Test Factor	Hasil	Keterangan
mengaktifkan akun/menonaktifkan akun	✓	Berhasil aktifkan/menonaktifkan akun
Screenshot		
Sebelum		Sesudah
 		
 		
 		

Pengujian *customer* admin berhasil, setelah mengaktifkan atau menonaktifkan akun, status akun berhasil diubah sesuai dengan perintah.

8. Pengujian Log-Out

Tabel 4.16 Pengujian Log-Out Admin

Test Factor	Hasil	Keterangan
Menekan tombol log-out	✓	Berhasil kembali ke halaman login
Screenshot		
Sebelum	Sesudah	
 		

Pengujian log-out admin berhasil, setelah menekan tombol log-out, pengguna berhasil diarahkan kembali ke halaman login.

Tabel 4.17 Pengujian Black Box Aplikasi Admin

No	Fitur Uji	Skenario Pengujian	Hasil yang diharapkan	Hasil Ujian
1	<i>Login</i>	Menekan tombol <i>login</i> akan menampilkan <i>pop-up google</i> dan memvalidasi apakah akun <i>google</i> tersebut terdaftar sebagai admin	Berhasil menampilkan <i>pop-up google</i> dan memvalidasi apakah akun <i>google</i> tersebut terdaftar sebagai admin	Berhasil masuk halaman utama

2	Tambah <i>Product</i>	Mengisi semua form dan menekan tombol <i>submit</i>	Tidak ada form yang boleh kosong	Berhasil menambah <i>product</i>
3	Edit <i>Product</i>	Mengisi semua form dan menekan tombol <i>submit</i>	Tidak ada form yang boleh kosong	Berhasil mengedit <i>product</i>
4	Identifikasi <i>image</i> dengan <i>machine learning</i>	Memilih <i>image</i> dan menekan tombol <i>identify</i>	Berhasil mengidentifikasi <i>image</i> dengan <i>machine learning</i>	Berhasil menampilkan hasil identifikasi
5	Mengubah status pesanan	Menekan ikon edit dan mengubah status pesanan	Berhasil mengubah status pesanan menjadi selesai	Berhasil mengubah status pesanan
6	Mengaktifkan akun/menonaktifkan <i>customer</i>	Menekan icon edit dan mengubah status <i>customer</i>	Berhasil mengubah status customer	Berhasil aktifkan/menonaktifkan <i>customer</i>
7	<i>Log-out</i>	Menekan tombol <i>log-out</i>	Berhasil menampilkan <i>alert log-out</i>	Berhasil kembali ke halaman <i>login</i>

Pengujian *black box* pada aplikasi admin berhasil dilakukan, menunjukkan bahwa semua fungsi utama berjalan sesuai dengan yang diharapkan

4.3.4. Aplikasi *Mobile*

Pengujian Sistem yang dilakukan dengan menggunakan metode pengujian langsung berdasarkan teknik *Black Box* dengan menguji fungsionalitas dari aplikasi, tombol dan kesesuaian hasil aplikasi, berdasarkan pada pembahasan bab-bab sebelumnya hasil penelitian aplikasi *mobile* sebagai berikut:

1. Pengujian *Register*

Tabel 4.18 Pengujian *Register Customer*

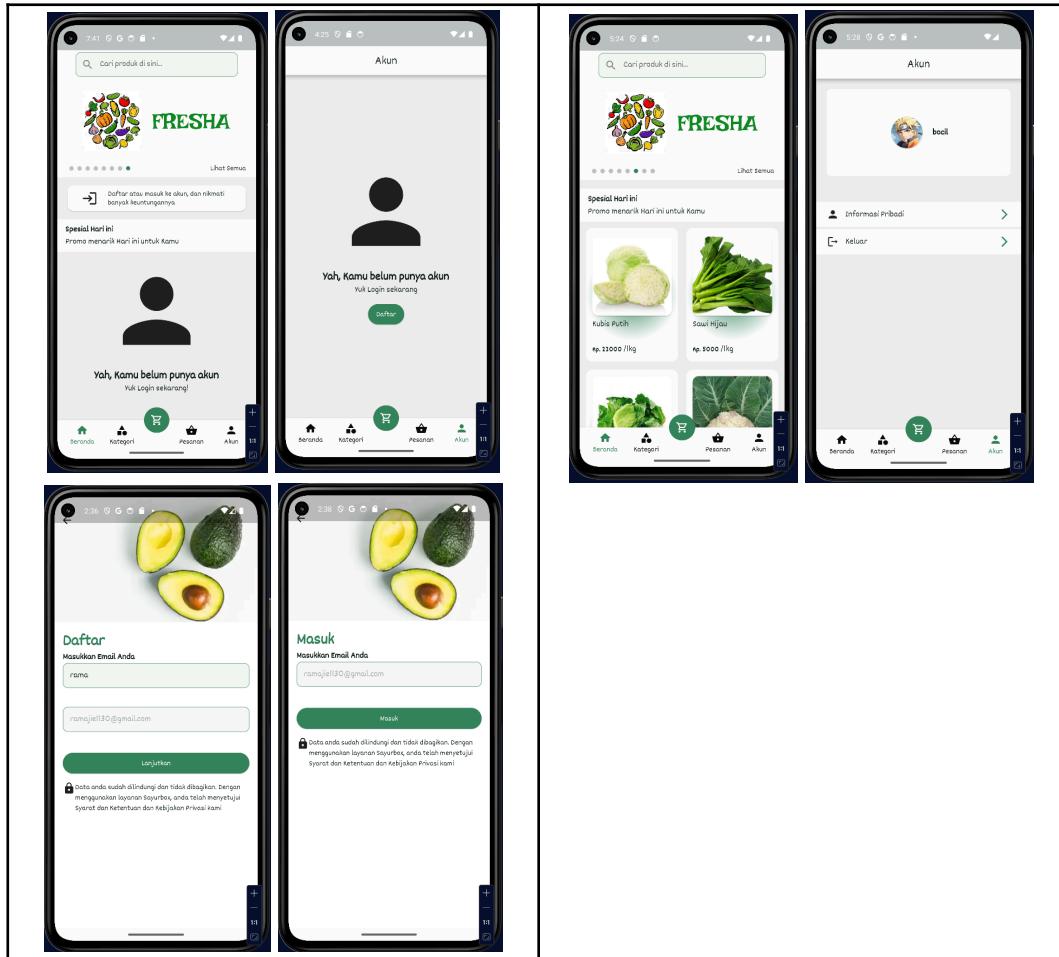
Test Factor	Hasil	Keterangan
Mengisi form nama dan <i>email google</i>	✓	Berhasil masuk ke halaman <i>login</i>
Screenshot		
Sebelum		Sesudah
		

Pengujian registrasi *customer* berhasil, setelah mengisi form dengan nama dan *email Google*, pengguna berhasil masuk ke halaman *login*.

2. Pengujian Login

Tabel 4.19 Pengujian Login Customer

Test Factor	Hasil	Keterangan
Memasukkan <i>email google</i> untuk <i>login</i>	✓	Berhasil masuk halaman utama dan menampilkan akun di halaman profil
Screenshot		
Sebelum		Sesudah

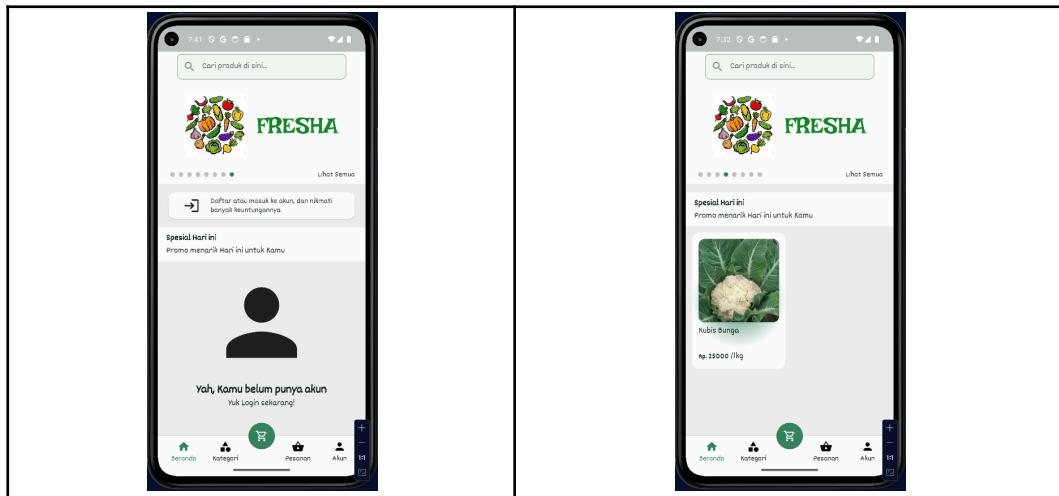


Pengujian *login customer* berhasil, setelah memasukkan *email Google* untuk *login*, pengguna berhasil masuk ke halaman utama dan akun ditampilkan di halaman profil.

3. Pengujian Rekomendasi

Tabel 4.20 Pengujian Rekomendasi *Customer*

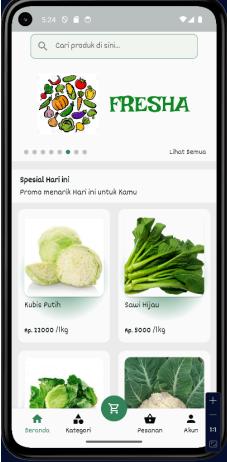
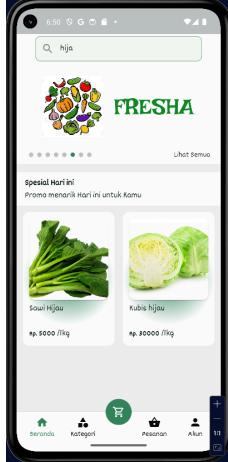
<i>Test Factor</i>	<i>Hasil</i>	<i>Keterangan</i>
Mengisi form <i>search</i>	✓	Berhasil melakukan filter data dengan <i>value</i> form <i>search</i> dan berpatokan pada nama sayur
<i>Screenshot</i>		
Sebelum		Sesudah



Pengujian rekomendasi *customer* berhasil, setelah mengisi form pencarian dan melakukan filter data berdasarkan nilai dari form pencarian dengan acuan nama sayur.

4. Pengujian *Search* Rekomendasi

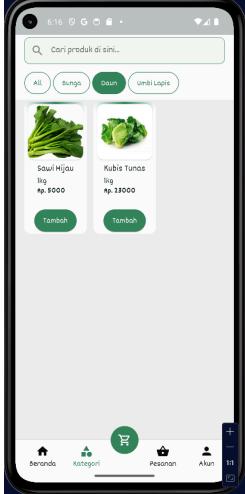
Tabel 4.21 Pengujian *Search* Rekomendasi *Customer*

<i>Test Factor</i>	<i>Hasil</i>	<i>Keterangan</i>
Mengisi form <i>search</i>	✓	Berhasil melakukan filter data dengan <i>Value</i> form <i>search</i> dan berpatokan pada nama sayur
<i>Screenshot</i>		
Sebelum		Sesudah
		

Pengujian *search* rekomendasi *customer* berhasil, setelah mengisi form pencarian dan melakukan filter data berdasarkan nilai dari form pencarian dengan acuan nama sayur.

5. Pengujian *filter* Kategori

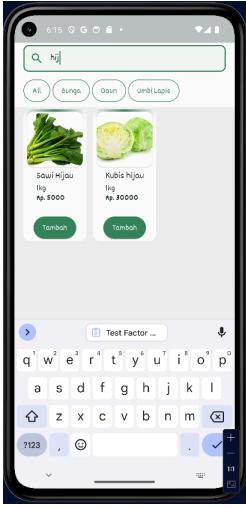
Tabel 4.22 Pengujian *Filter* Kategori *Customer*

Test Factor	Hasil	Keterangan
Menekan salah satu tombol kategori	✓	Berhasil melakukan filter data berdasarkan kategori
Screenshot		
Sebelum		Sesudah
		

Pengujian filter kategori *customer* berhasil, setelah menekan salah satu tombol kategori, sistem berhasil memfilter data berdasarkan kategori yang dipilih.

6. Pengujian *Search* Kategori

Tabel 4.23 Pengujian Search Kategori Customer

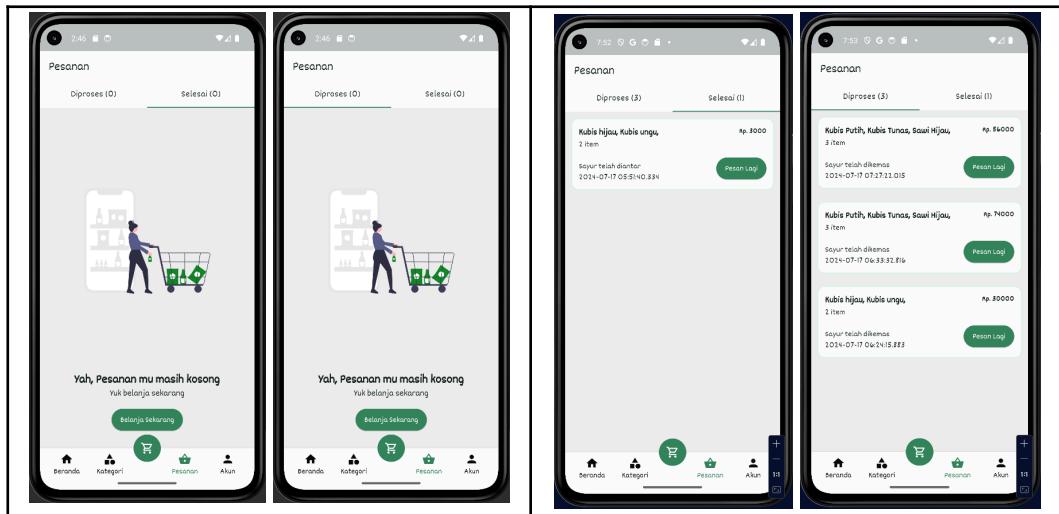
Test Factor	Hasil	Keterangan
Mengisi form search	✓	Berhasil melakukan filter data dengan <i>value</i> form search dan berpatokan pada nama sayur
Screenshot		
Sebelum	Sesudah	
		

Pengujian *search* kategori *customer* dilakukan dengan mengisi form pencarian dan berhasil memfilter data berdasarkan nilai dari form pencarian, dengan acuan pada nama sayur.

7. Pengujian Riwayat Pesanan

Tabel 4.24 Pengujian Riwayat Pesanan Customer

Test Factor	Hasil	Keterangan
Lakukan pesanan untuk mengisi riwayat pesanan	✓	Riwayat pesanan akan terisi setelah melakukan pesanan
Screenshot		
Sebelum	Sesudah	



Pengujian riwayat pesanan *customer* dilakukan dengan melakukan pesanan untuk mengisi riwayat pesanan, dan riwayat pesanan akan terupdate setelah pesanan berhasil dilakukan.

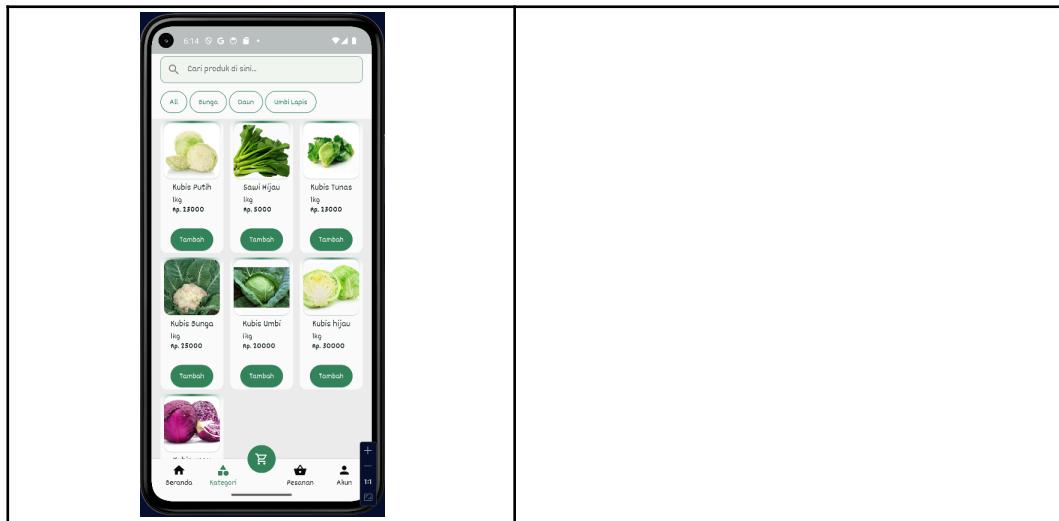
8. Pengujian Keranjang

Tabel 4.25 Pengujian Keranjang *Customer*

Test Factor	Hasil	Keterangan
Memilih salah satu sayur yang diinginkan	✓	Keranjang akan terisi ketika sudah memilih sayur

Screenshot

Sebelum	Sesudah



Pengujian keranjang *customer* dilakukan dengan memilih salah satu sayur yang diinginkan, dan keranjang akan otomatis terisi setelah sayur dipilih.

9. Pengujian *Checkout*

Tabel 4.26 Pengujian *Checkout Customer*

Test Factor	Hasil	Keterangan
Menekan tombol pesan	✓	Berhasil masuk halaman utama dan menampilkan <i>alert</i> berhasil melakukan pesanan
Screenshot		
Sebelum		Sesudah

Pengujian *checkout customer* dilakukan dengan menekan tombol pesan, berhasil masuk ke halaman utama, dan menampilkan alert yang mengonfirmasi pesanan telah berhasil dilakukan.

10. Pengujian *Update Profile*

Tabel 4.27 Pengujian *Update Profile Customer*

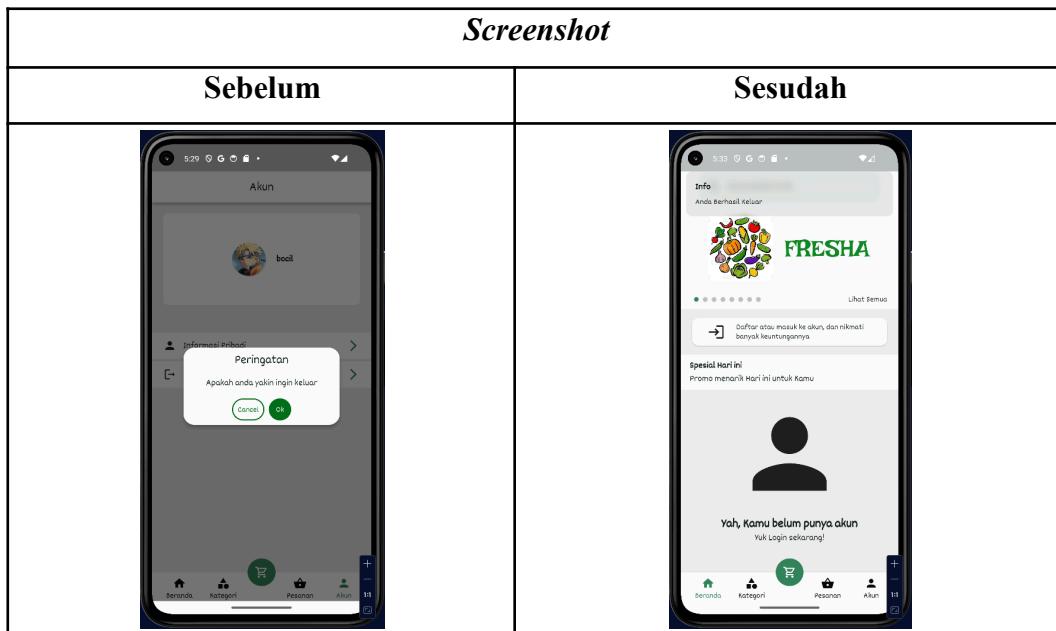
Test Factor	Hasil	Keterangan
Memasukkan <i>email google</i> untuk <i>login</i>	✓	Berhasil masuk halaman utama
Screenshot		
Sebelum		Sesudah

Pengujian *update profile customer* dilakukan dengan memasukkan email Google untuk login, dan berhasil masuk ke halaman utama.

11. Pengujian *Log-Out Customer*

Tabel 4.28 Pengujian *Log-Out Customer*

Test Factor	Hasil	Keterangan
Memasukkan <i>email google</i> untuk <i>login</i>	✓	Berhasil masuk halaman utama



Pengujian *Log-Out Customer* dilakukan dengan memasukkan *email Google* untuk *login*, kemudian berhasil masuk ke halaman utama.

Tabel 4. 29 Pengujian Black Box Aplikasi *Android*

No	Fitur Uji	Skenario Pengujian	Hasil yang diharapkan	Hasil Ujian
1	<i>Register</i>	Mengisi form nama dan <i>email google</i>	Berhasil memvalidasi form kosong dan menampilkan <i>pop-up google</i> dan memvalidasi apakah akun <i>google</i> tersebut terdaftar sebagai <i>customer</i>	Berhasil masuk ke halaman <i>login</i>
2	<i>Login</i>	mengisi form <i>email google</i> untuk <i>login</i>	Berhasil menampilkan <i>pop-up google</i> dan memvalidasi apakah akun <i>google</i> tersebut	Berhasil masuk halaman utama dan menampilkan akun di halaman <i>profile</i>

			terdaftar sebagai <i>customer</i>	
3	Rekomenda si sayur	Melakukan pemesanan terlebih dahulu	Dapat memfilter sayur berdasarkan riwayat pesanan <i>customer</i>	Berhasil memfilter sayur berdasarkan riwayat pesanan <i>customer</i>
4	<i>Search</i> rekomendasi	Mengisi form <i>search</i>	Menampilkan product sayur sesuai form <i>search</i> secara <i>realtime</i>	Berhasil melakukan filter produk
5	Filter kategori	Menekan salah satu tombol kategori	Menampilkan <i>product</i> sayur sesuai dengan kategori produk yang dipilih	Berhasil melakukan filter produk
6	<i>Search</i> Kategori	Mengisi form <i>search</i>	Menampilkan product sayur sesuai form <i>search</i> secara <i>realtime</i>	Berhasil melakukan filter produk
7	Riwayat Pesanan	Lakukan pesanan untuk mengisi riwayat pesanan	Berhasil membedakan antara pesanan yang sudah selesai dan masih di proses	Riwayat pesanan akan terisi setelah melakukan pesanan
8	Keranjang	Memilih salah satu sayur yang diinginkan	Berhasil menambah dan mengurangi jumlah sayur yang ingin dipesan	Keranjang akan terisi ketika sudah memilih sayur
9	<i>Checkout</i>	Menekan tombol pesan	Dapat memesan beberapa sayur yang diinginkan	Berhasil masuk halaman utama dan menampilkan <i>alert</i> berhasil melakukan pesanan

10	<i>Update Profile</i>	Mengisi ulang form info profil	Dapat memvalidasi form kosong dan memperbarui <i>database</i>	Berhasil mengubah info profil
11	<i>Log-out</i>	Menekan tombol <i>log-out</i>	Berhasil menampilkan <i>alert log-out</i>	Berhasil kembali ke halaman <i>login</i>

Pengujian *black box* pada aplikasi *mobile* berhasil dilakukan, menunjukkan bahwa semua fungsi utama berjalan sesuai dengan yang diharapkan.

BAB V

KESIMPULAN

5.1. Kesimpulan

Berdasarkan hasil penelitian dan pengembangan aplikasi *mobile android* yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

1. Perancangan dan Pengembangan Aplikasi *Mobile Android*:

Aplikasi yang dikembangkan berhasil memenuhi kebutuhan pengguna perangkat *mobile* dalam memesan sayur-sayuran secara *online*. Aplikasi ini dirancang dengan antarmuka yang *user-friendly* dan menyediakan fitur-fitur yang memudahkan pengguna dalam menjelajahi, memilih, dan memesan sayuran.

2. Pengenalan Pola pada Gambar Menggunakan *Machine Learning*:

Implementasi *machine learning* dalam aplikasi untuk mengenali pola pada gambar berhasil dilakukan. Aplikasi mampu mengenali dan mengklasifikasikan jenis-jenis sayuran berdasarkan gambar yang diunggah pengguna dengan tingkat akurasi yang memadai. Ini menunjukkan bahwa teknik *machine learning* yang digunakan efektif dalam pengenalan gambar.

3. Rekomendasi Sayur Menggunakan *Content-Based Filtering*:

Aplikasi berhasil menerapkan metode *Content-Based Filtering* untuk merekomendasikan sayur berdasarkan riwayat pemesanan pengguna. Metode ini mampu memberikan rekomendasi yang relevan dan sesuai dengan preferensi pengguna, sehingga meningkatkan kepuasan pengguna dalam menggunakan aplikasi.

5.2. Saran

1. Pengembangan Lebih Lanjut pada Fitur Pengenalan Gambar:

Meskipun aplikasi sudah mampu mengenali gambar sayuran, masih ada ruang untuk peningkatan akurasi dan kecepatan proses pengenalan. Diperlukan penelitian lebih lanjut dalam hal optimasi *algoritma machine learning* serta penggunaan *dataset* yang lebih besar dan lebih bervariasi untuk meningkatkan performa sistem.

2. Penambahan Fitur Kustomisasi Rekomendasi:

Untuk meningkatkan kepuasan pengguna, disarankan agar aplikasi dikembangkan lebih lanjut dengan menambahkan fitur kustomisasi pada sistem rekomendasi. Misalnya, pengguna bisa memberikan preferensi terhadap jenis sayuran tertentu atau memilih sayuran berdasarkan kandungan nutrisi yang diinginkan.

3. Pengembangan Versi *iOS*:

Untuk menjangkau lebih banyak pengguna, disarankan agar aplikasi ini juga dikembangkan dalam versi *iOS*. Dengan demikian, pengguna perangkat *Apple* juga dapat memanfaatkan layanan yang disediakan oleh aplikasi ini.

4. Edukasi dan Promosi di Masyarakat Baruga:

Masyarakat Baruga perlu diberikan edukasi tentang manfaat menggunakan aplikasi ini, baik dari segi kemudahan akses maupun kualitas sayuran yang ditawarkan. Promosi yang efektif juga perlu dilakukan untuk meningkatkan jumlah pengguna aplikasi ini di daerah tersebut.

DAFTAR PUSTAKA

- Affif Surya Diantika Yuki Firmanto, SE., MSA., CA., Ak., IMPLEMENTASI MACHINE LEARNING PADA APLIKASI PENJUALAN PRODUK DIGITAL (STUDI PADA GRABKIOS)
- Alkaff, M., Khatimi, H., & Eriadi, A. (2020). Sistem Rekomendasi Buku Menggunakan Weighted Tree Similarity dan Content Based Filtering. MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komputer, 20(1), 193-202. <https://doi.org/10.30812/matrik.v20i1.617>
- Amara, Swastika. A. (2019). “Pengembangan Aplikasi Media Pembelajaran IPA Biologi Berbasis Multimedia Interaktif Untuk Kelas VII di SMP Negeri 1 Ngargoyoso”. Metode Penelitian Pendidikan: R&D.
- Hanif F. I., & Sinambela G. M. (2020). PEMBUATAN APLIKASI E-TATIB BERBASIS ANDROID MENGGUNAKAN BAHASA PEMROGRAMAN DART. Jurnal Teknologi dan Terapan Bisnis. 2(3).<https://doi.org/10.0301/jttb.v3i2.79>
- Krisnada F. E., & Radius Tanoe. (2019). Aplikasi Penjualan Tiket Kelas Pelatihan Berbasis *Mobile* menggunakan *Flutter*. Jurnal Teknik Informatika dan Sistem Informasi. 5(3). <http://dx.doi.org/10.28932/justisi.v5i3.1865>
- Muhammad Fajriansyah, Putra Pandu Adikara, & Agus Wahyu Widodo. (2021) “Sistem Rekomendasi Film Menggunakan Content Based Filtering” Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 5, No. 6, Mei 2021, hlm. 2188-2199.
- Muslihudin, d. (2018). “Implementasi aplikasi rumah pintar berbasis android dengan arduino microcontroller”. Jurnal keteknikan dan sains, 1(1), 23–31.
- Prisky Ratna Aningtiyas, Agus Sumin, & Setia Wirawan. (2020) “Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra –Terlatih” Jurnal Ilmiah KOMPUTASI, Vol. 19, No. 3.
- Putri Nasiti. (2019). Penerapan Metode Content Based Filtering dalam Implementasi Sistem Rekomendasi Tanaman Pangan. Jurnal Teknika 8(1), 1-10.
- Reddy, S. R. S. et al. (2019) “Content-Based Movie Recommendation System Using Genre Correlation”, in Satapathy, S. C., Bhateja, V., and Das, S. (eds) Smart Intelligent Computing and Applications. Singapore: Springer Singapore, pp. 391–397.

Rhesa Havilah Mondi,Ardhi Wijayanto, Winarno. (2019) “RECOMMENDATION SYSTEM WITH CONTENT-BASED FILTERING METHOD FOR CULINARY TOURISM IN MANGAN APPLICATION”, Jurnal Ilmiah Teknologi dan Informasi Vol. 8, No. 2, December 2019.

Susanty, w., astari, i. N., & thamrin, t. (2019). “Aplikasi gis menggunakan metode location based service (lbs) berbasis android. Explore: jurnal sistem informasi dan telematika, 10(1). <Https://doi.org/10.36448/jsit.v10i1.1218>.

Sutjiadi, Raymond & Pattiasina, Timothi Jhon. (2020). Deteksi Objek Menggunakan Dashboard Camera Untuk Sistem Peringatan Pencegah Kecelakaan Pada Mobil. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), Vol. 7, No. 2, April 2020, hlm. 427-434.

LAMPIRAN

1. Implementasi machine learning pada aplikasi web

```
const [identifyLoading, setIdentifyLoading] = useState(false);

const [isModelLoading, setModelLoading] = useState(false);
const [model, setModel] = useState<mobilenet.MobileNet | null>(null);
const [results, setResults] = useState<
{
  className: string;
  probability: number;
}[]
>([]);

const imageRef = useRef<HTMLImageElement | null>(null);

const loadModel = async () => {
  setModelLoading(true);
  try {
    const initModel = await mobilenet.load();
    setModel(initModel);
    setModelLoading(false);
  } catch (e) {
    console.log(e);
    setModelLoading(false);
  }
};

const handleSubmit = async () => {
  setUploading(true);
  if (selectedFile) {
    const imageUrl =
      "https://omhmokdygpqbhwdtshvk.supabase.co/storage/v1/object/public/images/sayur/";

    const imageName = Date.now().toString() + selectedFile.name;
    const { data, error } = await supabase.storage
      .from("images")
      .upload("sayur/" + imageName, selectedFile);

    createNewProduct.mutate({
      name: name,
      image: imageUrl + imageName,
      category: category,
    });
  }
};
```

```
        hastag_ml: hastag,
        desc: description,
        price: price,
        stock: stock,
    });
    setUploading(false);
    console.log("berhasil guys");

    if (data) {
        console.log(data);
    } else if (error) {
        console.log(error);
    }
}
};

{results.length > 0 ? (
    <div className="imageResult">
        {results.map((result, index) => {
            return (
                <div className={styles.borderCard} key={result.className}>
                    <span className={styles.name}>{result.className}</span>
                    <div className={styles.confidence}>
                        <span className={styles.confidenceText}>
                            Confidence level:{" "}
                            {(result.probability * 100).toFixed(2)}%
                        </span>
                        {index === 0 && (
                            <span className={styles.bestGuess}>Best Guess</span>
                        )}
                    </div>
                </div>
            );
        })}
    </div>
) : (
    <div className="text-center font-medium italic text-gray-500">
        kosong
    </div>
)}
```

2. Implementasi Algoritma Content Base Filtering Pada Beck-end

```
productRouter.get("/:hashtags", async (c) => {
  const { hashtags } = c.req.param();
  const hashtagList = hashtags.split(","); // Memisahkan string menjadi array
  const productsRekomen = await
  productService.getsRekomenProduct(hashtagList);
  return c.json({
    code: HttpStatus.OK,
    status: "Response hastag_ml Ok",
    data: productsRekomen,
  });
});

type TFIDF = Record<string, number>; // Changed from index signature to
Record
type ProductWithTFIDF = Product & { tfidf: TFIDF };

// Fungsi untuk menghitung TF
function calculateTF(term: string, document: string[]): number {
  const termFrequency = document.filter(word => word === term).length;
  return termFrequency > 0 ? 1 + Math.log(termFrequency) : 0;
}

// Fungsi untuk menghitung IDF
function calculateIDF(term: string, allDocuments: string[][]): number {
  const docCount = allDocuments.length;
  const docsWithTerm = allDocuments.filter(doc => doc.includes(term)).length;
  return Math.log((docCount + 1) / (1 + docsWithTerm));
}

// Fungsi untuk menghitung TF-IDF
function calculateTFIDF(
  terms: string[],
  document: string[],
  allDocuments: string[][],
): TFIDF {
  const tfidf: TFIDF = {};
  terms.forEach((term) => {
    const tf = calculateTF(term, document);
    const idf = calculateIDF(term, allDocuments);
    tfidf[term] = tf * idf;
  });
}
```

```

        return tfidf;
    }

    // Fungsi untuk menghitung Cosine Similarity
    function cosineSimilarity(vecA: TFIDF, vecB: TFIDF): number {
        const intersection = Object.keys(vecA).filter((term) => term in vecB);
        const dotProduct = intersection.reduce(
            (sum, term) => sum + (vecA[term] ?? 0) * (vecB[term] ?? 0),
            0,
        );

        const magnitudeA = Math.sqrt(
            Object.values(vecA).reduce((sum, val) => sum + val * val, 0),
        );
        const magnitudeB = Math.sqrt(
            Object.values(vecB).reduce((sum, val) => sum + val * val, 0),
        );
        return dotProduct / (magnitudeA * magnitudeB);
    }

    export async function getsProductRekomen(
        list_hastag: string[],
    ): Promise<Product[]> {
        // Mengambil produk dari database berdasarkan hashtag yang diberikan pengguna
        const products = await db.product.findMany({
            where: { hastag_ml: { in: list_hastag } },
            orderBy: { createdAt: "desc" },
        });

        // Memisahkan deskripsi produk menjadi kata-kata
        const productDocuments = products.map((product) => product.hastag_ml.split(""));

        // Menggabungkan semua kata yang ada dalam deskripsi produk menjadi satu set unik
        const allTerms = Array.from(new Set(productDocuments.flat()));

        // Menghitung nilai TF-IDF untuk setiap produk
        const productTFIDFs: ProductWithTFIDF[] = products.map((product) => ({
            ...product,
            tfidf: calculateTFIDF(allTerms, product.hastag_ml.split(" "), productDocuments),
        }));

        // Menghitung nilai TF-IDF untuk hashtag pengguna

```

```

const penggunaTFIDF = calculateTFIDF(allTerms, list_hashtag,
productDocuments);

// Memfilter produk yang memiliki nilai TF-IDF tidak null
const filteredProductTFIDFs = productTFIDFs.filter((product) => {
  return Object.values(product.tfidf).some((value) => value !== null);
});

// Mengurutkan produk berdasarkan kesamaan cosine tertinggi
filteredProductTFIDFs.sort((a, b) => {
  const simA = cosineSimilarity(a.tfidf, penggunaTFIDF);
  const simB = cosineSimilarity(b.tfidf, penggunaTFIDF);
  return simB - simA; // Urutkan berdasarkan kesamaan tertinggi
});

return filteredProductTFIDFs.map((product) => product);
}

```

3. Implementasi Algoritma Content Base Filtering Pada Android

```

Future findAllHastagMI() async {
prefService.prefInit();
if (prefService.getIdCustomer != null) {
  hastagMIProvider
    .getHastagMIWhereIdCustomer(prefService.getIdCustomer.toString())
    .then((result) {
  if (result.code == 200) {
    // Hitung kemunculan setiap hastag
    for (var e in result.data) {
      if (hastagCounts.containsKey(e.name)) {
        hastagCounts[e.name] = hastagCounts[e.name]! + 1;
      } else {
        hastagCounts[e.name] = 1;
      }
    }
    // Buat daftar ModelValueHastagMI
    listHastagMI = hastagCounts.entries
      .map((entry) => ModelValueHastagMI(
        title: entry.key,
        value: entry.value,
      ))
      .toList();
    log(listHastagMI.toString(), name: 'data listHastagMI');
  } else {
    log('data listHastagMI kosong', name: 'kosong');
  }
}

```

```
        }
    }, onError: (err) {
        log('error listHastagMI: $err', name: 'error');
    });
} else {
    log('data getIdCustomer kosong', name: 'getIdCustomer');
}
}

Future findAllProduct() async {
productProvider
    .fetchProductsWhereHastag(
        hastag1: listHastagMI[0].title, hastag2: listHastagMI[0].title)
    .then((result) {
    if (result.code == 200) {
        listProduct = result.data;
        log(listProduct.length.toString(), name: 'data model');
        change(listProduct, status: RxStatus.success());
    } else {
        log('kosong', name: 'data kosong');
        change([], status: RxStatus.empty());
    }
}, onError: (err) {
    change(null, status: RxStatus.error(err.toString()));
});
}
```

RIWAYAT HIDUP



Rahmat Hidayat, lahir di Jeneponto pada 5 Februari 1999, putra dari pasangan Makka dan Nursiah. penulis merupakan anak ke 2 dari 2 bersaudara. Penulis menempuh pendidikan di sekolah dasar SD 50 Negeri Taba mulai tahun 2004 dan selesai pada tahun 2010. Kemudian pada tahun yang sama penulis melanjutkan pendidikan ke sekolah menengah pertama di MTs Negeri Binamu Jeneponto dan selesai pada tahun 2013. Selanjutnya penulis melanjutkan pendidikan ke sekolah menengah atas di MA Negeri Binamu Jeneponto dan selesai pada tahun 2016. Di Tahun 2019 penulis melanjutkan pendidikan di Universitas Dipa Makassar sebagai tingkat akhir jurusan Teknik Informatika. Semasa kuliah penulis aktif di kegiatan organisasi salah satunya menjadi pengurus bidang program periode 2021/2022 KeDai Computerworks, serta menjadi koordinator keilmuan kemudian naik jabatan menjadi wakil ketua 2 periode 2022/2023 KeDai Computerworks.



Muhammad Fahrezy Al Dzuhri, lahir di Makassar pada 18 Maret 2001, putra dari pasangan Zulfitri, SE., M.A.P dan Asry Samad, SE., M.Si. Penulis merupakan anak ke 1 dari 2 bersaudara. Penulis menempuh pendidikan di sekolah dasar SD Athahiriyyah Yapis Wamena mulai tahun 2007 dan selesai pada tahun 2013. Kemudian pada tahun yang sama penulis melanjutkan pendidikan ke sekolah menengah pertama di SMP Negeri 2 Wamena dan selesai pada tahun 2016. Selanjutnya penulis melanjutkan pendidikan ke sekolah menengah atas di SMAS Islam Athirah Bukit Baruga dan selesai pada tahun 2019. Ditahun 2019 penulis melanjutkan pendidikan di Universitas Dipa Makassar sebagai tingkat akhir jurusan Teknik Informatika. Semasa kuliah penulis aktif dikegiatan organisasi UKM KeDai *Computerworks* dan organisasi kedaerahan IMPAS DIPA MAKASSAR. Penulis memegang jabatan di KeDai *Computerworks* yaitu, staf *multimedia* KeDai *Computerworks* periode 2021/2022 dan Kordinator Keorganisasian KeDai *Computerworks* periode 2022/2023. Penulis memegang jabatan di IMPAS DIPA MAKASSAR yaitu, wakil ketua divisi internal IMPAS DIPA MAKASSAR periode 2020/2021, Ketua Umum IMPAS DIPA MAKASSAR periode 2021/2022, dan Kordinator Dewan Pertimbangan Organisasi IMPAS DIPA MAKASSAR periode 2023/2024.