BAB I

PENDAHULUAN

A. Latar Belakang

Cumi-cumi merupakan salah satu hasil perikanan penting di dunia. Di Indonesia kelompok hewan cumi ini mempunyai urutan ketiga produksi di dalam dunia perikanan setelah ikan dan udang. Namun sampai saat ini, perikanan cumi masih sangat jauh dari yang diharapkan dalam mengisi ekspor non migas, salah satu sebabnya adalah produksi cumi-cumi hingga kini masih tergantung dari hasil tangkapan di alam.Informasi beberapa aspek yang berhubungan dengan hewan cumi-cumi masih sangat jarang.

Di Indonesia tidak semua jenis cumi-cumi disukai oleh masyarakat untuk di konsumsi segar karena mempunyai daging yang sangat tebal. Oleh karena itu perlu pengolahan yang menjadikan produk ini lebih menarik. Cumi olahan merupakan salah satu alternatif yang dapat dibuat dalam pengembangan produk makanan berbahan baku cumi-cumi. Cumi-cumi yang telah mengalami perebusan, pengeringan dan dilanjutkan dengan cara menggoreng agar cumi-cumi tersebut memiliki penampakan yang menarik dan aromanya yang khas.

Namun, permasalahan yang muncul adalah masih banyak masyarakat yang belum mengetahui cara membedakan cumi-cumi yang segar dengan yang tidak segar. Apalagi jika di suatau wilayah tersebut tidak ada laut atau sungai, sehingga untuk mendapatakan cumi harus dikirim dari daerah lain. Hal tersebut dapat membuat cumi menjadi tidak segar karena memakan waktu pengiriman yang

cukup lama bahkan bisa berhari-hari.

Dengan kemajuan terbaru dalam teknologi pemrosesan gambar dan algoritma pembelajaran mesin telah menjanjikan dalam meningkatkan efisiensi dan akurasi deteksi kualitas kesegaran cumi-sumi. Oleh karena itu, dirancanglah sebuah aplikasi untuk mendeteksi tingkat kesegaran cumi-cumi, yaitu aplikasi untuk mendeteksi tingkat kesegaran cumi-cumi ini menggunakan pengolahan citra digital. Pada penelitian ini objek yang dideteksi adalah warna kulit dengan menggunakan metode *K-Nearest Neigbours (KNN)* untuk dapat mengelompokkan dengan 3 kondisi yaitu segar, tidak segar, dan busuk..

Aplikasi pengolahan citra pendeteksi kesegaran cumi-cumi merupakan suatu teknologi yang melibatkan analisis dan manipulasi cira digital mengunakan algoritma matematika yang digunakan untuk menentukan kualitas kesegaran cumi-cumi itu sendiri yang dilihat dari warna kulitnya dengan menggunakan sebuah citra gambar warna kulit. Dalam konteks pendeteksian kulalitas cumi-cumi, pengolahan citra digunakan untuk mengkstrak fitur yang relevan dari citra cumi-cumi dan mengklasifikasiannya ke dalam kategori kualitas yang berbeda. Teknologi ini bisa digunakan masyarakat luas yang ingin mengetahui kualitas kesegaran cumi-cumi yang akan dibelinya atau yang akan diolah sebelum dikonsumsi demi mendapatkan nutrisi yang baik untuk jaringan tubuh.

Berdasarkan latar belakang tersebut yaitu dengan membuat suatu aplikasi pendeteksi kesegaran cumi-cumi dalam rangka mempermudah pengguna untuk mengidentifikasi tingkat kesegaran cumi-cumi. Pada aplikasi ini mengunakan metode ruang warna *RGB (Red, Green, Blue)* untuk mendapatkan ciri pada citra warna kulit, sedangkan untuk meningkatkan akurasi dengan menggunakan metode

KNN *(K-Nearest Neighbours).* Metode KNN merupakan salah satu metode klasifikasi yang memiliki keunggulan yaitu bekerja berdasarkan jarak terpendek dari *query instance* ke training sample untuk menentukan KNN-nya. Training sample diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi mempresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi *training sample.* Dakat atau jauhnya tetangga biasanya dihitung berdasarkan *Euclidean Distance.* Jarak *Euclidean* paling sering digunakan menghitung suatu jarak. Dimana jarak *euclidean* memiliki fungsi untuk menguji ukuran yang bisa digunakan sebagai interpretasi kedekatan jarak antara dua objek yang akan diuji. Dengan metode KNN ini akurasi yang dihasilkan akan lebih besar dibanding dengan metode klasifikasi pada penelitian.

B. Rumusan masalah

Berdasarkan uraian latar belakang diatas, adapun rumusan masalah yang akan dibahas adalah "Bagaimana membuat Aplikasi Pendeteksi Kesegaran Cumi-cumi Berbasis Pengolahan Citra Dengan Metode *K-Nearest Neighbour* (KNN)?

C. Batasan masalah

Untuk dapat menghasilkan penelitian yang lengkap, akurat dan tepat maka diperlukan adanya ruang lingkup dan batasan masalah, oleh karenanya judul yang diangkat penulis membatasi beberapa masalah.

1. Aplikasi ini hanya dapat mendeteksi kesegaran di cumi-cumi melalui warna kulitnya.
2. Dalam penelitian ini kelayakan cumi-cumi hanya dikategorikan dalam 3 kelompok yaitu cumi-cumi segar, tidak segar, dan busuk

D. Tujuan Penelitian

Tujuan dari penelitian yang dilakukan adalah untuk mengembangkan sebuah sistem yang dapat secara otomatis mendeteksi tingkat kesegaran cumi-cumi berdasarkan data yang diberikan. Metode KNN digunakan untuk melakukan klasifikasi berdasarkan kemiripan data dengan tetangga terdekatnya. Dengan menggunakan metode ini, diharapkan aplikasi ini dapat memberikan prediksi yang akurat mengenai kesegaran cumi-cumi berdasarkan warna kulitnya. Hal ini dapat membantu pengguna, seperti pedagang atau konsumen, dalam mengambil keputusan yang tepat terkait pembelian atau pemrosesan cumi-cumi

E. Manfaat Penelitian

Manfaat dari penelitian ini sebagai berikut:

1. Bagi penulis

   Melatih penulis dalam mengembangkan dan menambah ilmu khususnya pada Algoritma *k-Nearest Neighbourd.*

2. Manfaat untuk masyarakat umum

   Adapun manfaat bagi masyarakat adalah dapat dijadikan alat sebagai pendeteksian kesegaran cumi-cumi dengan aplikasi

3. Manfaat untuk universitas

   Adapun manfaat bagi universitas adalah dapat dijadikan refrensi bagi penelitian selanjut.

F. Sistematika Penulisan

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka akan di uraikan beberapa tahapan dari penulisan secara

sistematis, yaitu :

BAB I : PENDAHULUAN

Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan laporan tugas akhir.

BAB II : TINJAUAN PUSTAKA

Bab ini berisi teori-teori tentang hal-hal yang digunakan dalam Perancangan Aplikasi Pendeteksi Kesegaran Cumi-cumi Berbasisi Pengolahan Citra Dengan Metode *K-Nearest Neigbour* (KNN)?

BAB III : METODE PENELITIAN

Bab ini membahas tentang waktu dan tempat penelitian, metode penelitian serta alat dan bahan yang digunakan dalam penelitian

BAB IV : PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi tentang percancangan sistem yang berjalan, analisis sistem yang di usulkan, rancangan sistem, rancangan *input* dan rancangan *output* serta pengujian sistem.

BAB V : PENUTUP

Bab ini berisi tentang kesimpulan yang dihasilkan dari pembahasan tentang Aplikasi Pendeteksi Kesegaran Cumi-cumi Berbasisi Pengolahan Citra Dengan Metode *K-Nearest Neigbour* (KNN) untuk memudahkan dalam menentukn ikan yang segar , dan beberapa saran sebagai hasil dari penelitian yang telah dilakukan.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Aplikasi

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau yang diharapkan maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya. Aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *User*.

Aplikasi menurut sumber yang dikutip dati Kamus Besar Bahasa Indonesia aplikasi



Gambar 2. 1 *Apps Icon*

adalah penerapan dari perancangan sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan aatau melaksanakan tugas khusus dari pengguna.

Aplikasi Komputer, Laptop, atau *Smartphone* Merupakan perangkat lunak atau program yang dirancang untuk menjalankan fungsi tertentu. Kata aplikasi sendiri berasal dari bahasa inggris yaitu *Application* yang berarti pengguna atau penerapan.(Sulaiman., dkk)

Aplikasi berasal dari kata *application* yang artinta penerapan penggunaan secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju.(Aziz, 2020)

2. Pendeteksi

Pendeteksi memiliki arti dalam kelas nomina atau kata benda sehingga pendeteksi dapat menyatakan nama dari seseorang, tempat, atau semua benda dan segala yang dibendakan. Pendeteksi berasal dari kata dasar deteksi. Menurut Kamus Besar Bahasa Indonesia (KBBI), arti kata pendeteksi adalah alat untuk mendeteksi. Pendeteksi berasal dari kata dasar deteksi.

Pendeteksi (dalam konteks teknologi dan ilmu komputer) adalah suatu sistem, alat, atau perangkat lunak yang dirancang untuk mengidentifikasi, mengenali, atau menemukan keberadaan atau karakteristik tertentu dalam suatu data atau lingkungan. Pendeteksi digunakan untuk mengenali objek, pola, atau kejadian tertentu yang mungkin sulit atau bahkan tidak mungkin dikenali secara manual.

Pendeteksi biasanya berdasarkan pada teknik-teknik dari bidang pengolahan citra, pembelajaran mesin, pengenalan pola, analisis statistik, dan lainnya. Penggunaan pendeteksi dapat sangat beragam, dari keperluan keamanan, analisis data, pengenalan pola, hingga aplikasi dalam kehidupan sehari-hari seperti filter gambar atau pencarian teks.

3. Kesegaran

Kesegaran adalah kondisi atau sifat dari sesuatu yang segar atau baru. Istilah ini biasanya digunakan untuk menggambarkan makanan, bahan makanan, atau barang-barang lain yang memiliki kualitas atau keadaan yang baik, tidak rusak, dan tidak mengalami pembusukan atau perubahan yang merugikan.

Menurut suratman (1975), suatu aspek fisik dari adanya keseluruhan yang menjadikan seseorang untuk dapat melakukan hidup produktif dan menysesuaikan pada tiap pembebanan atau stres fisik yang layak.

Kesegaran juga dapat berkaitan dengan kualitas dan daya tahan suatu barang atau produk. Dalam beberapa kasus, kesegaran menjadi penting untuk menjaga keamanan dan kualitas produk, seperti dalam industri makanan atau farmasi. Selain itu, kesegaran juga merupakan faktor penting dalam memilih makanan sehat dan menghindari konsumsi makanan yang telah basi atau tidak layak dikonsumsi.

4. Cumi-cumi

Cumi-cumi adalah hewan laut yang termasuk dalam kelas Cephalopoda, yang juga mencakup gurita, sotong, dan nautilus. Cumi-cumi memiliki tubuh yang lunak, berbentuk tabung dengan sirip di bagian ujung tubuhnya, serta delapan lengan pendek dan dua tentakel panjang yang dilengkapi pengisap untuk menangkap mangsa. Sebagai hewan invertebrata, cumi-cumi tidak memiliki tulang punggung, namun beberapa jenis memiliki cangkang internal yang disebut gladius atau "pena", yang berfungsi sebagai penopang tubuh. Cumi-cumi dikenal memiliki kemampuan untuk menyemprotkan tinta sebagai mekanisme pertahanan dari predator, sekaligus memiliki kemampuan kamuflase untuk beradaptasi dengan lingkungannya.

Cumi-cumi hidup di berbagai habitat laut, mulai dari perairan dangkal dekat pantai hingga laut dalam, dan dapat ditemukan di seluruh dunia. Mereka merupakan predator

aktif yang memangsa ikan kecil, krustasea, dan plankton. Secara komersial, cumi-cumi menjadi salah satu sumber makanan laut yang populer dan dikonsumsi dalam berbagai bentuk masakan di seluruh dunia. Keberadaan dan peran cumi-cumi dalam ekosistem laut sangat penting karena berkontribusi pada rantai makanan sebagai mangsa bagi ikan besar dan mamalia laut, serta sebagai pemangsa bagi spesies laut lainnya.



Gambar 2. 2 Cumi-cumi

Terdapat beberapa jenis cumi-cumi yang sering dikenal, dan setiap jenisnya memiliki karakteristik serta habitat yang berbeda. Berikut beberapa jenis cumi yang umum:

a. Cumi-Cumi Loligo (*Loligo spp.)*

   Ini adalah jenis cumi yang paling umum dikonsumsi. Memiliki tubuh panjang dengan sirip yang berbentuk seperti panah. Cumi jenis ini sering ditemukan di pasar-pasar dan restoran.

b. Cumi-Cumi Sepioteuthis (*Sepioteuthis spp.)*

   Juga dikenal sebagai "*squid cuttlefish*," memiliki sirip yang lebih lebar hampir sepanjang tubuhnya. Cumi ini memiliki daging yang lebih tebal dan sering digunakan dalam hidangan laut.

c. Cumi-Cumi Humboldt (*Dosidicus gigas*)

ni adalah cumi berukuran besar yang hidup di Samudera Pasifik. Dikenal dengan sifat agresifnya dan sering dijuluki "cumi jumbo" atau "cumi raksasa".

d. Cumi-Cumi Neon (*Watasenia scintillans*)

Cumi-cumi kecil yang dikenal sebagai "*firefly squid*" atau cumi neon karena memiliki kemampuan bioluminesensi. Biasanya ditemukan di perairan Jepang.

e. Cumi-Cumi Pasifik Raksasa (*Thysanoteuthis rhombus*)

Merupakan cumi-cumi besar yang hidup di Samudera Pasifik. Dapat tumbuh hingga ukuran yang sangat besar dan sering terlihat di laut dalam.

f. Cumi-Cumi Terbang (*Ommastrephidae spp.)*

Cumi jenis ini dapat meluncur di atas permukaan air, tampak seperti "terbang" untuk menghindari pemangsa. Cumi ini sering ditemukan di Samudera Atlantik dan Pasifik.

g. Cumi-Cumi Karang (*Sepioteuthis lessoniana*)

Juga disebut *"bigfin reef squid,"* hidup di perairan dangkal dekat terumbu karang. Cumi ini berukuran sedang dengan tubuh berwarna yang berubah-ubah.

Namun dalam penelitian kali ini, jenis cumi yang saya gunakan sebagai objek dalam aplikasi pendeteksi adalah Cumi-Cumi Loligo (*Loligo spp.),* yang dikenal luas di pasar dan perairan lokal. Jenis cumi ini dipilih karena karakteristiknya memungkinkan pendeteksian tingkat kesegarannya, yang menjadi aspek penting dalam pengujian kualitas hasil tangkapan laut. Dengan bentuk tubuh yang konsisten dan warna yang mudah diamati, Cumi-Cumi Loligo dapat dianalisis untuk mengidentifikasi perubahan fisik yang menjadi indikator kesegaran, seperti tekstur, warna kulit, dan elastisitas tubuh. Hal ini membuat Cumi-Cumi Loligo sangat cocok dijadikan sampel utama dalam aplikasi pendeteksi kesegaran yang dikembangkan dalam penelitian ini.

5. Pengolahan Citra Digital

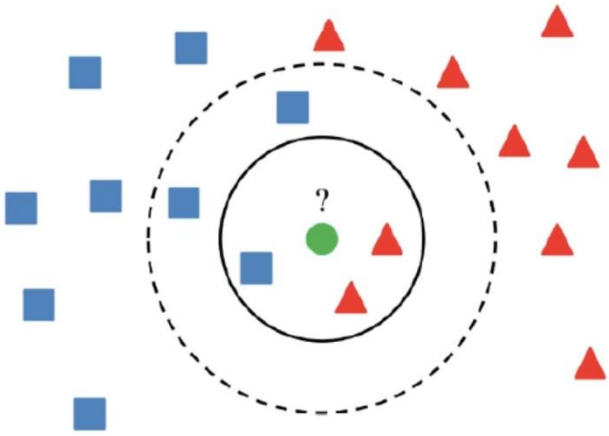
Pengolahan citra digital yaitu bidang ilmu yang mempelajari mengenai bagaimana suatu citra itu dibentuk, diolah dan dianalisis sehingga dapat menghasilkan sebuah informasi yang bisa dipahami oleh manusia. Dapat pula dijelaskan bahwa pengolahan citra digital adalah manipulasi dan interprestasi digital dari citra dengan bantuan komputer.

bentuk sinyal Berdasarkan pembuatannya, citra dapat digolongkan menjadi dua jenis yaitu citra simple dan citra computerized. Citra sederhana adalah citra yang dibentuk dari sinyal sederhana yang bersifat kontinyu, sedangkan citra terkomputerisasi adalah citra yang dibentuk dari sinyal canggih yang bersifat diskrit.

a. Menurut Jian (1989) Bahwa istilah pengolahan citra digital adalah pemrosesan gambar dua dimensi melalui komputer digital.
b. Menurut Efford (2000) Pengolahan citra digital merupakan istilah yang umum untuk berbagai teknik yang keberadaannya adalah untuk memanipulasi dan memodifikasi citra dengan menggunakan berbagai cara.
c. Menurut Suhandy (2003) Pengolahan citra digital adalah sebuah teknologi visual yang dipakai untuk mengamati dan menganalisis sebuah objek tanpa berhubungan secara langsung dengan objek yang diamati itu. Teknologi ini bisa dipakai untuk mengevaluasi mutu suatu produk tanpa merusak produk itu sendiri.

6. K-Nearest Neighbors (KNN)

Algoritma K-Nearest Neighbor (k-NN) adalah salah satu algoritma klasifikasi paling sederhana berdasarkan pembelajaran data. Algoritma ini termasuk dalam supervised learning. Idenya adalah untuk mencari kecocokan terdekat (closest match) dari data uji dalam ruang fitur. Untuk lebih jelasnya dapat dilihat pada Gambar dibawah.



Gambar 2. 3 Visualisasi tentang Klasifikasi menggunakan KNN

Gambar diatas, terdapat sejumlah data dengan dua kelas, yaitu kelas dengan kotak biru dan kelas dengan segitiga merah. Data-data tersebut diproyeksikan dalam ruang fitur. Misalnya, dapat dilihat pada contoh Gambar diatas, data-data diproyeksikan pada ruang koordinat 2D, yang artinya untuk setiap data memiliki dua fitur, yaitu koordinat x dan y. Jika ditambah jumlah fitur, misalnya dengan tiga fitur, maka diperlukan ruang 3D. Dan jika menggunakan sebanyak N fitur, maka membutuhkan ruang N-D (N-Dimensi), ruang N-Dimensi ini adalah ruang fiturnya. Pada Gambar diatas terdapat anggota baru atau data baru yang ditampilkan dengan lingkaran hijau. Data tersebut akan ditambahkan ke salah satu kelas biru atau kelas merah. Proses tersebut dinamakan klasifikasi. Lalu, bagaimana cara untuk mengklasifikasikan data tersebut? Salah satu metode adalah dengan cara memeriksa siapa tetangga terdekatnya. Dari gambar juga terlihat jelas bahwa data baru tersebut lebih dekat pada data dengan kelas segitiga merah. Maka data baru tersebut dapat diklasifikasikan untuk masuk ke kelas segitiga merah. Metode ini disebut Nearest Neighbor, karena klasifikasi hanya bergantung pada tetangga terdekat.

Data dengan kelas segitiga merah mungkin yang terdekat tetapi bagaimana jika ada banyak data dengan kelas kotak biru yang dekat dengannya. Jadi mengecek yang

terdekat saja tidak cukup namun dapat dilakukan pengecekan untuk beberapa k tetangga terdekat.

Dalam Gambar k=3,yaitu 3 tetangga terdekat. Data baru tersebut memiliki dua data pada kelas merah dan satu pada kelas biru. Jadi dengan k =3,data baru tersebut diklasifikasikan untuk masuk ke kelas segitiga merah. Tetapi bagaimana jika kita menggunakan k = 7, terdapat 5 tetangga terdekat di kelas biru dan 2 tetangga terdekat di kelas merah. Jadi dengan k = 7 data baru tersebut diklasifikasikan untuk masuk ke kelas kotak biru. Maka data dilihat hasil klasifikasi dapat berubah dengan perubahan nilai k. Metode ini disebut k-Nearest Neighbor.

Penting untuk mencatat bahwa algoritma KNN adalah algoritma "lazy learning," yang berarti algoritma ini tidak melakukan proses pembelajaran (training) pada tahap awal. Sebaliknya, algoritma KNN menyimpan seluruh dataset pelatihan dan menghitung jarak ketika ada data uji yang ingin diprediksi. Karena alasan ini, KNN dapat mengalami kinerja yang lebih lambat ketika dihadapkan pada dataset besar.or (kNN) karena klasifikasi tergantung pada k tetangga terdekat.

7. VSC (Visual Studio Code)



Gambar 2. 4 Visual Studio Code

Visual Studio Code adalah sebuah lingkungan pengembangan terintegrasi IDE *(Integrated Development Environment*) yang dikembangkan oleh Microsoft. IDE ini

menyediakan berbagai alat dan fitur untuk membantu pengembang dalam proses pengembangan perangkat lunak. Aplikasi ini juga dapat digunakan untuk membuat serta mengedit *source code* memakai berbagai bahasa pemrograman seperti Node.js, TypeScript, dan JavaScript. Bahkan untuk visualnya tergolong kompatibel dengan bahasa serta runtime environment lainnya seperti .NET, Java, bahasa Python, dan PHP. Berikut adalah beberapa fitur utama *Visual Studio Code;*

a. Editor Kode:

- Visual Studio menyertakan editor kode yang kuat dengan penyorotan sintaks, penyelesaian otomatis, dan integrasi pengelolaan kode sumber.
- Mendukung berbagai bahasa pemrograman seperti C#, C++, VB.NET, F#, Python, dan banyak lagi.

b. Debugging:

- Fitur debugging yang canggih dengan kemampuan langkah demi langkah, poin henti, dan penelursuran variabel.
- Dukungan untuk debug aplikasi lokal atau lingkungan yang di host.

c. IntelliSense:

- Menyediakan penyelesaian kode otomatis dan informasi kontekstual saat menulis kode.
- Membantu meningkatkan produktivitas dan mengurangi kesalahan.

d. Designer Antarnuka Pengguna (UI):

- Alat desain UI yang memungkinkan pengembang membuat antarmuka pengguna dengan mudah menggunakan drag-and-drop
- Mendukung pembuatan antarmuka untuk aplikasi dekstop, web, dan seluler.

e. Pengelolaan Proyek:

- Alat untuk membuat, mengelola, dan memelihara proyek.

- Mendukung pengelolaan referensi, paket *NuGet*, dan integrasi dengan sistem kontrol versi seperti *Git.*

f. Pendukung Pengembangan Berkas Besar:

- Dukungan untuk pengembangan proyek besar dengan ribuan file dan struktur folder.

g. Integrasi dengan *Azure:*

- Integrasi yang kuat dengan *platform cloud Microsoft Azure* untuk pengembangan, pengujian, dan penyebaran aplikasi *cloud*.

h. Pendukung Pemecahan Masalah:

- Alat untuk menganalisis kode dan memberikan saran perbaikan untuk meningkatkan kualitas dan keamanan kode.

i. Pengujian dan Performa:

- Alat pengujian otomatis dan analisis performa untuk membantu mengidentifikasi dan memper
- baikimasalah performa aplikasi.

j. Ekosistem Ekstensi:

- Duukungan untuk ekstensi pihak ketiga yang memperluas fungsionalitas Visual Studio.

k. Pemeliharaan Kode Sumber

- Integrasi dengan sistem kontrol versi seperti Git dan kemampuan untuk melihat histori perubahan, melakukan gabungan (*merge*), dan lainnya.

l. Pendukung Pemograman *Cross-Platform*

- Mendukung pengembangan untuk platform berbeda seperti Windows, Linux, dan macOS.

8. UML (Unified Modeling Language)

UML (*Unified Modeling Language*) adalah "bahasa" yang telah berubah menjadi standar bisnis untuk membayangkan, merencanakan, dan melaporkan kerangka kerja pemrograman. UML menawarkan norma untuk model perencanaan kerangka kerja.

(Kroenke et al 2018), *Unified Modelling Language (UML)* adalah seperangkat diagram, struktur, dan teknik untuk memodelkan dan merancang program dan aplikasi berorientasi objek. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem. *UML* adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap seb uah sistem atau perangkat lunak berbasis objek. *UML* merupakan singkatan dari *Unified Modeling Language*. *UML* juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan.

Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena developer harus melakukan penelusuran dan mempelajari kodeprogram *UML* juga dapat menjadi alat bantu untuk transfer ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainya. Tidak hanya antar *developer* terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya *UML*.

UML diciptakan oleh *Object Management Group* yang diawali dengan versi 1.0 pada Januari 1997. Dalam pengembangan berorientasi objek ada beberapa prinsip yang harus dikenal: *Object, Class, Abstraction, Encapsulation, Inheritance* dan *Polymorphism.*

Dalam UML sendiri terdapat beberapa diagram yaitu :

a. *Use Case* Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor

dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case* adalah suatu urutan interaksi yang saling berkaitan antara sistem dan aktor. *Use case* dijalankan melalui cara menggambarkan tipe interaksi antara *user* suatu program (sistem) dengan sistemnya sendiri (Ade Hastuty dkk., 2023).

Tabel 2. 1 *Use Case* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Use case*	Menggambarkan kegunaan yang diberikan oleh kerangka kerja sebagai unit yang secara efektif memperdagangkan pesan antar unit, yang dikomunikasikan menggunakan kata-kata tindakan.
2		*Actor*	Penghibur adalah pertimbangan dari orang lain atau kerangka kerja yang menggerakkan kemampuan kerangka objektif. Harus diperhatikan bahwa penghibur itu terkait dengan kasus penggunaan, tetapi tidak memiliki

Tabel 2. 2 Lanjutan Tabel

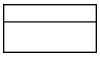
NO	GAMBAR	NAMA	KETERANGAN
			kendali atas kasus penggunaan
3		*Association*	Apa yang menghubungkan antara objek satu dengan objek yang lainnya.
4		*Generalization*	Hubungan di mana objek anak berbagi cara berperilaku dan desain informasi item di atas objek induk.
5		*Include*	Pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan

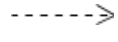
			mempengaruhi elemanyang bergantung padanya elemen yang tidak mandiri.
7		*Extend*	Menunjukkan bahwa use case objektif memperluas cara berperilaku use case sumber pada titik tertentu
8		*System*	Menunjukkan bundel yang menunjukkan kerangka kerja terbatas.
9		*Collaboration*	Kerja sama memutuskan dan berbagai komponen yang bekerja sama memberikan perilaku yang lebih penting daripada agregat dan komponen-komponennya.
10		*Note*	Komponen aktual yang ada saat aplikasi dijalankan dan menangani aset pemrosesan.

b. *Class* diagram

Class diagram adalah visual dari struktur sistem program pada jenis-jenis yang di bentuk. *Class* diagram merupakan alur jalannya database pada sebuah sistem. *Class* diagram merupakan penjelasan proses database dalam suatu program. Dalam sebuah laporan sistem maka class diagram ini wajib ada.

Tabel 2. 3 Simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk *(ancestor).*
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*
5		*Realization*	Operasi yang benar-banar dilakukan oleh suatu *object.*

NO	GAMBAR	NAMA	KETERANGAN
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.

c. *Activity* diagram

Activity diagram adalah sesuatu yang menjelaskan tentang alir kegiatan dalam program yang sedang dirancang, bagaimana proses alir berawal, keputusan yang mungkin terjadi, dan bagaimana sistem akan berakhir.

Tabel 2. 4 Simbol *Activity* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu Aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang berubah menjadi beberapa aliran.

d. *Sequence* Diagram

Sequence Diagram adalah suatu diagram yang menjelaskan interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut.

NO	GAMBAR	NAMA	KETERANGAN

Tabel 2.5 Simbol *sequence* diagram

1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi objek yang memuat informasi-informasi tentang aktifitas yang terjadi

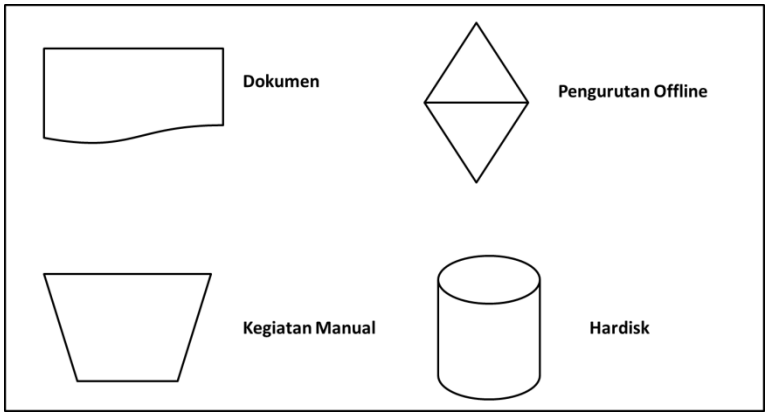
9. *Flowchart*

Flowchart dalam bahasa indonesia diterjemahkan sebagai diagram alir. Dari dua kata ini, maka dapat kita bayangkan bahwa flowchart itu berbentuk diagram yang bentuknya dapat mengalirkan sesuatu. Hal ini memang benar, flowchart memang melukiskan suatu aliran kegiatan dari awal hingga akhir mengenai suatu langkah-langkah dalam penyelesaian suatu masalah. Masalah tersebut bermacam-macam, mulai dari masalah sederhana sampai yang kompleks.

Flowchart adalah representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri atas sekumpulan simbol, dimana masing-masing simbol merepresentasikan suatu kegiatan tertentu. Dengan kata lainflowchart merupakan suatu gambar yang menjelaskan urutan: pembacaan data, pemrosesan data, pengmbilan eputusan terhadap data, penyajian hasil pemrosesan data (Ade Hastuty dkk., 2023)

a) *Flowchart* Sistem

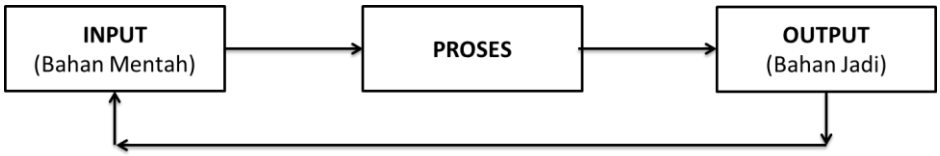
Bagan dengan simbol-simbol tertentu yang mengambarkan ururtan prosedur dan proses suatu filedalam suatu media menjadi file di dalam media lain, dalam suatu sitem pengolahan data. Beberapa contoh flowchart sistem:



Gambar 2. 5 Flowchart Sistem

b) *Flowchart* Program

Yaitu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses dan hubungan antar proses secara mendatil di dalam suatu program. Dalam pembuatan flowchart program tidak ada rumus atau patokanyang bersifat mutlak karena flowchart merupkan gambaran hasil pemikiran dalam menganalisis sesuatu masalah dengan komputer sehingga flowchart yang dihasilkan bervariasi antara satu pemrogram dengan yang lainnya. Namun secara garis besar setiap pengolahan selalu terdiri atas 3 bagian utama, yaitu: input, proses pengolahan dan Output.



Gambar 2. 6 Flowchart Program

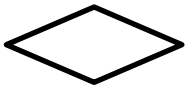
Untuk pengolahan data dengan komputer urutan dasar pemecahan sutau masalah.

- *START*, Berisi pernyataan untuk persiapan peralatan yang diperlukan sebelum pemecahan menangani pemecahan masalah.
- *READ*, berisis peryataan kegiatan untuk membaca data dari suatuperalatan input.

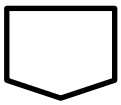
- PROSES, berisi kegiatan yang berkaitan dengan pemecahan persoalan sesuai dengan yang dibaca.
- *WRITE*, berisis pernyataan untuk merekam hasil kegiatanke peralatan output.
- *END*, menakhiri kegiatan pengolahan..

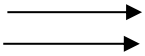
Berikut merupakan beberapa contoh simbol flowchart yang disepakati oleh dunia pemograman

Tabel 2. 6 Simbol-Simbol Flowchart

Simbol	Nama dan Fungsinya
	*Terminator* *Menandai awal dan akhir dari flowchart*
	*Input* *Digunakan untuk melambangkan kegiatan penerimaan input.*
	*Process* *Melambangkan kegiatan pemrosesan input.*
	*Decision* *Menunjukan suatu kondisi tertentu yang akan menghasilkan dua kemunkinan jawaban:ya / tidak*
	*Connector (one-page)* *Menyatakan sambungan dari proses ke proses lainya dalam halaman yang sama*

Tabel 2. 7 Tabel Lanjutan

Simbol	Nama dan Fungsinya
	*Connector (offpage)* *Menyatakan sambungan dari proses ke proses lainya dalam halaman yang berbeda*
	*Prefefned Process* *Menyatakan penyediaan tempat penyimpanan suatu pengolaan untuk memberi harga awal*

	*Punched card* *Menyatakan input berasal dari kartu atau output ditulis ke kartu*
	*Document* *Mencetak keluaran dalam bentuk dokumen(Melalui printer)*
	*Flow* *Menyatakan jalannya arus suatu proses*

10. Pixel

Pixel, singkatan dari "*picture element*," adalah unit terkecil dari gambar digital atau tampilan layar. Setiap pixel mewakili satu titik pada gambar atau layar yang dapat menampilkan warna tertentu. Dalam konteks gambar digital, pixel adalah bagian dari matriks yang membentuk gambar secara keseluruhan. Semakin banyak pixel yang ada dalam gambar atau layar, semakin tinggi resolusi dan detail gambar tersebut.

Pada layar komputer atau perangkat seluler, pixel juga bisa merujuk pada titik-titik cahaya individual yang menyusun tampilan layar. Setiap pixel pada layar terdiri dari sub-pixel yang biasanya berwarna merah, hijau, dan biru (RGB) yang digabungkan untuk menghasilkan berbagai warna. Resolusi layar, misalnya 1920x1080, menunjukkan jumlah pixel horizontal dan vertikal pada layar tersebut.

11. Algoritma

Ditinjau dari asal-usul katanya, kata Algoritma sendiri mempunyai sejarah yang aneh. Orang hanya menemukan kata algorism yang berarti proses menghitung dengan angka arab. Ditinjau dari asal-usul katanya, kata Algoritma sendiri mempunyai sejarah yang aneh. Orang hanya menemukan kata algorism yang berarti proses menghitung dengan angka arab.

Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis". Kata logis merupakan kata kunci dalam algoritma. Langkah-

langkah dalam algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar. Dalam bebeurutan langkah untuk melakukan pekerjaan tertentu. Pertimbangan dalam pemilihan algoritma adalah, pertama, algoritma haruslah benar. Artinya algoritma akan memberikan keluaran yang dikehendaki dari sejumlah masukan yang diberikan. Tidak peduli sebagus apapun algoritma, kalau memberikan keluaran yang salah, pastilah algoritma tersebut bukanlah algoritma yang baik.

Algoritma yang baik tanpa pemilihan struktur data yang tepat akan membuat program menjadi kurang baik, Demikian juga sebaliknya. Dalam pembuatan algoritma mempunyai banyak keuntungan di antaranya:

a) Pembuatan atau penulisan algoritma tidak tergantung pada bahasa pemrograman manapun, artinya penulisan algoritma independen dari bahasa pemograman dan komputer yang melaksanakannya.
b) Notasi algoritma dapat diterjemahkan ke dalam berbagai bahasa pemrograman.
c) Apapun bahasa pemrogramannya, output yang akan dikeluar- kan sama karena algoritmanya sama.

Beberapa hal yang perlu diperhatikan dalam membuat algoritma :

a) Teks algoritma berisi deskripsi langkah-langkah penyelesaian masalah. Deskripsi tersebut dapat ditulis dalam notasi apa-pun asalkan mudah dimengerti dan dipahami.
b) Tidak ada notasi yang baku dalam penulisan teks algoritma seperti notasi bahasa pemrograman. Notasi yang digunakan dalam menulis algoritma disebut notasi algoritmik.
c) Setiap orang dapat membuat aturan penulisan dan notasi algoritmik sendiri. Hal ini dikarenakan teks algoritma tidak sama dengan teks program. Namun, supaya notasi

algoritmik mudah ditranslasikan ke dalam notasi bahasa pemrograman tertentu, maka sebaiknya notasi algoritmik tersebut berkorespondensi dengan notasi bahasa pemrograman secara umum.

d) Notasi algoritmik bukan notasi bahasa pemrograman, karena itu pseudocode dalam notasi algoritmik tidak dapat dijalankan oleh komputer. Agar dapat dijalankan oleh komputer, pseudocode dalam notasi algoritmik harus ditranslasikan atau diterjemahkan ke dalam notasi bahasa pemrograman yang dipilih. Perlu diingat bahwa orang yang menulis program sangat terikat dalam aturan tata bahasanya dan spesifikasi mesin yang menjalannya.
e) Algoritma sebenarnya digunakan untuk membantu kita dalam mengkonversikan suatu permasalahan ke dalam bahasa pemrograman.
f) Algoritma merupakan hasil pemikiran konseptual, supava dapat dilaksanakan oleh komputer, algoritma harus ditranslasikan ke dalam notasi bahasa pemrograman

Agar algoritma bisa ditranslasika kedalam bahasa pemrograman. Ada beberapa hal yang harus diperhatikan pada translasi tersebut, yaitu:

a) Pendeklarasian variabel Untuk mengetahui dibutuhkannyapendeklarasianvariabeldalampenggunaanbahasa pemrogramanapabila tidak semua bahas pemrograman membutuhkannya.
b) Pemilihan tipe data Apabila bahasa pemrograman yang akandigunakan membutuhkan pendeklarasian variabel maka perluhal ini dipertimbangkan pada saat pemilihan tipe data.
c) Pemakaian instruksi-instruksi Beberapa instruksi mempunyai kegunaan yang sama tetapi masing-masing memiliki kelebihan dan kekurangan yang berbeda.
d) Aturan sintaksis Pada saat menuliskan program kita terikat dengan aturan sintaksis dalam bahasa pemrograman vang akan digunakan.

e) Tampilan hasil Pada saat membuat algoritma kita tidak memikirkan tampilan hasil yang akan disajikan. Hal-hal teknis ini diperhatikan ketika mengkonversikan-nya menjadi program.

f) Cara pengoperasian compiler atau interpreter. Bahasa pemrograman yang digunakan termasuk dalam kelompok Compiler atau interpreter.

Algoritma yang dapat melakukan enkripsi terhadap file yang akan dikirim, salah satu algoritma yang sering digunakan adalah Algoritma *vernam chiper*.(Febrian,2023). Algoritma c4.5 salah satu Algoritma yang dapat digunakan untuk membuat pohon keputusan (decision tree).(Syam,.dkk2024)

12. Pemograman

Belajar memprogram tidak sama dengan belajar bahasa pemrograman. Belajar memprogram adalah belajar tentang metodologi pemecahan masalah, kemudian menuangkan dalam suatu notasi tertentu yang mudah dibaca dan dipahami. Sedangkan belajar bahasa pemrograman berarti belajar memakai Suatu bahasa aturan-aturan tata bahasanya, pernyataan. pernyataannya, tata cara pengoperaasian compiler-nya, dan memanfaatkan pernyataanpernyataan tersebut untuk membua bahasa itu saja. Sampal Saat pemrogram, antara lain Bahasa rakitan (assembly), Fortran, Cobol, Ada, PL/I, Algol, Pascal, C program yang ditulis hanya dalam ba ini terdapat puluhan Bahasa Ct+, Basic, Prolog, LISP, PRG, b bahasa-bahasa simulasi seperti CSMP, Simscript, GPSS, Dinamo. Berdasarkan terapannya bahasa pemrograman dapat digolongkan atas dua kelompok.

Bahasa pemrograman bertujuan khusus,besar: Yang termasuk kelompok ini adalah Cobol (untuk terapar bisnis dan administrasi). Fortran (terapan komputasi ilmiah bahasa rakitan (terapan pemrograman mesin), Prolog (terapa kecerdasan buatan), bahasa-bahasa simulasi, dan sebagainya. Bahasa perograman bertujuan umum, Yang dapat digunakan

untuk berbagai aplikasi. Yang termasuk kelompok ini adalah bahasa Pascal, Basic dan C. Tentu sa pembagian ini tidak kaku.

Bahasa-bahasa bertujuan khusus tidak berarti tidak bisa digunakan untuk aplikasi lain. Cobol misalnya, dapat juga digunakan untu terapan ilmiah, hanya saja kemampuannya terbatas. Yang jelas bahasa-bahasa pemrograman yang berbeda dikembangkan untu bermacam-macam terapan yang berbeda pula.

Berdasarkan pada apakah notasi bahasa pemroaman lebih "deka ke mesin atau ke bahasa manusia, maka bahasa pemrograma dikelompokkan atas dua macam:

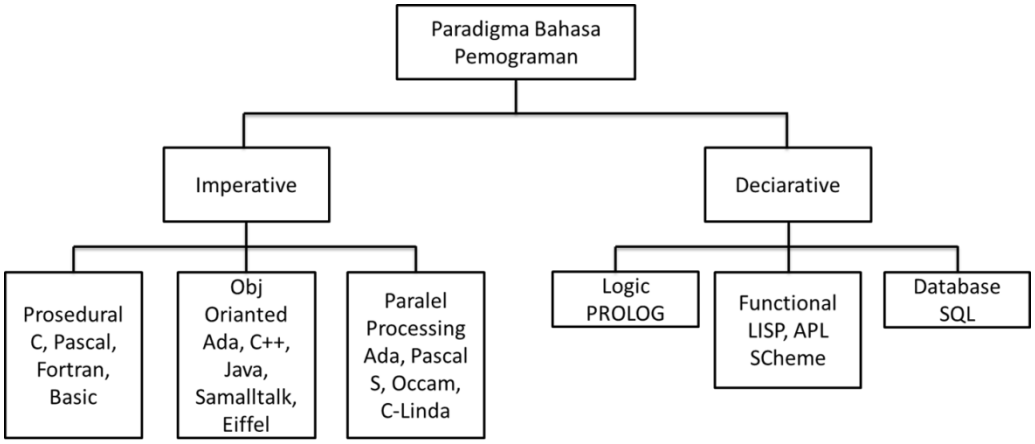
a) Bahasa tingkat rendah.

Bahasa jenis ini dirancang agar setiap instruksinya langsung dikerjakan komputer tanpa harus melalui penerjemah (translator) contohnya adalah bahasa mesin, CPU mengambil instruksi dari memori, langsung mengerti dan langsung mengerjakan operasinya. Bahasa tingkat rendah bersifat primitif, sangat sederhana orientasinya lebih dekat ke mesin, dan sulit dipahami manusia. Sedangkan bahasa rakitan dimasukkan ke dalam kelompok ini karena alasan notasi yang dipakai dalam bahasa ini lebih dekat ke mesin, meskipun untuk melaksanakan instruksinya masih perlu penerjemahan ke dalam bahasa mesin.

b) Bahasa tingkat tinggi,

yang membuat pemrograman lebih mudah dipahami, lebih "manusiawi", dan berorientasi ke Bahasa manusia (bahasa Inggris). Hanya saja, program dalam Bahasa tingkat tinggi tidak dapat langsung dilaksanakan oleh komputer. la perlu diterjemahkan terlebih dahulu oleh sebuah translator bahasa (yang disebut

kompilator atau *compiler*) ke dalam bahasa mesin sebelum akhirnya dieksekusi oleh CPU. Contoh bahasa tingkat tinggi adalah Pascal, PL/I, Ada, Cobol, Basic, Fortran,C, C++, dan sebagainya. Bahasa pemrograman bias juga dikelompokkan berdasarkan pada tujuan dan fungsinya. Di antaranya adalah:



Gambar 2. 7 Pembagian Bahasa Pemograman

13. Python

Python adalah bahasa pemrograman tingkat tinggi yang mudah dipelajari dan digunakan karena sintaksnya yang sederhana dan mendekati bahasa manusia. Dikembangkan oleh Guido van Rossum dan dirilis pertama kali pada tahun 1991, Python dirancang untuk menjadi bahasa yang mudah dibaca dan ditulis, sehingga banyak digunakan untuk berbagai keperluan, mulai dari pemrograman dasar hingga proyek ilmiah kompleks.



Gambar 2. 8 Logo Python

Beberapa karakteristik utama Python adalah:

a) Bahasa Serbaguna (*General-purpose*): *Python* dapat digunakan untuk banyak jenis pengembangan, termasuk web, aplikasi desktop, pengolahan data, kecerdasan buatan, dan otomatisasi.

b) Sintaks yang Mudah Dipahami: Python memiliki sintaks yang sederhana, yang membuatnya lebih mudah dipahami oleh pemula. Contohnya, untuk menampilkan teks, cukup menulis `print("Hello, World!")`.

c) Interaktif dan Dinamis: Python dapat dijalankan dalam mode interaktif, memungkinkan pengguna untuk langsung melihat hasil dari setiap baris kode. Ini juga mendukung pengelolaan memori secara otomatis.

d) *Library* dan Komunitas yang Luas: Python memiliki ribuan *library* atau pustaka yang membantu dalam berbagai tugas, seperti *NumPy* untuk perhitungan ilmiah, Pandas untuk analisis data, dan *TensorFlow* untuk pembelajaran mesin.

e) Multi-Paradigma: Python mendukung berbagai gaya pemrograman, termasuk pemrograman berorientasi objek, pemrograman fungsional, dan prosedural.

Python banyak digunakan dalam pengembangan aplikasi web, analisis data, sains data, otomatisasi, hingga pengembangan kecerdasan buatan, karena fleksibilitasnya yang tinggi.

Python termasuk bahasa pemrograman yang imperative, tetapi juga mendukung beberapa pendekatan declarative. Dalam imperative programming, kita memberi tahu komputer bagaimana melakukan sesuatu, dengan menulis langkah demi langkah secara eksplisit. Contoh perintah imperatif di Python adalah penggunaan *loop for* atau *while* untuk mengulang tugas tertentu. Sebagian besar kode Python ditulis dengan cara ini.

Namun, Python juga mendukung declarative *programming* melalui beberapa fitur, seperti pemrograman fungsional (misalnya, fungsi map, filter, atau lambda) dan pustaka eksternal, termasuk ORM (*Object-Relational Mapping*) seperti *SQLAlchemy* untuk pengelolaan database, yang membuat kita hanya perlu menuliskan apa yang ingin dicapai, bukan bagaimana caranya.

Jadi, Python adalah bahasa multi-paradigm karena memungkinkan gaya imperatif dan deklaratif sesuai kebutuhan.

B. Metode Pengujian

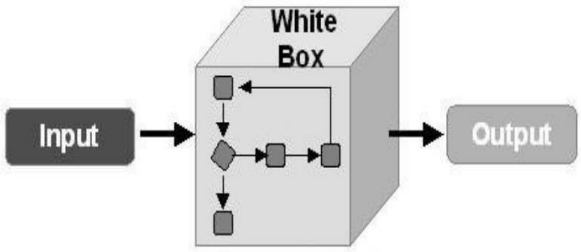
Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi, query, atau jalur navigasi yang mewakili penggunaan sistem yang *tipical*, kritis atau abnormal. Isu kunci pada pengembangan sistem adalah pemilihan sekelompok *test-case* yang cocok, sekecil dan secepat mungkin, untuk meyakinkan perilaku sistem secara detail. Pengujian harus mencakup *unit testing*, yang mengecek validasi dari prosedur dan fungsi-fungsi secara independen dari komponen sistem yang lain. Kemudian modul testing harus menyusul dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan.

Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk

melakukan unit testing, yaitu:

1. **White Box Testing**

Pengujian kotak putih adalah teknik rencana kasus uji yang memanfaatkan desain untuk mendapatkan kasus uji, pengujian ini digunakan untuk meramalkan bagaimana produk berfungsi secara lengkap dengan cara rasional, produk dicoba dengan keadaan ada dan siklus.

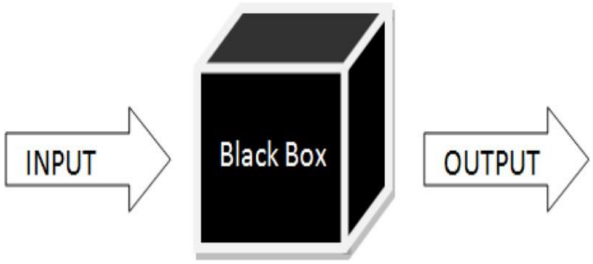


Gambar 2. 9 White Box Testing

Ilustrasi *white box testing* ini merupakan peringatan ketika klien memasukkan beberapa frase rahasia klien yang tidak dapat diterima, karena blunder semacam ini akan memberikan data kepada klien tentang kesalahan yang dilakukan.

2. ***Black Box Testing***

Pengujian black box adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui informasi uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitamnya. Sama seperti pengujian black box, penemuan dari tampilan luarnya (interfacenya), fungsionalitas tanpa mengetahui apa yang sebenarnya terjadi dalam komposisi detilnya (hanya mengetahui input dan hasil).



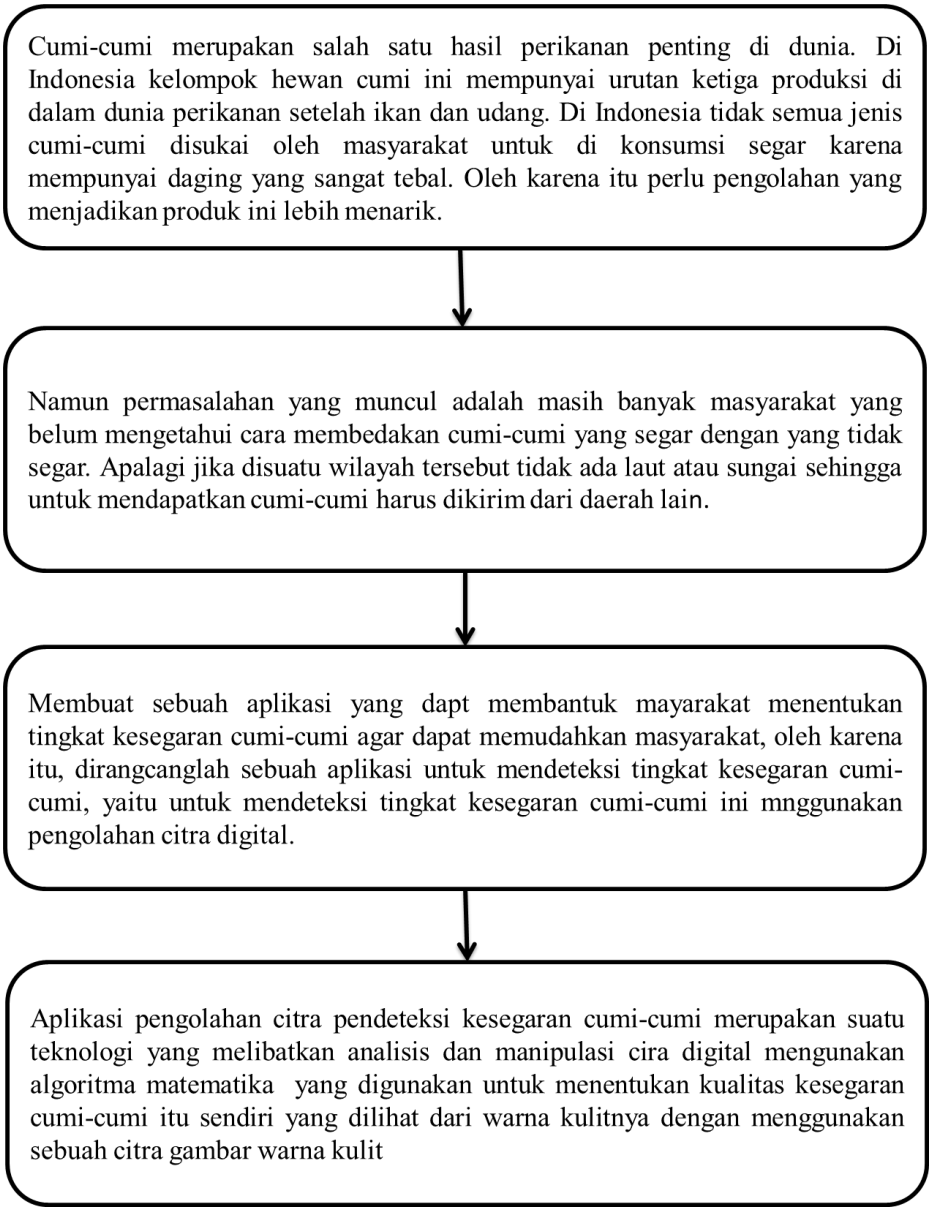
Gambar 2. 10 *Black Box Testing*

Berikut dari bagian-bagian dalam teknik yang digunakan dalam *Black Box Testing* antara lain :

a. sebuah. Digunakan untuk menguji elemen eksplisit produk.
b. Keabsahan produk yang dicoba hanya dilihat dari hasil yang diberikan.
c. Kapasitas program untuk mengatasi masalah klien dapat diperkirakan juga karena kesalahan dapat dibedakan.

C. Kerangka Pikir

Untuk lebih memahami alur penelitian di atas, di uraikan ke dalam kerangka berpikir yang akan disajikan dalam bentuk diagram berikut ini :



D. Tinjauan Penelitian Sebelumnya

Dalam melakukan sebuah penelitian atau acuan terhadap penelitian yang telah dilakukan sebelumnya, melalui berbagai buku, jurnal maupun skripsi sehingga dapat memudahkan penelitian dalam menentukan kerangka awal sebagai konsep untuk meneliti. Sehingga dalam penelitian kali ini akan dicantumkan dari hasil penelitian sebelumnya :

1. Wiwin Styorini, Anggita Pratiwi, dan Cyntia Widiasari “IDENTIFIKASI TINGKAT KESEGARAN IKAN BERBASIS ANDROID”. Pada penelitian objek yang dideteksi

adalah ikan dengan mengambil sample citra ikanang akan didentifikasi matanya menggunakan smarphone, kemudian di upload. Kemudian mengekstraksi ciri warna maksudnya adalah citra mata diekstraksi menggunakan ruang warna RGB (red,green,blue) kemudian citra mata maka akan dilanjutkan ke bagian klasifikasi. Proses selanjutnya adalah mengklasifikasi ikan, proses ini sesuai dengan warna yang telah di ektraksi menggunakan metode KNN (K-Nearest Neighbor) dan akan tampil hasilnya yaitu sangant segar, segar, kurang segar dan tidak segar.

2. Saleh Dwiyatno, Iksal, Dan Sandi Nugraha. "ALAT PENDETEKSI KESEGARAN IKAN MNGGUNAKAN METODE *K-NEAREST NEIGHBOUR* BERDASARKAN WARNA MATA BERBASIS ATMEGA 328". Pada penelitian dia menggunakan sensor warna yang digunakan untuk mengambil data berupa karakteristik warna dari mata ikan yang sedang diuji. Terdapat tiga buah filter warna led yaitu red, green, dan blue. Output dari sensor ini berupa pantulan sinar cahaya yang frekuensinya akan bervariasi terhadap nilai warna yang terdeteksi oleh LDR. Proses pengambilan data warna dari mata ikan dilakukan dengan mendekatkan sensor warna menuju ke permukaan mata ikan yang akan diuji sehingga nilai warna dari mata ikan akan ditangkap oleh LDR dari sensor ini. Alat pendeteksi kesegaran ikan yang dibuat dalam penelitian ini terdiri dari satu buah LCD yang berfungsi sebagai indikator dari hasil pembacaan sensor yang menentukan kesegaran ikan tersebut. Dan untuk menampilkan hasil pembacaan sensor LCD harus dihubungkan menuju ke mikrokontroller arduino.
3. Charmelia Ynizar Jerandu, Patrisius Batarius, dan Alfry Aristo Jansen Sinlae. "Identifikasi Kualitas Kesegaran Ikan Menggunakan Algoritma *K-Nearest Neighbour* Berdasarkan Ekstraksi Ciri Warna Hue, Saturation, Dan Value (Hsv)". Pada penelitian ini pembagian sampel citra yang ditunjukkan dapat dijelaskan bahwa

keseluruhan sampel ikan berjumlah 60 ekor yang terbagi menjadi 30 sampel untuk ikan dengan kualitas segar dan 30 sampel untuk ikan dengan kualitas tidak segar dimana jumlah jumlah citra untuk sampel ikan segar dan tidak segar masing-masing sebanyak 120 citra yang didapat dari pengambilan gambar untuk 1 ekor ikan.citra yang telah diakuisisi selanjutnya akan dilakukan pemotongan dan pengubahan ukuran citra menjadi kecil agar oemrosesan citrapada sistem dapat berjalan dengan cepat dan lancar.untuk melakukan sebuah ekstraksi fitur dalam mengekstraksi ciri warna pada citra ikan menggunakan ruan warna *hue, saturation,* dan *velue* yang dimulai dengan membaca folder data latih yang telah disiapkan pada current folder tersebut.

Pada penelitian kali ini saya mencoba membuat aplikasi pendeteksi kesegaran cumi-cumi menggunakan metode KNN dengan mengambil sample permukaan badan cumi-cumi menggunakan smarphone, kemudian di upload. Kemudian mengekstraksi ciri warna maksudnya adalah citra warna kulit/tubuh cumi-cumi diekstraksi menggunakan ruang warna RGB (red,green,blue) kemudian citra tersebut akan dilanjutkan ke bagian klasifikasi. Pr voses selanjutnya adalah mengklasifikasi warna kulit/tubuh cumi-cumi yang telah di ektraksi menggunakan metode KNN (K-Nearest Neighbour) dan akan tampil hasilnya yaitu segar, tidak segar, dan cumi busuk. Cumi-cumi segar yang ukurannya 20 sentimeter umumnya berwarna putih dengan sedikit bintik hitam. Sementara cumi-cumi yang tidak segar berukuran kecil biasanya memiliki warna keunguan dan berbintik hitam, dan cumi-cumi yang busuk meimiliki warna kuliat kecoklatan atau kehitaman.

E. Fokus Penelitian

1. Fokus pada pengumpulan data cumi-cumi yang berkaitan dengan kesegaran, termasuk fitur-fitur seperti warna selanjutnya dilakukan prepprocessing data untuk memastikan kualitas dan konsistensi data.

2. Teliti pengaruh niai K (jumlah tetangga terdekat) terhadap performa model. Cari tahu bagaimana nilai K memengaruhi akurasi dan stabilitas prediksi.
3. Bandingkan kinerja model KNN dengan menngunakan metrik jarak. Analisis cara metrik jarak memprngaruhi hasil prediksi.
4. Mempertimbangkan penggunaan metode ekstaksi fitur yang lebih canggih untuk meningkatkan kualitas dan repesentasi fitur yng lebih baik.
5. Fokus pada pembuatan antar muka pengguna yang intuitif dan onformatif . desain antar muka memungkinkan pengguna menggugah data cumi-cumi dan melihat hasil prediksi dengan jelas.
6. Teliti tentang bagaimana menerapkan dan menjaga keamanan serta kendala aplikasi, terutama jika digunakan di dalam industri pangan atau sektor terkait.
7. Pertimbangkan bagaimana model dan aplikasi ini dapat ditingkatkan agar dapat menangani volume data yang lebih besar.
8. Selain uji coba dalam lingkungan terkendali, pertimbangkan juga validasi dalam situasi praktis di industri perikananatau restoran untuk memastikan aplikasi benar-benar bermanfaat.

F. Desain Logika

1. Pengumpulan Data

   Kumpulkan dataset citra cumi-cumi yang mewakili berbagai tingkat kesegaran. Setiap citra harus diberi label kesegaran (misalnya, segar atau tidak segar).

2. Pra-Pemrosesan Citra

   Lakukan pra-pemrosesan citra untuk mengurangi noise dan memperbaiki kualitas citra, dan Anda dapat mempertimbangkan teknik seperti segmentasi, penghilangan latar belakang, dan ekstraksi fitur untuk mengidentifikasi bagian penting dari citra cumi-cumi.

3. Ekstraksi Fitur

   Ekstrak fitur-fitur relevan dari setiap citra. Fitur-fitur ini bisa berupa tekstur, warna, atau bentuk yang dapat digunakan sebagai representasi numerik dari setiap citra.

4. Pembagian Dataset

   Pisahkan dataset menjadi data pelatihan dan data pengujian. Data pelatihan akan digunakan untuk melatih model KNN, sedangkan data pengujian akan digunakan untuk menguji performa model.

5. Pelatihan Model KNN

   Latih model KNN menggunakan data pelatihan dan fitur-fitur yang diekstraksi sebelumnya. Pilih jumlah tetangga terdekat (nilai K) yang optimal, yang dapat ditentukan melalui validasi silang atau metode lainnya.

6. Klasifikasi Citra Baru

   Ketika mendapatkan citra baru, lakukan pra-pemrosesan dan ekstraksi fitur yang sama seperti pada langkah 2 dan 3. Gunakan model KNN yang telah dilatih untuk mencari K citra pelatihan terdekat dari citra baru berdasarkan fitur-fitur yang diekstraksi. Kemudian tentukan mayoritas label kesegaran dari tetangga terdekat, dan inilah label yang akan diberikan pada citra baru.

7. Evaluasi dan Pengujian

   Gunakan data pengujian untuk menguji performa model KNN. Hitung metrik evaluasi seperti akurasi, presisi, recall, dan F1-score untuk mengukur seberapa baik model dapat mendeteksi kesegaran cumi-cumi.

8. Penyempurnaan dan Optimalisasi

Jika hasil evaluasi tidak memuaskan, pertimbangkan untuk memperbaiki pra-pemrosesan, eksperimen dengan berbagai nilai K, atau bahkan mencoba metode pengklasifikasian lain yang lebih kompleks.

9. Antarmuka Pengguna (Opsional)

   Jika diinginkan, Anda dapat membuat antarmuka pengguna yang intuitif untuk mengunggah citra cumi-cumi dan mendapatkan prediksi tentang kesegarannya menggunakan model KNN yang telah dilatih.

10. Penyimpanan dan Pemeliharaan Model

    Setelah Anda puas dengan performa model, simpan model KNN yang dilatih untuk digunakan di masa depan. Pastikan untuk memantau performa model dari waktu ke waktu dan memperbarui model jika diperlukan.

BAB III

METODOLOGI PENELITIAN

A. Jenis Penelitian

Jenis penelitian ini merupakan penelitian eksperimental yang bersifat aplikatif sehingga dari ruang lingkup masalah dapat dilakukan dengan metode studi pustaka (*library research*), metode pengumpulan data lapangan (*field research*).

B. Waktu dan Tempat Penelitian

Rencana waktu yang digunakan untuk penelitian ini berlangsung selama ± 3 bulan pada bulan januari hingga maret tahun 2024 dan bertempat di kota Parepare, Sulawesi Selatan, Indonesia.

C. Alat dan Bahan Penelitian

Dalam melakukan penelitian, maka diperlukan alat dan bahan penelitian yang mendukung kegiatan penelitian tersebut. Alat dan bahan yang diperlukan antara lain:

1. Perangkat Keras *(Hardware)*

   Perangkat keras yang digunakan selama proses penelitian yaitu Laptop ACER dengan spesifikasi sebagai berikut :

   a. Processor : Intel(R) Celeron(R) CPU N3350 @ 1.10GHz
   b. RAM : 4,00 GB
   c. Harddisk : 500 GB
   d. Monitor : 11 Inch

2. Perangkat Lunak *(Software)*

Perangkat lunak yang digunakan dalam pembuatan aplikasi adalah sebagai berikut :

a. Sistem Operasi : Windows 10 Enterprise 64 bit
b. Aplikasi : Visual Studio Code
c. Bahasa pemograman : Python

D. Tahapan Penelitian

Tahap – tahap penelitian yang dimaksud dalam penelitian ini adalah berkenaan dengan proses pelaksanaan penelitian yaitu:

1. Persiapan Penelitian.

   Pada tahap ini analis melakukan perencanaan penelitian. Penyusunan eksplorasi yang dimaksud adalah menyiapkan buku, artikel tentang pokok-pokok penelitian dan produk yang digunakan selama pemeriksaan.

2. Pengumpulan data

   Pada tahap ini peneliti mengumpulkan informasi tenteng topik dari tugas akhir melalui buku-buku dan Web sejenis.

3. Analisis

   Pada tahap pemeriksaan, spesialis membedah kerangka kerja yang diterapkan sekarang didirikan pada kemudian membentuk masalah yang menjadi subjek eksplorasi sehingga pemikiran kritis elektif dapat dibuat.

4. Perancangan

   Spesialis kemudian merencanakan aplikasi yang diinginkannya berdasarkan pemikiran kritis elektif.

5. Pengujian

Setelah melakukan rencana, spesialis kemudian, pada saat itu, mencoba konsekuensi dari rencana yang telah dibuat. Jika hasil rencana memiliki kekurangan atau kekurangan, kembali ke tahap pemeriksaan.

6. Implementasi

Setelah tidak terdapat kekurangan pada program maka aplikasi siap digunakan oleh *user*.

E. Desain Sistem

Pada gambar di bawah ini adalah sebuah tampilan dari proses system yang di usulkan atau alur dari apkasi yang akan di bangun.

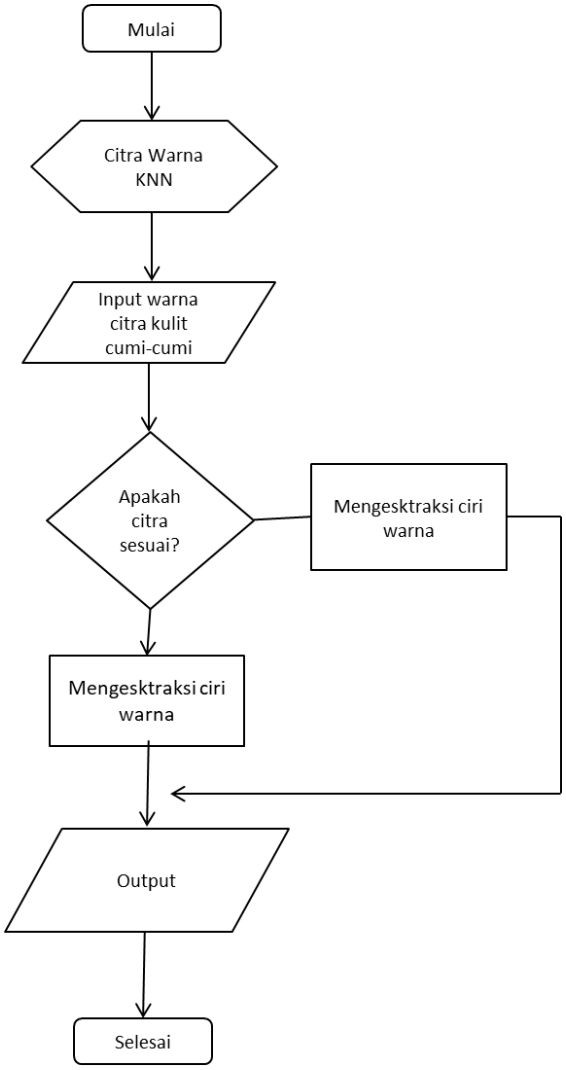
Desain sistem



Gambar 3. 1 Desain sistem

F. Flowchart

Pada gambar di bawah ini adalah sebuah tampilan dari flowchart yang di usulkan atau alur dari apkasi yang akan di bangun.



Gambar 3. 2 Flowchart

BAB IV

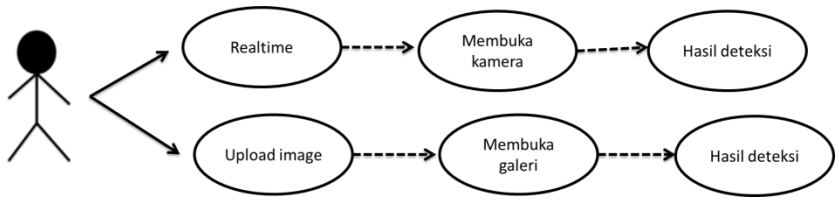
HASIL DAN PEMBAHASAN

A. Analisis Aliran Data Dengan UML

Dalam analisis sistem ini. Penulis menggunakan *Usse Case Diagram, Actifity Diagram* dan *Sequence Deiagram.*

1. Use Case Diagram

Use Case Diagram berfungsi untuk menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (aktor).



Gambar 4. 1 Desain Sistem

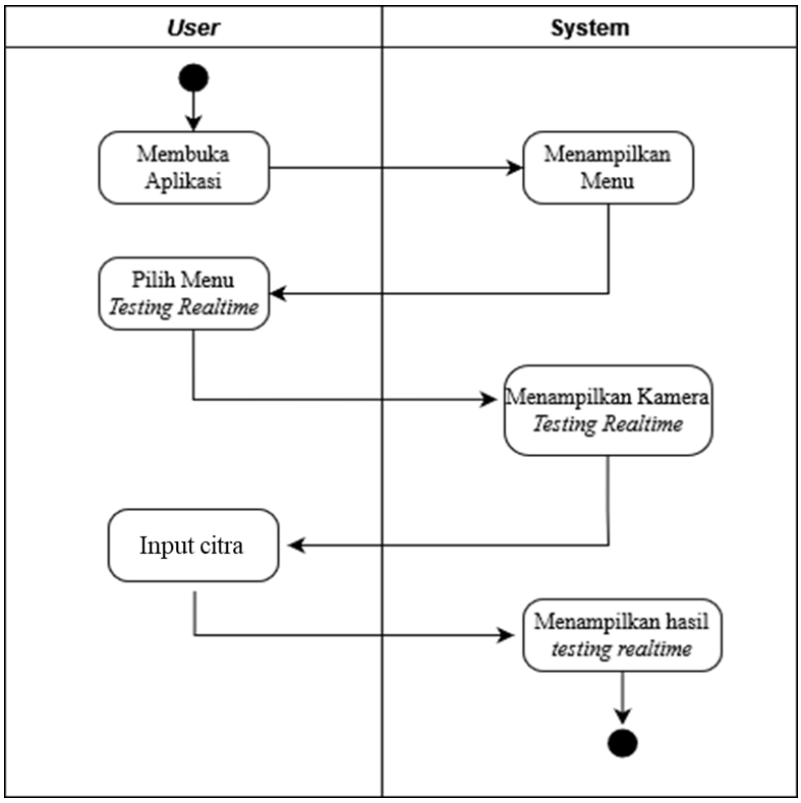
Tabel 2. 8 Penjelasan use case diagram *actor user*

Nama *use case*	Deskripsi *use case*
Testing Realtime	*Use case* ini menjelaskan bagaimana proses membuka kamera untuk deteksi objek secara langsung dan mendapatkan hasil secara langsung juag.
Test gambar	*Use case* ini menjelaskan bagaimana menampilkan objek gambar yang ingin dideteksi dengan mendeteksi gambar yang sudah ada lalu dilihat nilai histogramnya stelah tu baru didapatkan hasilnya

2. Activity diagram

Activity Diagram ini menjelaskan tentang aktivitas-aktivitas yang terjadi dalam sebuah aliran proses pada sebuah sistem. Activity ini menjelaskan mengenai proses urutan yang akan dilalui oleh user dalam menjalankan sebuah aplikasi yang mulai dari halaman utama yang terdiri dari beberapa menu yaitu menu *Realtime, dan Upload image,*

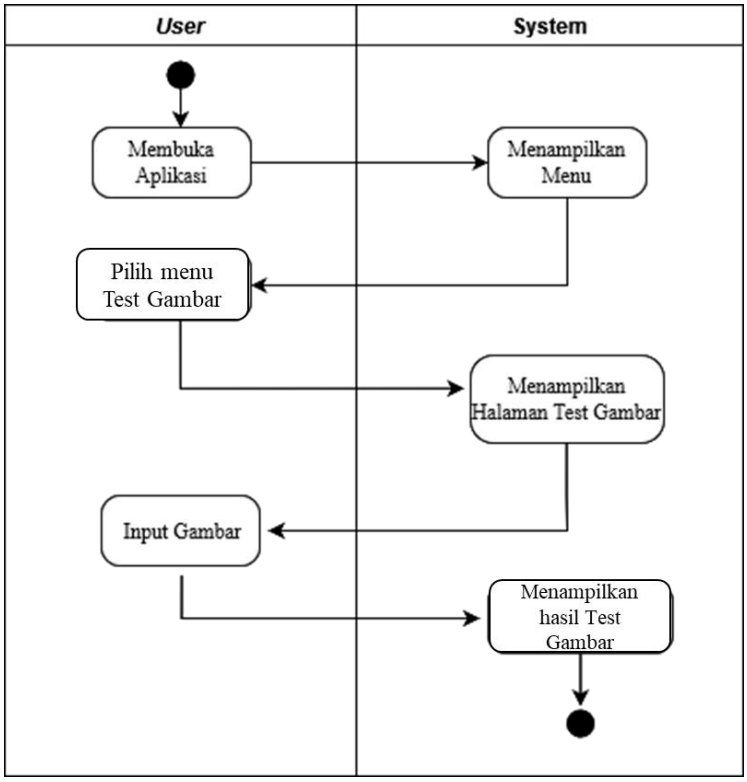
1) Activity diagram *Realtime*



Gambar 4. 2 *Activity diagram Realtime Detection*

Activity diagram di atas menjelaskan bahwa ketika *user* membuka aplikasi ini akan menampilkan menu kemudian *user* memilih menu *realtime detection* setelah itu menampilkan kamera *testing realtime* user mengimput objek secara langsung dan makan akan menampilkan hasil dari testing realtime.

2) Activity diagram Upload image



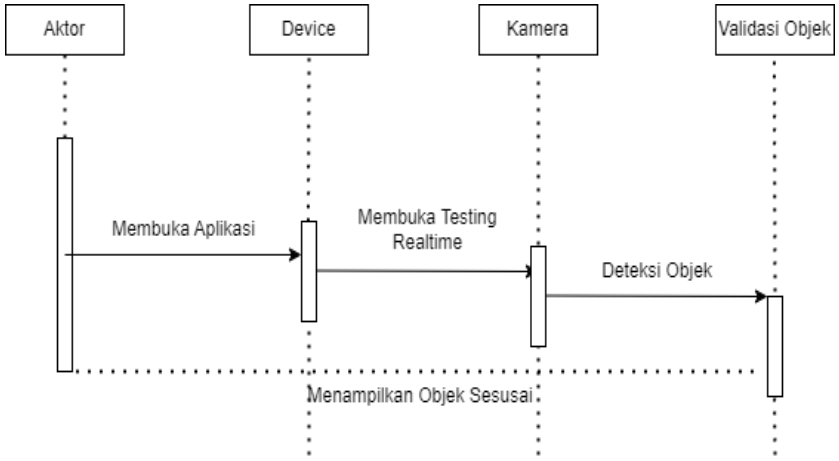
Gambar 4. 3 *Activity diagram Upload image*

Activity diagram di atas menjelaskan bahwa ketika *user* membuka aplikasi ini akan menampilkan menu kemudian *user* memilih menu test gambar setelah itu menampilkan halaman test gambar kemudian *user* mengimput gambar dari galeri dan makan akan menampilkan hasil dari test gambar

3. *Sequence diagram*

Sequence diagram adalah salah satu jenis diagram interaksi dalam UML (*Unified Modeling Language)* yang digunakan untuk menggambarkan interaksi antara objek atau komponen dalam sebuah sistem secara kronologis. Diagram ini mengilustrasikan alur pesan atau peristiwa di antara berbagai elemen sistem dalam rentang waktu tertentu.

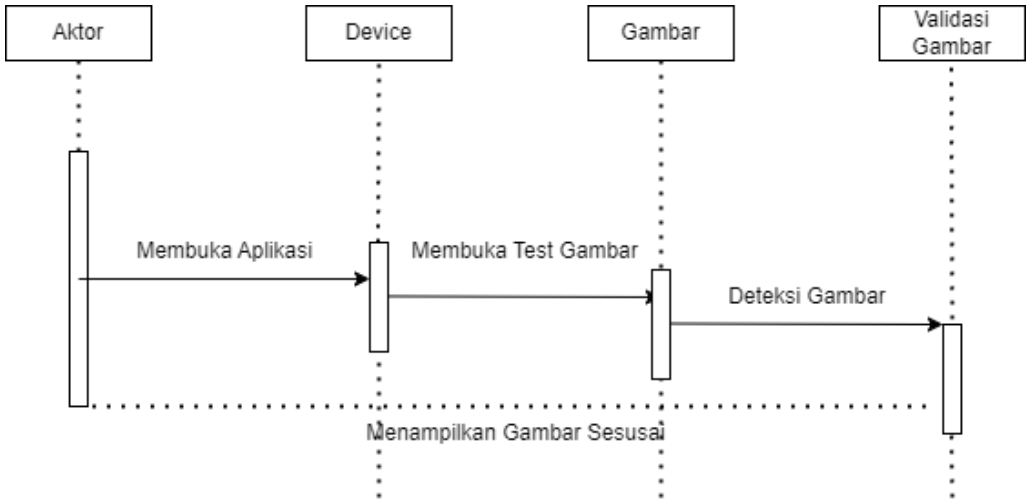
1) *Sequence diagram realtime detection*



Gambar 4. 4 *sequence diagram realtime detection*

Diagram Sequence di atas menjelaskan bahwa ketika *user* memulai Aplikas maka *Device Android* yang digunakan akan meload data untuk selanjutnya memulai aplikasi ini, setelah itu akan tampil menu *testing realtime* setelah itu user menginput suatu objek secara langsung dan akan menampilkan hasil dari objek

2) *Sequence diagram test* gambar



Gambar 4. 5 *Sequence diagram* test gambar

Diagram Sequence di atas menjelaskan bahwa ketika *user* memulai Aplikas maka *Device Android* yang digunakan akan meload data untuk selanjutnya memulai aplikasi ini, setelah itu akan tampil menu *test* gambar setelah itu user menginput suatu gambar yang telah tersimpan di galeri dan menampilkan hasil sesuai gambar.

B. Rancangan Sistem

1. Rancangan Tampilan Menu

   Merupakan Tampilan Menu pada aplikasi Deteksi Kesegaran Cumi Menggunakan Algoritma KKN

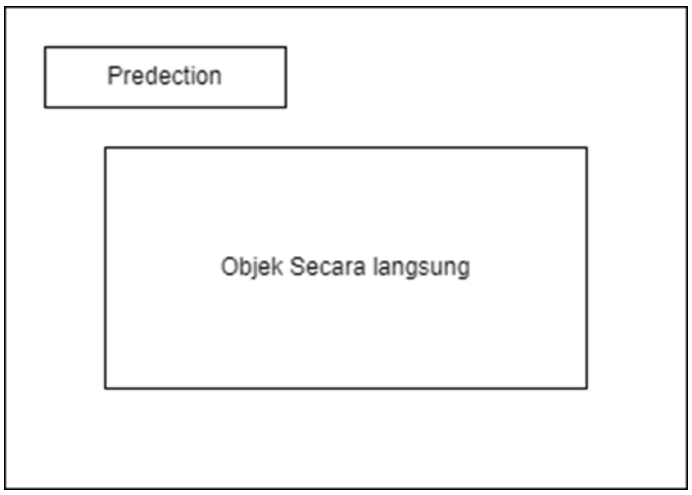


Gambar 4. 6 Rancangan Tampilan Menu

Tampilan pada aplikasi Deteksi Kesegaran Cumi Menggunakan Algoritma KKN terdiri dari tiga menu utama, yaitu Real-Time, dan Test Gambar.Pada menu Testing Real-Time, pengguna dapat memasukkan data sensor langsung. Menu Test Gambar memungkinkan pengguna mengunggah foto cumi, yang kemudian dianalisis menggunakan teknik pengolahan citra untuk menentukan tingkat kesegarannya. Hasil analisis dari kedua metode ditampilkan secara jelas dengan indikator visual dan deskripsi.

2. Rancangan Tampilan *Realtime Detection*

Rancangan tampilan *Real-Time* pada aplikasi Deteksi Kesegaran Cumi Menggunakan Algoritma KKN didesain untuk memungkinkan pengguna melakukan analisis kesegaran cumi secara langsung menggunakan kamera. Pada halaman ini, terdapat beberapa komponen yang langsung menampilkan kamera utama, Selain itu, Analisis yang akan memproses data menggunakan algoritma K-Nearest Neighbors (KNN) dan menampilkan hasil klasifikasi kesegaran cumi dalam bentuk indikator visual, seperti ikon (segar, tidak segar, atau busuk). Tampilan ini dirancang agar mudah dipahami dan digunakan oleh pengguna, baik dari kalangan industri perikanan maupun konsumen umum.



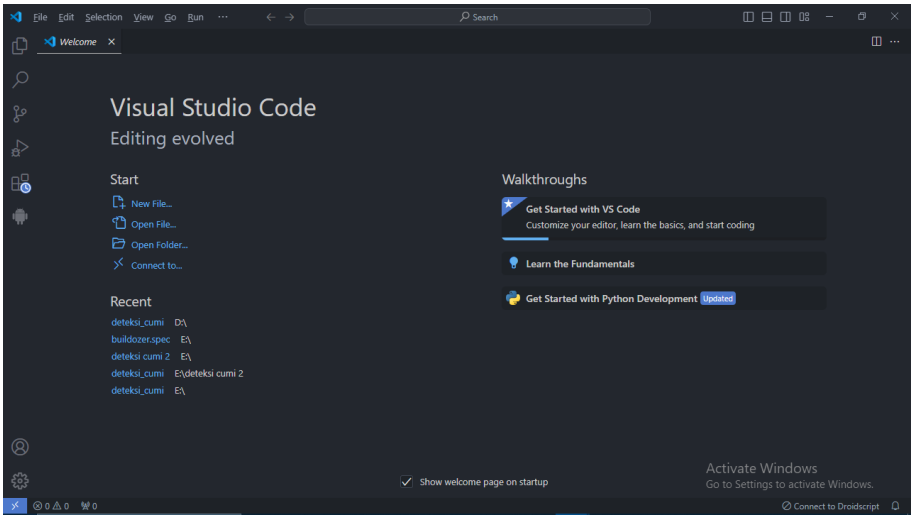
Gambar 4. 7 Rancangan Tampilan *Realtime Detection*

C. Detail Aplikasi

1. Tampilan Visual Studio Code

Tampilan utama Visual Studio Code (VS Code) terdiri dari beberapa elemen penting yang memudahkan pengembangan kode. Di sisi kiri, terdapat *Activity Bar* yang berisi ikon untuk berbagai fitur seperti Explorer, *Search*, *Source Control*, *Run* and *Debug*, dan *Extensions*. Bagian tengah adalah Editor Area di mana Anda dapat membuka dan mengedit file. Di bagian atas, terdapat *Command Palette* yang memungkinkan akses cepat ke berbagai perintah dan pengaturan. Di bagian bawah,

terdapat Status Bar yang menampilkan informasi tentang proyek dan editor saat ini, seperti cabang Git yang aktif, status sinkronisasi, dan notifikasi lainnya. Antarmuka ini dirancang untuk memberikan pengalaman pengkodean yang efisien dan terorganisir.

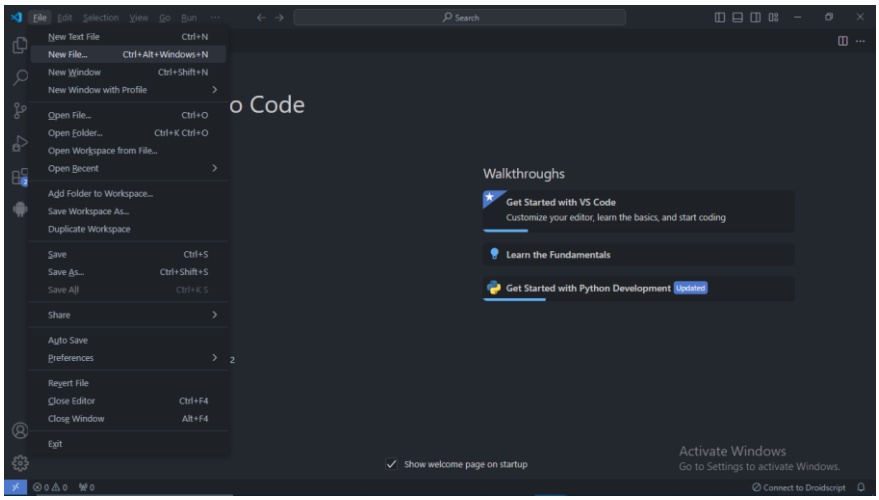


Gambar 4. 8 Tampilan Visual Studio

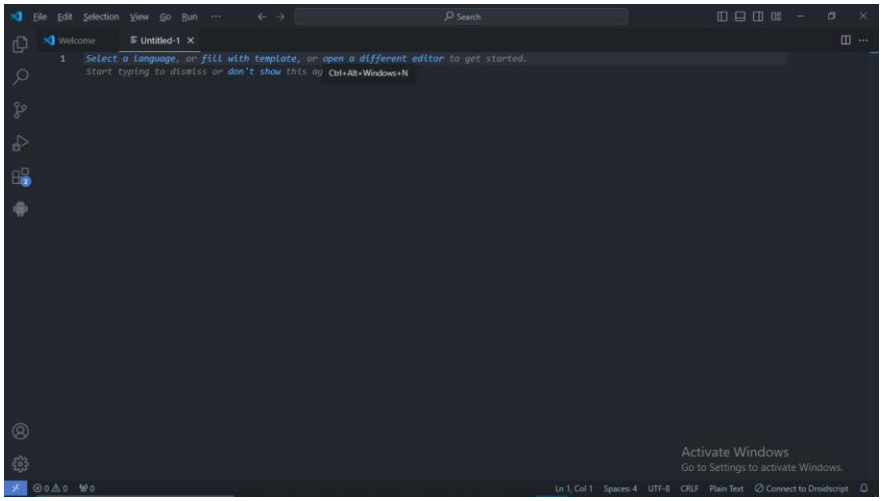
2. Tampilan *New File* Visual Studio

Tampilan "*New File*" di Visual Studio Code sangat intuitif dan mudah digunakan. Ketika Anda memilih opsi untuk membuat *file* baru, Anda akan melihat area editor kosong di tengah layar siap untuk diisi dengan kode Anda. Di bagian atas, terdapat tab baru yang menunjukkan nama *file* sementara seperti "*Untitled-1*" sampai Anda menyimpan *file* tersebut dengan nama yang diinginkan. Sidebar di sebelah kiri tetap menampilkan struktur proyek Anda, memungkinkan Anda untuk dengan mudah menavigasi dan mengelola *file* lain. Anda juga dapat memilih bahasa pemrograman untuk *file* baru ini melalui opsi di bagian bawah jendela atau menggunakan *Command*

Palette untuk akses cepat ke berbagai perintahdan pengaturan



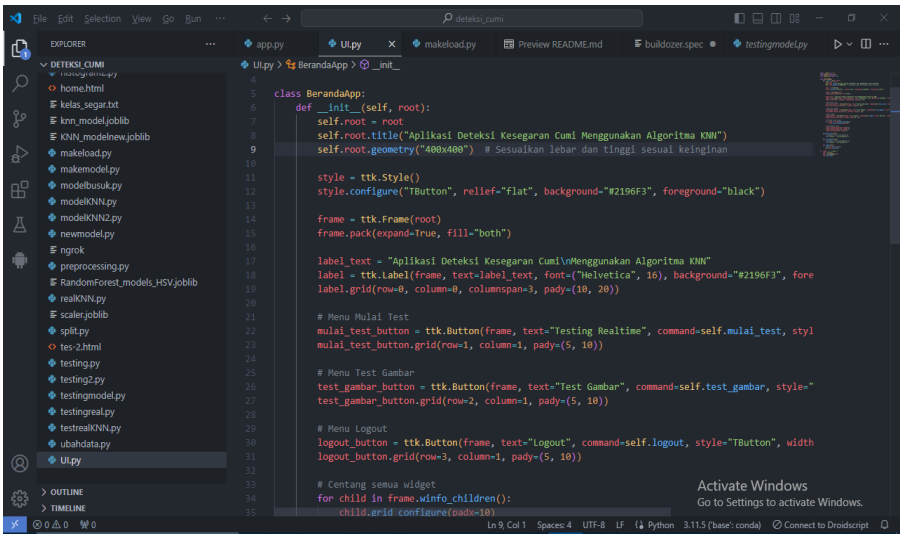
Gambar 4. 9 Tampilan *New File* Visual Studio Code



Gambar 4. 10 Tampilan *Untitled-1* atau *file* sementara

3. Tampilan *Scripct* Menu Utama

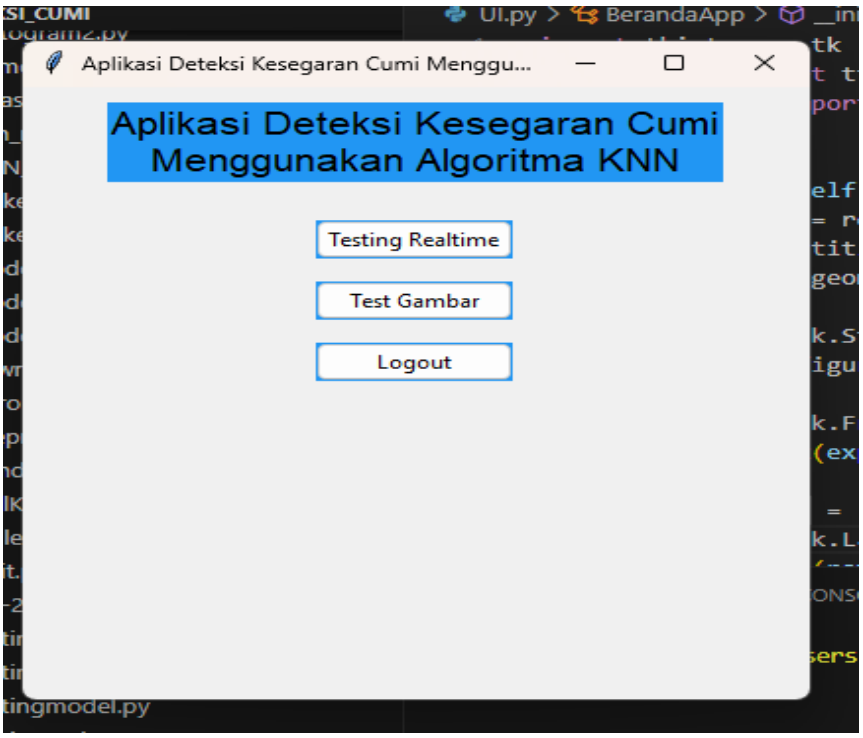
Merupakan Tampilan *Script* Menu Utama yang terdiri 2 bagian yaitu menu realtime, dan upload image,



Gambar 4. 11 Tampilan *Scripct* Menu Utama

4. Tampilan Menu Utama

Tampilan pada aplikasi Deteksi Kesegaran Cumi Menggunakan Algoritma KKN terdiri dari tiga menu utama, yaitu Testing Real-Time, Test Gambar, dan Logout. Pada menu Testing Real-Time, pengguna dapat membuka secara langsung dan dapat mendeteksi cumi secara reel. Menu Test Gambar memungkinkan pengguna mengunggah foto cumi, yang kemudian dianalisis menggunakan teknik pengolahan citra untuk menentukan tingkat kesegarannya. Hasil analisis dari kedua metode ditampilkan secara jelas dengan indikator visual dan deskripsi. Sementara itu, menu Logout berfungsi untuk keluar dari aplikasi guna menjaga keamanan dan privasi data pengguna

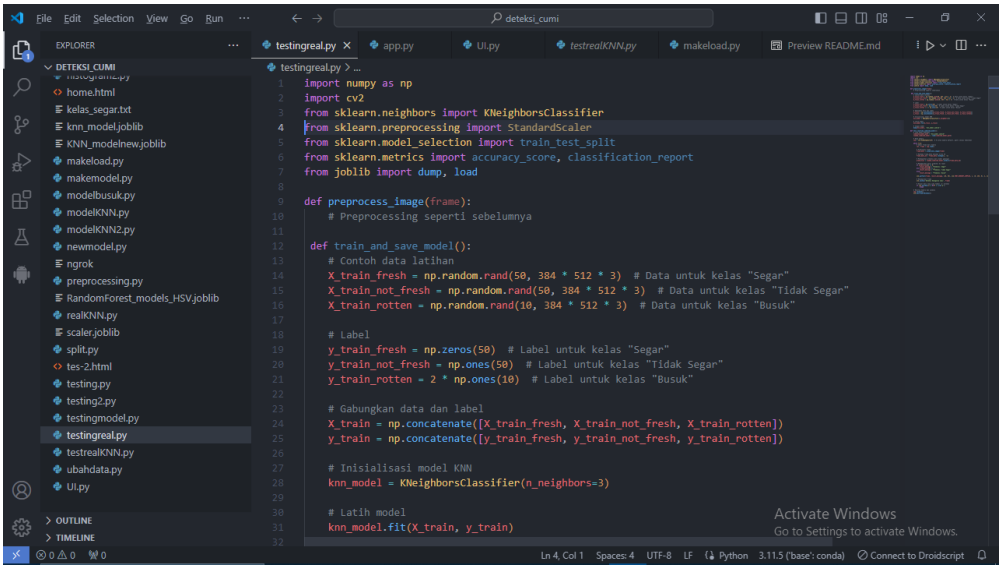


Gambar 4. 12 Tampilan Menu Utama

5. Tampilan Scripct Menu *Realtime*

Meruapakan tampilan realtime menggambarkan alur pengujian yang dilakukan secara langsung pada aplikasi yang sedang berjalan. Ini mencakup diagram alur pengujian yang menunjukkan langkah-langkah dari input data hingga hasil akhir, komponen aplikasi yang diuji seperti antarmuka pengguna dan *backend*, metode

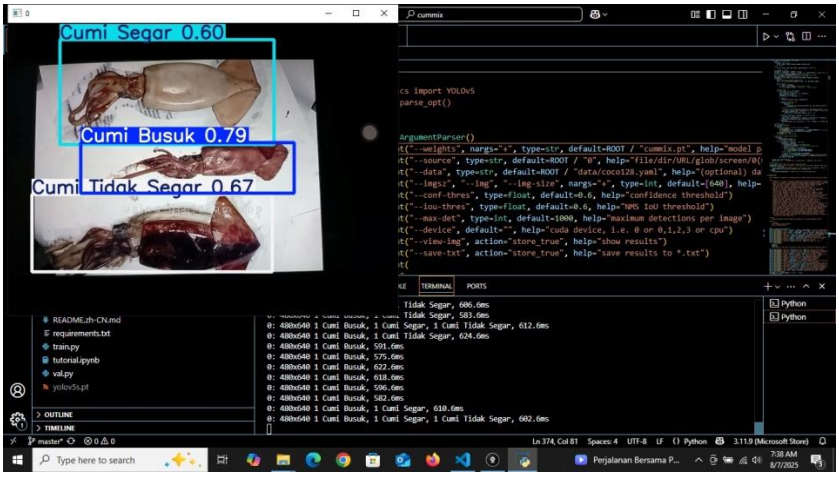
pengujian yang digunakan seperti *black box* atau *white box testing*, hasil pengujian berupa data atau log yang menunjukkan kinerja aplikasi,. Gambar ini membantu memvisualisasikan proses pengujian dan memastikan aplikasi berfungsi sesuai harapan.



Gambar 4. 13 Tampilan Scripct Testing Realtime

6. Tampilan Menu *Realtime*

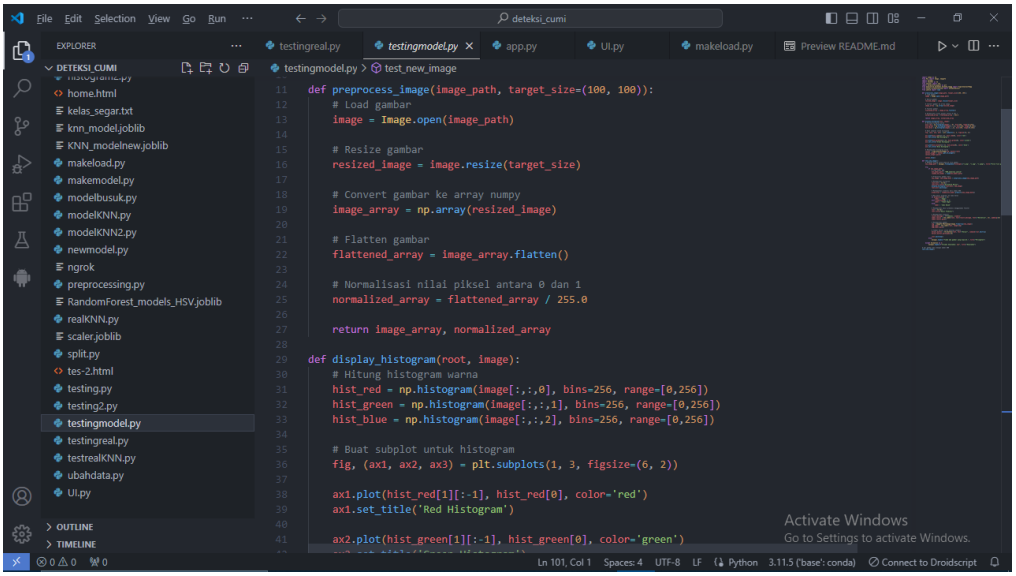
Merupakan Tampilan *Testing Realtime* atau secara langsung menggunakan kamera dalam aplikasi pendeteksi kesegaran cumi-cumi bia menampilkan antarmuka pengguna yang memungkinkan pengguna untuk mengambil gambar cumi-cumi secara langsung melalui kamera. Setelah gambar diambil, script akan menjalankan analisis realtime untuk menilai kesegaran cumi-cumi berdasarkan parameter warna. Hasil analisis ini kemudian ditampilkan akan menunjukkan tingkat kesegaran cumi-cumi tersebut. Berikut adalah gambar hasil pengujiannya.



Gambar 4. 14 Tampilan pengujian

7. Tampilan Script Menu Test Gambar

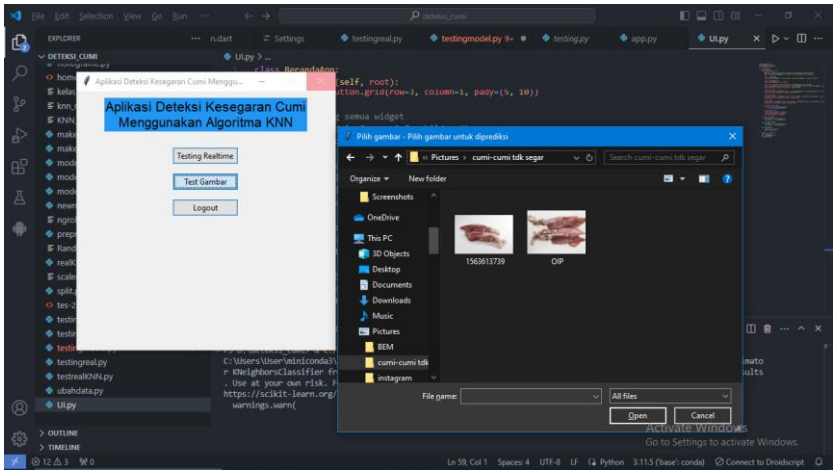
Script menu test dalam aplikasi pendeteksi kesegaran cumi-cumi menampilkan antarmuka pengguna yang memungkinkan pengguna untuk mengunggah gambar cumi-cumi yang akan diuji. Setelah gambar diunggah, script akan menjalankan serangkaian tes untuk menganalisis kesegaran cumi-cumi berdasarkan parameter tertentu seperti warna, tekstur, dan bentuk. Hasil analisis ini kemudian ditampilkan dalam bentuk grafik atau tabel yang menunjukkan tingkat kesegaran cumi-cumi tersebut. Antarmuka ini juga mungkin menyertakan tombol untuk memulai tes, dan mengakses pengaturan aplikasi.



Gambar 4. 15 Tampilan Script Menu Test Gambar

8. Tampilan Menu Test Gambar

a. Tampilan Pilih Gambar



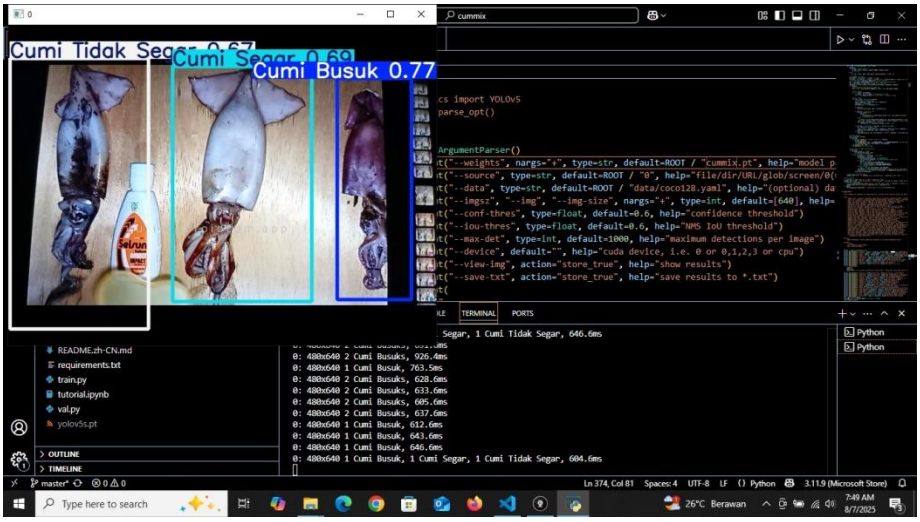
Gambar 4. 16 Tampilan Pilih Gambar

Tampilan menu test gambar dalam aplikasi pendeteksi kesegaran cumi-cumi mencakup beberapa fitur utama seperti pemilihan gambar cumi-cumi yang akan diuji, pengaturan parameter uji, dan hasil analisis. Pengguna dapat mengunggah gambar cumi-cumi yang diambil dari berbagai sudut, kemudian sistem akan memproses gambar tersebut menggunakan algoritma tertentu untuk menilai tingkat kesegaran berdasarkan warna. Hasil analisis akan

ditampilkan dalam bentuk kategori kesegaran, yang memudahkan pengguna untuk memahami kondisi cumi-cumi secara cepat dan akurat. Menu ini juga sering dilengkapi dengan opsi untuk menyimpan hasil uji

b. Tampilan Histogram Frekuensi

Tampilan hasil prediksi pada aplikasi pendeteksi kesegaran cumi menggunakan metode K-Nearest Neighbors (KNN) menampilkan informasi berupa status kesegaran cumi, seperti "Segar" "Tidak Segar," dan “Busuk”. Selain hasil prediksi, aplikasi juga dilengkapi dengan histogram yang menunjukkan distribusi data fitur dari sampel cumi, seperti frekuensi nilai kadar air atau pH pada kategori segar dan tidak segar. Histogram ini membantu pengguna memahami pola data dan visualisasi keputusan yang dibuat oleh model KNN, sehingga meningkatkan transparansi dan kepercayaan terhadap hasil prediksi.



Gambar 4. 17 Pengujian Cumi

9. Proses KNN

Berikut adalah penjelasan setiap tahapan dalam proses K-Nearest Neighbors (KNN) pada aplikasi pendeteksi kesegaran cumi berbasis pengolahan citra:

a. Pengumpulan Data (*Data Collection*)

- Pada tahap ini, citra cumi dari segar,tidak segar sampai busuk dikumpulkan untuk membangun dataset.
- Dataset biasanya terdiri dari gambar yang diambil dalam kondisi standar seperti pencahayaan, jarak kamera, dan sudut pengambilan gambar untuk konsistensi data.

b. Preprocessing Citra

- Konversi ke Grayscale atau Ruang Warna Lain: Gambar dikonversi ke grayscale atau ruang warna agar fitur yang relevan dapat diekstraksi.
- Resizing: Gambar diubah ke ukuran tertentu agar dimensi data seragam.
- Noise Reduction: Teknik seperti Gaussian blur digunakan untuk menghilangkan noise dari gambar.
- Segmentasi: Memisahkan objek cumi dari latar belakang, misalnya dengan metode thresholding atau algoritma segmentasi lain.

c. Ekstraksi Fitur

- Fitur citra yang relevan diambil untuk merepresentasikan karakteristik cumi, seperti:
- Warna: Mengambil nilai rata-rata intensitas warna tertentu (misalnya keabuan, kekuningan, dll.).
- Bentuk: Jika relevan, dapat menggunakan ciri bentuk seperti kontur.
- Fitur-fitur ini kemudian diubah menjadi representasi numerik (vektor fitur) yang akan digunakan oleh algoritma KNN.

d. Pembagian Dataset

- Dataset dibagi menjadi dua bagian:
- Data Latih (Training Data): Untuk membangun model KNN.
- Data Uji (Testing Data): Untuk menguji performa model.

e. Normalisasi Data

- Fitur-fitur yang diekstraksi dinormalisasi agar memiliki rentang nilai yang seragam, misalnya menggunakan Min-Max Scaling atau Z-Score Normalization. Hal ini penting untuk menghindari dominasi fitur tertentu dalam perhitungan jarak.

f. Implementasi Algoritma KNN

- Perhitungan Jarak: Algoritma menghitung jarak antara vektor fitur dari citra uji dengan setiap vektor fitur pada data latih. Metode umum yang digunakan:
- Euclidean Distance salah satu metode perhitungan jarak yang paling umum digunakan dalam algoritma K-Nearest Neighbors (KNN). Jarak ini dihitung menggunakan teorema Pythagoras untuk menentukan seberapa dekat dua titik dalam ruang berdimensi n.
- Manhattan Distance metode perhitungan jarak yang digunakan dalam algoritma K-Nearest Neighbors (KNN). Berbeda dengan Euclidean Distance yang mengukur jarak lurus (garis diagonal), Manhattan Distance menghitung jarak berdasarkan pergerakan horizontal dan vertikal saja, seperti pola pergerakan di jalanan kota (grid-based).
- Pemilihan K: Nilai K (jumlah tetangga terdekat) dipilih. Biasanya diuji beberapa nilai K untuk mencari yang paling optimal.

- Voting: Berdasarkan K tetangga terdekat, kategori mayoritas (segmen segar,tidak segar,dan cumi busuk) ditentukan.

g. Klasifikasi

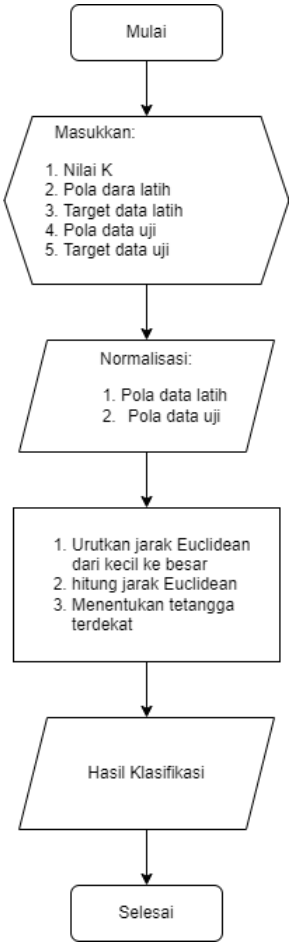
- Setelah voting dilakukan, algoritma akan memberikan label pada citra uji, misalnya Segar,Tidak Segar dan cumi busuk, berdasarkan kelas mayoritas tetangga terdekat.

h. Evaluasi Model

- Model diuji dengan data uji, dan performanya dievaluasi menggunakan metrik seperti:
- Akurasi (*Accuracy*): Rasio prediksi benar terhadap jumlah total data uji.
- *Precision* dan *Recall*: Untuk mengevaluasi kesalahan positif dan negatif.
- *Confusion Matrix*: Untuk melihat distribusi prediksi benar dan salah.

i. Implementasi dalam Aplikasi

- Model KNN yang sudah terlatih diintegrasikan ke dalam aplikasi pendeteksi kesegaran cumi.
- Aplikasi menerima input citra, memprosesnya, mengekstrak fitur, dan menggunakan model KNN untuk memberikan hasil klasifikasi secara real-time.



Gambar 4. 18 Flowchart Proses KNN

D. Implemntasi

Implementasi *system* merupakan tahap penerapan dari suatu teknologi yang di desain untuk siap di operasikan. Tahap ini merupakan terjemahan perancangan dari bab hasil analisis sebelumnya dalam suatu Bahasa pemrograman. Bahasa pemrograman yang digunakan dalam membangun sebuah aplikasi ini adalah bahasa *Python* dengan menggunakan *text editor visual studio code*

1. Kebutuhan Perangkat Keras

Spesifikasi minimum perangkat keras sebagai berikut :

Tabel 4. 1 Kebutuhan Perangkat Keras

Jenis	Spesifikasi
Laptop/PC	*ACER*
Processor	Intel(R) Celeron(R) CPU N3350 @ 1.10GHz 2.10
Memory	4.00 GB

2. Kebutuhan Perangkat Lunak.

Spesifikasi minimum perangkat lunak sebagai berikut:

Tabel 4. 2 Spesifikasi minimum perangkat lunak

Jenis	Spesifikasi
Sistem Operasi	*Windows 10*
Aplikasi	*Visual Studio Code*
Text Editor	*python*

E. Pengujian Sistem

1. BlackBox

Pengujian *system* dilakukan dengan cara pengujian *BlackBox. Blackox* adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi tanpa mengintip ke dalam struktur atau cara kerja internalnya. Metode pengujian ini dapat diterapkan secara *virtual* ke setiap tingkat pengujian perangkat lunak : unit, integrasi, *system*, dan penerimaan.

Tabel 4. 3 Hasil *BlackBox test*

Nama Pengujian	Tujuan	Skenario	Hasil Yang diharapkan	Hasil
Membuka Aplikas	Mengetahui aplikasi berjalan tanpa *error*	Mencoba Membuka aplikasi	Aplikasi Dapat berjalan dengan baik, Menampilkan	✓

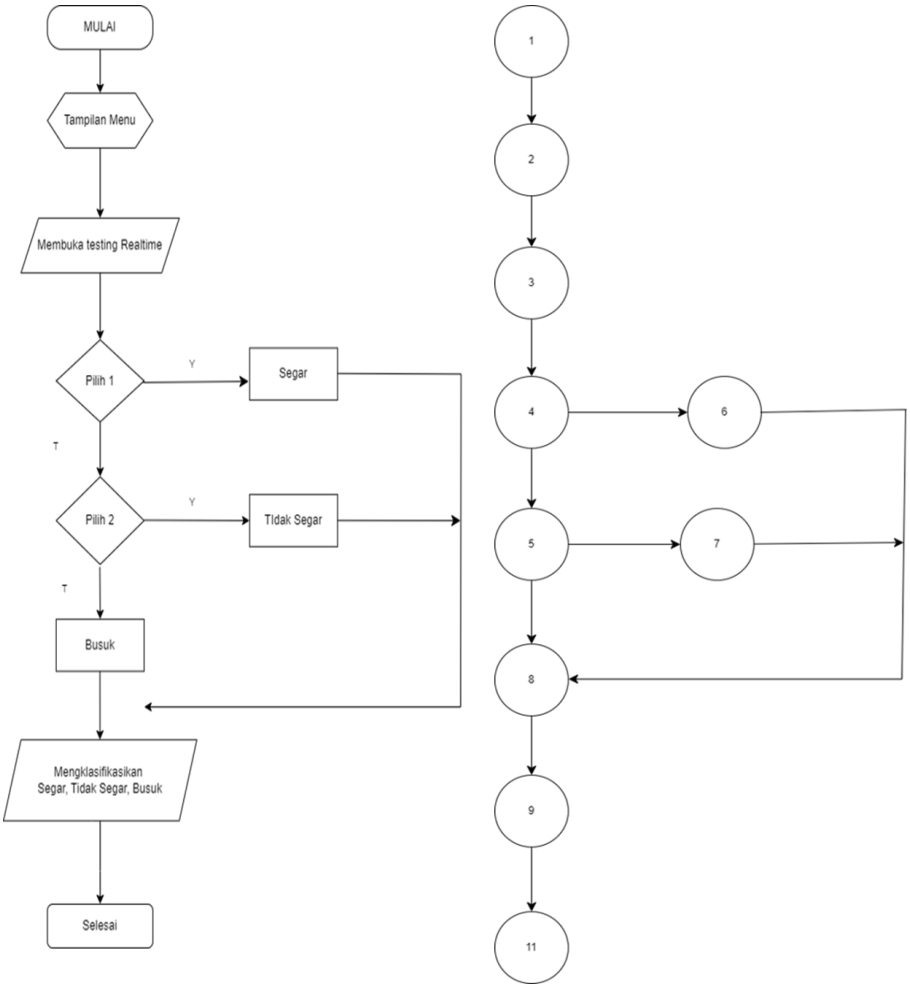
			menu utama	
Menu Utama	Mengakses menu pada aplikasi	Memilih menu	Menampilkan menu testing realtime dan test gambar	✓
Menu Testing Realtime	Untuk mendeteksi objek secara langsung	Memilih menu testing realtime	Menampilkan hasil deteksi kesegaran	✓
Menu Test Gambar	Untuk mendeteksi gambar	Memilih Menu Test Gambar	Menampilkan Hasil deteksi berupa histogram frekuensi dan hsil prediksi	✓

2. WhiteBox

Pengujian ini dilakukan dengan cara pngujian *WhiteBox*. *WhiteBox* adalah metode pengujian perangkat lunak yang menguji struktur internal atau cara kerja aplikasi yang bertentangan dengan fungsinya. Dalam pengujian kotak putih, perspektif internal sistem digunakan untuk merancang kasus uji.

a. *White Box realtime detection*

1) *Flowchart* & Flowgraph



Gambar 4. 19 *Flowchart & Flowgraph Realtime Detection*

Flowchart realtime detection pada aplikasi pendeteksi kesegaran cumi berbasis pengolahan citra dengan metode K-Nearest Neighbors (KNN) menggambarkan alur pengujian sistem secara langsung saat menerima input gambar cumi. Proses dimulai dengan pengambilan citra menggunakan kamera, lalu citra tersebut melalui tahap preprocessing seperti konversi warna, normalisasi, dan ekstraksi fitur. Setelah itu, fitur yang diperoleh dibandingkan dengan dataset menggunakan algoritma KNN untuk menentukan tingkat kesegaran cumi berdasarkan nilai kemiripan. Hasil klasifikasi kemudian ditampilkan kepada pengguna dalam

bentuk label kesegaran. Flowchart ini berfungsi untuk memastikan bahwa setiap tahapan berjalan sesuai spesifikasi, mengidentifikasi potensi kesalahan, serta mengevaluasi performa sistem dalam kondisi penggunaan sebenarnya.

Dari Flowgraph aktivitas diatas dapat dccxgilakukan proses perhitungan sebagai

berikut:

1. Menghitung *Cyclomatic Complexcity* V(G) dari Egde dan Node:

   Dengan rumus : V(G) = E – N + 2

   E (edge) = 11

   N (Node) = 9

   P (Predikat Node) = 3

   Penyelesaian : V(G) = E – N + 2

   = 11 – 9 + 2

   = 4

   Predikat (P) = P + 1

   = 3 + 1

   = 4

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari Flowgraph diatas memiliki *Region* = 4
3. *Independent path* pada Flowgraph diatas adalah:

   Path 1 = 1 – 2 – 3 – 4 – 8 – 11

   Path 2 = 1 – 2 – 3 – 5 – 9 – 11

   Path 3 = 1 – 2 – 3 – 5 – 6 – 10 – 11

   Path 4 = 1 – 2 – 3 – 5 – 6 – 7 – 11

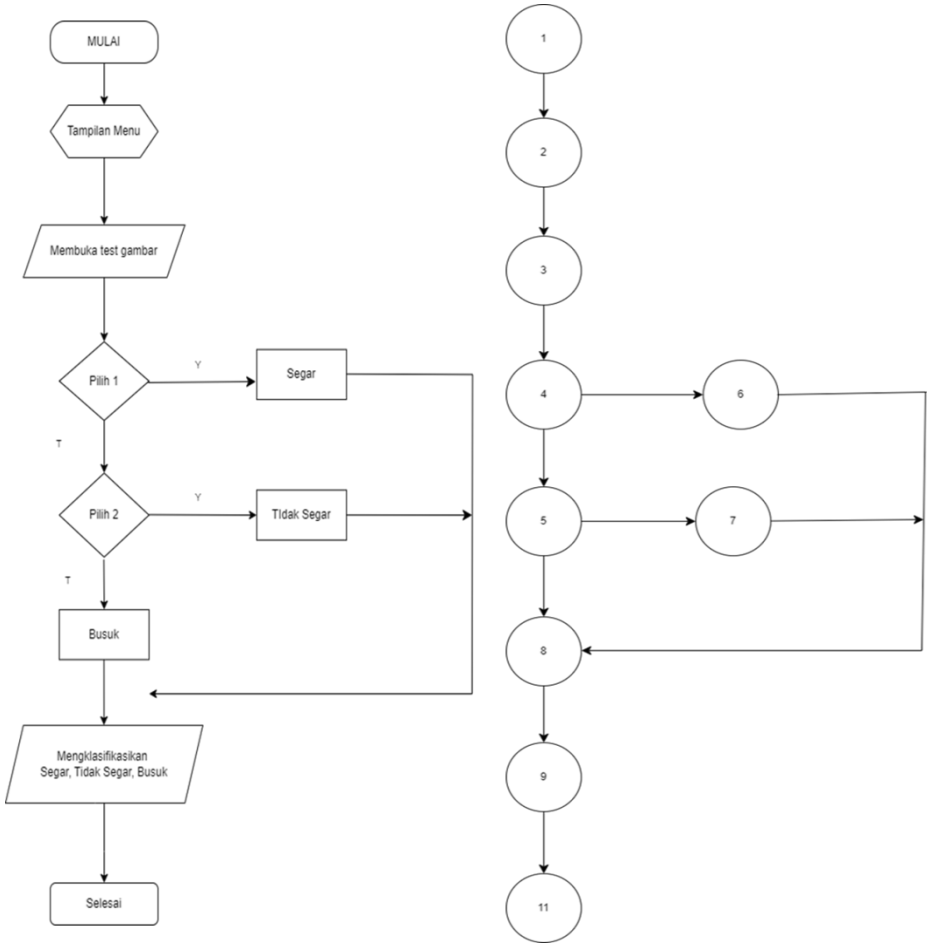
4.Grafik matriks halaman menu *testing realtime*

Tabel 4. 4 Grafik matriks *testing realtime*

	1	2	3	4	5	6	7	8	9	10	11	E-1
1		1										1 – 1 = 0
2			1									1 – 1 = 0
3				1	1							2 – 1 = 1
4								1				1 – 1 = 0
5						1			1			2 – 1 = 1
6							1					1 – 1 = 1
7										1		1 – 1 = 0
8										1		1 – 0 = 0
9										1		1 – 0 = 0
10										1		1 – 0 = 0
11												0
SUM (E + 1)												4 + 1 = 5

b. *White Box* test gambar

1) Flowchart & Flowgrap Upload image



Gambar 4. 20 *Flowchart & flowgrap Upload image*

Flowchart Upload Image dalam aplikasi pendeteksi kesegaran cumi berbasis pengolahan citra dengan metode K-Nearest Neighbors (KNN) menggambarkan proses pengujian sistem menggunakan gambar uji. Alur dimulai dengan pengguna mengunggah atau mengambil gambar cumi, yang kemudian diproses melalui tahap preprocessing seperti konversi warna, peningkatan kualitas citra, dan ekstraksi fitur. Setelah fitur diperoleh, sistem membandingkannya dengan dataset referensi menggunakan algoritma KNN untuk menentukan tingkat kesegaran berdasarkan kemiripan pola. Hasil klasifikasi kemudian ditampilkan dalam bentuk informasi kesegaran cumi, seperti "Segar" atau "Tidak Segar."dan “Busuk” Flowchart ini digunakan untuk menguji keakuratan

sistem dalam mengenali gambar uji, memastikan setiap tahapan berjalan dengan baik, serta mengidentifikasi potensi kesalahan dalam proses deteksi.

Dari Flowgraph aktivitas diatas dapat dccxgilakukan proses perhitungan sebagai

berikut:

1. Menghitung Cyclomatic Complexcity V(G) dari Egde dan Node:

   Dengan rumus : V(G) = E – N + 2

   E (edge) = 11

   N (Node) = 9

   P (Predikat Node) = 3

   Penyelesaian : V(G) = E – N + 2

   = 11 – 9 + 2

   = 4

   Predikat (P) = P + 1

   = 3 + 1

   = 4

2. Berdasarkan perhitungan Cyclomatic Complexcity dari Flowgraph diatas memiliki Region = 4

3. Independent path pada Flowgraph diatas adalah:

   Path 1 = 1 – 2 – 3 – 4 – 8 – 11

   Path 2 = 1 – 2 – 3 – 5 – 9 – 11

   Path 3 = 1 – 2 – 3 – 5 – 6 – 10 – 11

   Path 4 = 1 – 2 – 3 – 5 – 6 – 7 – 11

4. Grafik matriks halaman menu testing realtime

Tabel 4. 5 Grafik matriks test gambar

	1	2	3	4	5	6	7	8	9	10	11	E-1
1		1										1 – 1 = 0
2			1									1 – 1 = 0
3				1	1							2 – 1 = 1
4								1				1 – 1 = 0
5						1			1			2 – 1 = 1
6							1					1 – 1 = 1
7										1		1 – 1 = 0
8										1		1 – 0 = 0
9										1		1 – 0 = 0
10										1		1 – 0 = 0
11												0
SUM (E + 1)												4 + 1 = 5

3. Data Pengujian

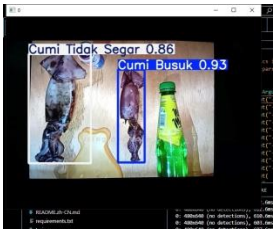
Data pengujian adalah kumpulan data yang digunakan untuk mengevaluasi kinerja dan akurasi suatu model atau sistem setelah proses pelatihan selesai. Data ini bersifat terpisah dan tidak digunakan selama tahap pelatihan, sehingga dapat membrikan gambaran objektif mengenai kemampuan model dalam mengolah infotmasi baru yang belum pernah ditemui sebelumnya.

Tabel 4. 6 Tabel Pengujian

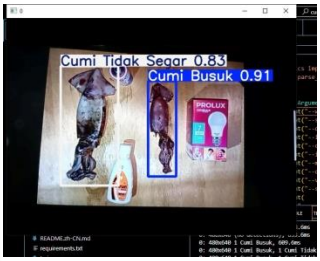
NO	Data Uji	Hasil	keterangan
1			Terdeterksi cumi tidak segar dan cumi busuk
2			Terdeteksi cumi segar dan cumi busuk
3			Terdeteksi cumi tidak segar, cumi segar dan cumi busuk
4			Terseteksi cumi segar dan cumi busuk
5			Terdeteski cumi busuk dan cumi tidak segar

Tabel 4. 6 Lanjutan Tabel

NO	Data Uji	Hasil	keterangan

6			Tersdeteksi cumi segar, cumi busuk, dan cumi tidak segar
7			Terdeteksi cumi tidak segar dan cumi busuk
8			Terdeteksi cumi tidak, segar cumi, segar dan cumi busuk
9			Terseteksi cumi idak segar dan cumi busuk

Tabel 4. 6 Lanjutan Tabel

NO	Data Uji	Hasil	keterangan
10			Terseteksi cumi tidak segar dan cumi busuk

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan rumusan masalah yang telah dikemukakan, dapat disimpulkan bahwa penelitian ini bertujuan untuk mengembangkan aplikasi yang mampu mendeteksi tingkat kesegaran cumi-cumi menggunakan teknik pengolahan citra dengan metode *K-Nearest Neighbour* (KNN). Dengan pendekatan ini, aplikasi diharapkan dapat mengklasifikasikan kesegaran cumi-cumi secara akurat berdasarkan karakteristik visualnya, sehingga dapat membantu konsumen dan pelaku usaha dalam memastikan kualitas produk secara lebih efisien dan objektif. Aplikasi ini berhasil mengidentifikasi tingkat kesegaran cumi berdasarkan fitur-fitur tertentu, seperti warna kulit. Secara keseluruhan, penelitian ini menunjukkan potensi penerapan teknologi pengenalan pola dalam industri perikanan, terutama dalam menjamin kualitas produk yang dihasilkan. Ke depan, pengembangan lebih lanjut dapat dilakukan untuk memperluas cakupan deteksi kesegaran pada jenis seafood lainnya.

B. Saran

Pada penelitian ini penulis menyadari bahwa masih ada bebrapa kekurangan yang sangat diperlukan perbaikan dan pengembangan di penelitian selanjutnya. Oleh karena itu, penulis memiliki beberapa saran untuk pengembangan selanjutnya, sebagai berikut:

1. Penelitian selanjutnya perlu menambah tampilan ataupun menu agar lebih menarik.

2. aplikasi yang penulis buat masih sederhana, terutama tampilan dan fitur yang masih kurang, sehingga untuk pengembangan berikutnya dapat diperbaharui tampilan lebih menarik dan lebih enteraktif.

DAFTAR PUSTAKA

Hasyim, Ade Hastuty. (2023). Dasar Pemprograman (Edisi ke-3). CV. Bangun Bumitama

Azis,N & Pribadi,G & Nurcahya, M. S.(2020) Analisis dan Perancangan Aplikasi Pembelajaran Bahasa Inggris Dasar Berbasis Android. *Jurnal IKRH-ITH informatika. (4)3:1-2*

Sulaiman, N. M. & Hastuty, A. & Yunus, M. (2024) Aplikasi Pembelajaran Bahasa Korea Selatan. *Jurnal Sintaks Logika* .3(3) 1-2

Febrian, & Hastuty, A. (2023) Penerapan Algoritma Vrnam Chiper Pada File Transfer Protocol Server Berbasis PHP. *Jurnal Sintaks Logika. 3(3) 1-2*

Syam, M. N. A. & Hastuty, A. & Wahyuddin. (2024) Penerapan Ant Colony Optimization Untuk Optimasi Pembuatan Jadwal Kuliah. *Jurnal Sintaks Logika 3(3)1-2*

Pravista, D.S. & Santoni, M.M. & Chamidah,N. (2023). Pengolahan Citra Digital Mengunakan OPENCV-PYTHON. Sleman:Deepublish

Cholil, R. S. & Handayani, T. & Prathivi, R. & Ardianita, T. (2021). Implementasi Algoritma Klasifikasi K-Nearest Neighbor (KNN) Untuk Klasifikasi Seleksi Penerima Beasiswa. *IJCIT (Indonesian Journal on Computer and Information Technology).*6(2): 118-127

Mulyono, & Imron, M. & Hestirianoto, T. & Kholilullah, I. & Prasetiyo, L. S. (2023). Efektivitas Atraktor Cumi-Cumi Di Lombok Timur, Nusa Tenggara Barat. 14(1) 1-4

Selao, A. & Sutriani. (2021). Aplikasi Pengolahan Citra Sebagai Media Pengenalan Batik Nusantara. *Jurnal Sintaks Logika.*1(3) 1-2

Jaya, I. & Ramdahan, K.D. (2016). Aplikasi Metode Akustik Untuk Uji Kesegaran Ikan. *Buletin Teknologi Hasil Perikanan.*6(2) 1-3

Syafar, A.M. & Baharuddin, M. S. I. A. & Zainal. M. & Fuadi, N. (2021). Peramalan Penderita Diabetes Menggunakan Algoritma KNN. *Jurnal Sintaks Logika.* 1(2) 113-114

Pratama, G., (2021). Visual Studio Code Untuk Pemula.Semarang:BuildWith Angga.

Sukatmi, & Pitri, E. S., (2018). Aplikasi Absensi Siswa Berbasis Web Dengan Dukungan Sms Gateway Pada Smk Kridawisata Bandar Lampung. *Jurnal Informasi Dan Komputer.* 6(1):23-24

Puspitasari, K. R. & Ahcmad, F. (2019). Dinamika Populasi Cumi-Cumi (Loligo sp.) di Perairan Teluk Banten, Provinsi Banten. *Journal of tropical fisheries management.* 3(2):36-38

Styorini,W. Pratiwi,A. & Widiasari, C. (2022). Identifikasi Tingkat Kesegaran Ikan Berbasis Android. *Jurnal Amplifier.*12(1):12-14.

Dwiyatno, S. Iksal, & Nugraha, S. (2018). Alat Pendeteksi Kesegaran Ikan Menggunakan Metode K-Nearest Neighbour Berdasarkan Warna Mata Berrbasis Atmega 328. *jurnal Prosisko.* 5(2):128-129.

Jerandu, C. Y. Batarius, P. & Sinlae, A. A (2022) . Identifikasi Kualitas Kesegaran Ikan Menggunakan Algoritma K-Nearest Neighbour Berdasarkan Ekstraksi Ciri Warna Hue, Saturation, dan Value (HSV). *ResearchGate.* 4(3):1539.

Wulandari, S. (2022) *Aplikasi Pengolahan Citra Sebagai Media Pengenalan Jenis Motif Ukiran Hias Toraja*. Skripsi tidak di publikasikan. Parepare: Universitas Muhammadiyah Parepare.