

RESEARCH ARTICLE

Perancangan dan Implementasi Aplikasi Android FishQ untuk Identifikasi Kesegaran Ikan

Rabby Fitriana Adawiyah , Ledy Novamizanti* and Suryo Adhi Wibowo

Fakultas Teknik Elektro, Universitas Telkom, Bandung, 40257, Jawa Barat, Indonesia

*Corresponding author: ledyaldn@telkomuniversity.ac.id

Abstrak

Penelitian ini mengembangkan aplikasi Android FishQ yang memanfaatkan model *deep learning* YOLOv8 untuk mengidentifikasi kesegaran ikan dengan tujuan meningkatkan kualitas dan keamanan produk perikanan. Metode konvensional untuk menilai kesegaran ikan memerlukan waktu dan keahlian khusus, sementara FishQ menawarkan solusi digital yang cepat dan efisien. Aplikasi ini dirancang menggunakan *Android Studio* dengan bahasa pemrograman Kotlin, serta diintegrasikan dengan *Google Cloud Platform* untuk pemrosesan gambar secara *real-time*. Model YOLOv8 dilatih menggunakan dataset gambar ikan yang telah dianotasi, dan di-host di *Cloud Run* untuk skalabilitas dan keandalan. Pengujian aplikasi menggunakan metode *System Usability Scale* (SUS) menunjukkan nilai rata-rata 86,87, yang mengindikasikan tingkat penerimaan pengguna yang tinggi dan menempatkan aplikasi dalam kategori "acceptable" dengan *grade A*. Hasil ini menunjukkan bahwa FishQ tidak hanya akurat dalam mengidentifikasi kesegaran ikan, tetapi juga userfriendly, sehingga memiliki potensi besar untuk diadopsi dalam industri perikanan.

Key words: Aplikasi Android, Kesegaran Ikan, Identifikasi Ikan.

Pendahuluan

Kesegaran ikan adalah salah satu faktor penting dalam industri perikanan untuk memastikan kualitas dan keamanan produk yang dikonsumsi oleh masyarakat [1]. Metode konvensional, seperti inspeksi visual dan pengujian laboratorium, seringkali memerlukan waktu yang cukup lama dan keahlian khusus untuk menentukan kesegaran ikan. [2]. Dengan perkembangan teknologi, solusi berbasis digital telah menjadi perhatian utama dalam meningkatkan efisiensi dan akurasi proses identifikasi kesegaran ikan. Teknologi *deep learning* telah menunjukkan potensi yang besar dalam bidang pengenalan citra, termasuk dalam identifikasi kesegaran ikan [3]. *Deep learning*, khususnya *Convolutional Neural Network* (CNN), mampu mengenali pola dan fitur dalam gambar dengan akurasi tinggi [4]. Penggunaan model *deep learning* yang di-host pada *cloud* juga menawarkan keunggulan dalam hal kecepatan pemrosesan dan skalabilitas, memungkinkan aplikasi untuk menangani jumlah data yang besar dengan cepat dan efisien [5].

Namun, masih terdapat tantangan dalam mengintegrasikan teknologi ini ke dalam aplikasi yang mudah digunakan oleh pengguna awam, terutama dalam konteks industri perikanan. Aplikasi yang kompleks dan tidak *user-friendly* dapat menghambat adopsi teknologi ini di lapangan. Oleh karena itu, diperlukan solusi yang tidak hanya akurat dan efisien tetapi juga mudah diakses dan digunakan oleh pengguna sehari-hari [6] & [7]. FishQ dikembangkan sebagai aplikasi Android yang bertujuan untuk mengatasi masalah ini dengan menyediakan *platform* yang intuitif dan mudah digunakan untuk mengidentifikasi kesegaran ikan. Tujuan

dari penelitian ini adalah untuk mengembangkan dan mengimplementasikan aplikasi Android FishQ yang dapat mengidentifikasi kesegaran ikan yang dapat diterima oleh pengguna dengan mengevaluasi kinerja aplikasi menggunakan metode SUS (*System Usability Scale*) [8].

Tinjauan Pustaka

Pada bagian kajian teori disajikan teori yang berkaitan dengan penelitian yaitu tentang *android development*, *cloud integration*, *deep learning*, dan identifikasi kesegaran ikan.

Android Development

Pengembangan aplikasi Android melibatkan pemahaman mendalam tentang beberapa prinsip dasar, termasuk penggunaan *Android Studio*, bahasa pemrograman Kotlin, dan desain antarmuka pengguna menggunakan perangkat lunak Figma. *Android Studio* merupakan *Integrated Development Environment* (IDE) yang dirancang khusus untuk memfasilitasi pembuatan aplikasi Android. IDE ini menyediakan berbagai fitur yang mendukung pengembangan aplikasi, mulai dari penulisan kode, debugging, hingga pengujian aplikasi. Kotlin, sebagai bahasa pemrograman resmi untuk pengembangan aplikasi Android, menawarkan sintaksis yang ringkas dan ekspresif, serta kompatibilitas yang baik sehingga memungkinkan pengembang untuk memanfaatkan ekosistem Java Android yang sudah mapan [9]. Pemilihan Kotlin sebagai bahasa utama dalam pengembangan aplikasi FishQ memungkinkan untuk

meningkatkan produktivitas dan keamanan kode, serta memanfaatkan fitur-fitur *modern* dalam pengembangan aplikasi. Selain itu, desain antarmuka pengguna menjadi aspek kunci dalam pengalaman pengguna aplikasi Android. Penggunaan perangkat lunak desain seperti Figma mempermudah dalam merancang antarmuka yang intuitif dan menarik secara visual [10]. Figma memungkinkan kolaborasi tim yang efektif dalam proses desain, menyediakan berbagai komponen *User Interface* (UI) yang dapat digunakan ulang, dan menghasilkan prototipe interaktif yang membantu dalam menguji dan merancang pengalaman pengguna yang optimal sebelum aplikasi dibangun secara lengkap.

Cloud Integration

Integrasi dengan layanan *cloud* merupakan komponen krusial dalam pengembangan aplikasi Android, terutama ketika aplikasi tersebut membutuhkan pemrosesan data yang intensif seperti model *deep learning* [11]. Dalam konteks aplikasi FishQ, pemrosesan gambar untuk identifikasi kesegaran ikan dilakukan melalui model *deep learning* YOLOv8 yang di-host di *Google Cloud Platform* (GCP) menggunakan layanan *Cloud Run*. *Cloud Run* adalah layanan komputasi terkelola yang secara otomatis menskalakan aplikasi *containerized*. Dengan menggunakan *Cloud Run*, model *deep learning* dapat *deploy* dalam *container*, memungkinkan pengelolaan yang lebih mudah dan efisien serta meningkatkan skalabilitas dan keandalan [12].

Hal ini memastikan bahwa model dapat menangani volume permintaan yang tinggi dari berbagai pengguna aplikasi FishQ tanpa mengalami penurunan kinerja [13]. Dari perspektif pengembangan aplikasi Android, integrasi dengan layanan *Cloud Run* di GCP dilakukan dengan mengkonfigurasi URL API yang disediakan oleh *Cloud Run* setelah model *deep learning* di-*deploy* [14]. Aplikasi FishQ memanfaatkan URL API ini untuk mengirim permintaan berupa gambar ikan yang diambil oleh pengguna. Gambar tersebut kemudian diproses oleh model *deep learning* di *cloud*, dan hasil pemrosesan yang berupa informasi mengenai kesegaran ikan dikembalikan ke aplikasi dan ditampilkan kepada pengguna. Proses ini melibatkan penggunaan *library* seperti *Retrofit* dan *OkHttp* untuk memfasilitasi komunikasi HTTP antara aplikasi dan *cloud*.

Deep Learning

Penggunaan *deep learning* dalam aplikasi Android FishQ didasarkan pada model *You Only Look Once version 8* (YOLOv8), yang merupakan salah satu model *deep learning* terkini untuk deteksi objek [15]. YOLOv8 dikenal karena kemampuannya dalam mendeteksi objek secara *real-time* dengan akurasi yang tinggi [16]. Model ini dilatih menggunakan dataset ikan cakalang, di mana setiap gambar ikan dalam dataset tersebut diberi *label* "segar" atau "tidak segar". YOLOv8 adalah model *deep learning* untuk deteksi objek yang bekerja dengan cara memprediksi *bounding boxes* dan *label* kelas dari objek yang ada dalam sebuah gambar melalui satu pass dari *Convolutional Neural Network* (CNN) [17]. Keunggulan utama YOLOv8 adalah efisiensinya dalam mendeteksi objek secara cepat dan akurat, yang sangat penting dalam aplikasi seperti FishQ.

Dalam konteks aplikasi FishQ, model YOLOv8 di-host di *cloud* dan diakses melalui API. Pengguna aplikasi dapat mengambil gambar ikan cakalang menggunakan kamera perangkat Android, kemudian gambar tersebut dikirim ke model YOLOv8 melalui API. Model ini kemudian memproses gambar tersebut untuk mendeteksi dan mengklasifikasikan kesegaran ikan berdasarkan dua *label* yaitu "segar" dan "tidak segar". Proses pelatihan model YOLOv8 melibatkan beberapa langkah penting. Pertama, dataset gambar ikan cakalang dikumpulkan dan diberi anotasi dengan *label* kesegaran. Selanjutnya, dataset ini digunakan untuk melatih model YOLOv8, di mana model belajar mengenali fitur-fitur visual yang membedakan ikan segar dari yang tidak segar. Pelatihan model dilakukan menggunakan infrastruktur komputasi di *cloud* untuk

memastikan efisiensi dan kecepatan. Setelah model dilatih, hasilnya di-*deploy* ke layanan *Cloud Run* di GCP. Hal ini memungkinkan model untuk diakses secara efisien oleh aplikasi Android melalui permintaan API. Penggunaan *deep learning* yang di-host di *cloud* memberikan beberapa keuntungan, termasuk pemrosesan yang cepat dan akurat, serta skalabilitas yang tinggi, memungkinkan aplikasi FishQ untuk menangani sejumlah besar permintaan dari pengguna secara simultan.

Identifikasi Kesegaran Ikan

Identifikasi kesegaran ikan merupakan aspek penting dalam industri perikanan untuk memastikan kualitas dan keamanan produk yang dikonsumsi oleh masyarakat. Menurut dokumen Standar Nasional Indonesia (SNI), terdapat beberapa parameter dan indikator utama yang digunakan untuk menentukan kesegaran ikan. Parameter-parameter ini meliputi tekstur, warna, dan bau ikan. Tekstur ikan segar biasanya kenyal dan elastis, di mana daging ikan yang ditekan akan kembali ke bentuk semula. Sebaliknya, ikan yang tidak segar memiliki tekstur yang lembek dan tidak elastis, serta meninggalkan bekas ketika ditekan. Warna ikan segar umumnya cerah dan mengkilap, terutama pada bagian mata dan insang. Mata ikan segar tampak jernih dan menonjol, sedangkan insang berwarna merah cerah. Ikan yang tidak segar akan menunjukkan warna mata yang buram dan insang yang berwarna kecoklatan atau abu-abu. Selain itu, ikan segar memiliki aroma yang khas laut atau air tawar dan tidak berbau amis yang menyengat. Ikan yang tidak segar akan mengeluarkan bau yang tidak sedap, seperti bau busuk atau asam [18][19].

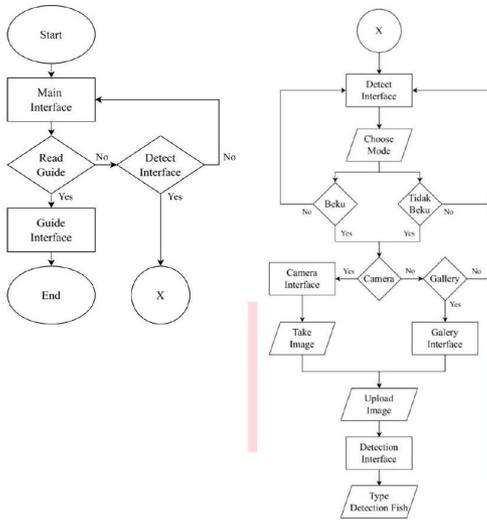
Pada aplikasi FishQ, identifikasi kesegaran ikan dilakukan berdasarkan visual ikan itu sendiri. Menggunakan model *deep learning* YOLOv8 yang dilatih dengan dataset gambar ikan cakalang, aplikasi ini memanfaatkan parameter visual seperti tekstur dan warna untuk menentukan kesegaran ikan. Gambar ikan yang diambil oleh pengguna melalui kamera perangkat Android kemudian dianalisis oleh model YOLOv8 untuk mendeteksi fitur-fitur visual yang mengindikasikan kesegaran ikan. Hasil dari analisis ini kemudian dikirim kembali ke aplikasi dan ditampilkan kepada pengguna, memberikan informasi mengenai apakah ikan tersebut segar atau tidak segar. Dengan pendekatan ini, FishQ menyediakan solusi bagi pengguna untuk mengevaluasi kesegaran ikan secara cepat dan akurat, berdasarkan visual ikan itu sendiri sesuai dengan pedoman SNI.

Metodologi Penelitian

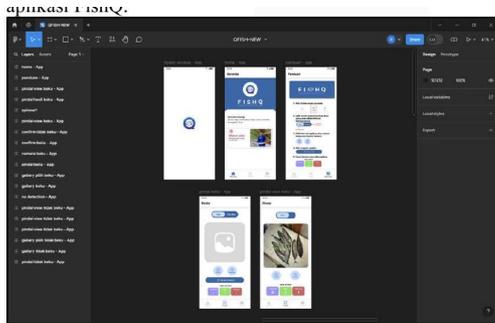
Penelitian ini menggunakan pendekatan pengembangan aplikasi berbasis Android dan integrasi dengan *cloud* untuk implementasi *deep learning* YOLOv8.

Desain Antarmuka Pengguna

Secara umum, aplikasi *mobile* FishQ memiliki bagian utama yaitu antarmuka pengguna, *mode toggle*, pengambilan citra dari kamera atau galeri, pengunggahan citra, dan keluaran hasil deteksi. Antarmuka pengguna menampilkan menu kepada pengguna dalam desain yang intuitif dan mudah digunakan. *Mode toggle* memberikan pilihan antara *mode* "beku" dan "tidak beku" agar pengguna dapat memilih model *deep learning* yang sesuai. Selain mengambil citra ikan secara langsung menggunakan kamera, pengguna juga dapat memilih citra ikan yang sudah tersimpan di galeri. Citra ikan yang telah dipilih, selanjutnya akan dikirimkan oleh aplikasi *mobile* ke *cloud* untuk diproses oleh model *deep learning*. Dengan komponen-komponen tersebut, aplikasi *mobile* dirancang agar dapat melakukan deteksi objek ikan dan menampilkan hasilnya kepada pengguna secara efektif dan mudah dipahami. Gambar 1 merupakan *flowchart* yang memberikan gambaran mengenai alur kerja aplikasi berdasarkan pilihan fitur yang dipilih pengguna. Desain antarmuka pengguna aplikasi FishQ dikembangkan



Gambar 1. Flowchart Aplikasi FishQ.



Gambar 2. Tampilan Figma Project FishQ.



(a) layar pembuka (b) layar utama (c) layar panduan (d) layar deteksi
GAMBAR 3

Gambar 3. Desain Antarmuka Aplikasi.

dengan tujuan untuk menciptakan pengalaman pengguna yang intuitif dan mudah digunakan. Proses desain antarmuka dilakukan menggunakan perangkat lunak desain Figma, yang memungkinkan kolaborasi tim yang efektif dan pembuatan prototipe interaktif. Desain antarmuka terdiri dari beberapa layar yaitu, layar pembuka, layar utama, layar panduan, dan layar deteksi. Gambar 2 berikut merupakan tampilan Figma untuk *project* dari aplikasi FishQ.

Layar pembuka memberikan sambutan dengan menampilkan logo dari FishQ. Pada layar utama ditampilkan informasi singkat tentang aplikasi dan memberikan akses cepat ke fitur utama aplikasi. Layar panduan menyediakan instruksi penggunaan untuk membantu pengguna memahami cara menggunakan aplikasi. Layar deteksi adalah inti dari aplikasi, di mana pengguna dapat mengambil atau mengunggah gambar ikan untuk dianalisis. Halaman deteksi ikan menampilkan antarmuka deteksi yang menunjukkan fungsi pemindaian atau pengambilan citra. *Toggle* "beku" dan "tidak beku" digunakan untuk mengubah mode pemindaian sesuai dengan kondisi ikan yang akan dideteksi. Selain itu, pengguna juga diberikan opsi untuk mengakses galeri atau kamera secara langsung untuk mengunggah citra ikan yang akan dideteksi.

Pengembangan Aplikasi Android

Pengembangan aplikasi FishQ dilakukan menggunakan *Android Studio*, dengan bahasa pemrograman Kotlin untuk *back-end* dan XML sebagai *front-end* yang mendefinisikan tata letak antarmuka pengguna. Fitur utama aplikasi terdapat pada halaman deteksi, di mana pengguna

dapat mengambil gambar ikan menggunakan kamera atau memilih gambar dari galeri. Aplikasi ini dilengkapi dengan fitur *counter* yang menampilkan jumlah ikan yang terdeteksi serta fitur *toggle* mode yang memungkinkan pengguna untuk memilih mode deteksi ikan beku atau tidak beku, sesuai dengan kondisi ikan yang akan dideteksi. Gambar 3 adalah hasil implementasi dari desain aplikasi FishQ yang telah dibuat dalam *Android Studio*.

Cloud Integration

Integrasi *cloud* merupakan komponen penting dalam pengembangan aplikasi FishQ, yang memungkinkan pemrosesan gambar secara efisien menggunakan model *deep learning* YOLOv8 yang di-*host* di GCP. Dari sisi aplikasi Android, integrasi dilakukan dengan mengkonfigurasi URL API yang disediakan oleh *Cloud Run* setelah model YOLOv8 di-*deploy*. Tabel 1 merupakan kode *back-end* yang digunakan untuk menyambungkan aplikasi dengan *cloud* melalui API.

Penggunaan *library* seperti *Retrofit* dan *OkHttp* memfasilitasi komunikasi HTTP antara aplikasi dan server *cloud*. Kode tersebut menyambungkan sisi aplikasi dengan *cloud* yang mencakup pengiriman gambar ikan yang diambil oleh pengguna ke server *cloud* untuk diproses oleh model YOLOv8, serta penerimaan hasil pemrosesan yang kemudian ditampilkan kepada pengguna. Integrasi ini memastikan bahwa

Table 1. Integrasi Aplikasi ke *Cloud*

```

interface ObjectDetectionService \{
    @Multipart
    @POST("img_object_detection")
    fun detectObjects(
        @Query("mode") mode: String,
        @Part imagePart: MultipartBody.Part
    ): Call<ObjectDetectionResponse>
    companion object \{
        operator fun invoke(): ObjectDetectionService \{
            return Retrofit.Builder()
                .baseUrl("https://fishq-lpsxny64fq-
                as.a.run.app/")
                .addConverterFactory(GsonConverterFactory.create())
                .build()
                .create(ObjectDetectionService::class.java)
        }
    }
}

data class ObjectDetectionResponse(
    var total: Int,
    var fresh: Int,
    var no_fresh: Int,
    var image_data: String)

```

Table 2. Pernyataan *System Usability Scale*

No.	Pernyataan
Q1	Saya rasa saya ingin sering menggunakan sistem ini.
Q2	Saya merasa sistem ini rumit.
Q3	Menurut saya sistem ini mudah digunakan.
Q4	Saya rasa saya akan membutuhkan dukungan teknis untuk dapat menggunakan sistem ini.

aplikasi dapat memanfaatkan komputasi *cloud* untuk mengidentifikasi kesegaran ikan.

Pengujian Aplikasi

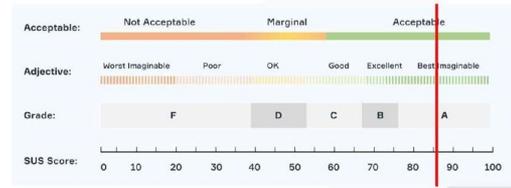
Pengujian aplikasi FishQ dilakukan menggunakan metode SUS untuk mengevaluasi aplikasi dari perspektif pengguna [8]. Pengujian melibatkan 10 *statement* SUS yang mencakup berbagai aspek penggunaan aplikasi, seperti kemudahan penggunaan, efisiensi, dan kepuasan pengguna. Tabel 2 merupakan list pernyataan yang digunakan pada metode SUS. Kuesioner dilakukan terhadap responden yang memiliki perangkat Android. Setiap *statement* dinilai menggunakan skala Likert 5 poin, di mana responden menunjukkan tingkat persetujuan atau ketidaksetujuan mereka terhadap setiap *statement*. Persamaan (1) merupakan rumus untuk menghitung skor total SUS.

$$\begin{aligned}
 \text{Skor total SUS} = & (((Q1 - 1) + (5 - Q2) + (Q3 - 1) + \\
 & (5 - Q4) + (Q5 - 1) + (5 - Q6) + (Q7 - 1) \\
 & + (5 - Q8) + (Q9 - 1) + (5 - Q10)) * 2, 5)
 \end{aligned} \quad (1)$$

Rumus SUS digunakan untuk menghitung skor keseluruhan dari tanggapan pengguna, dengan skor akhir berkisar antara 0 hingga 100.

$$\bar{x} = \sum_n x \quad (2)$$

Rumus (2) digunakan untuk menghitung nilai rata-rata skor SUS dengan menjumlahkan skor kontribusi dari setiap item sehingga nilai rata-rata SUS dari aplikasi FishQ mendapat mendapat nilai 86,87 .

**Gambar 4.** Kategori SUS.

Hasil dan Pembahasan

Hasil Pengujian Aplikasi

Setiap halaman aplikasi diuji untuk memastikan fungsionalitasnya berjalan dengan baik. Pada halaman utama, tombol "Pindai" dan "Panduan" diuji. Kedua tombol tersebut berhasil mengarahkan pengguna ke masing-masing fitur yaitu halaman pindai dan halaman panduan. Pada halaman deteksi, *mode toggle* untuk pilihan jenis ikan "Beku" dan "Tidak Beku" berfungsi dengan baik. Tombol "Galeri" dan "Kamera" juga berhasil diuji. Kedua tombol tersebut dapat mengarahkan pengguna ke galeri lokal dan kamera untuk memilih atau menangkap citra secara langsung kemudian menginputkannya ke aplikasi. Selain itu, tombol "Unggah Gambar" berfungsi dengan baik untuk mengunggah citra dari aplikasi ke *cloud* dan aplikasi dapat menampilkan hasil deteksi kesegaran ikan serta menampilkan jumlah ikan yang terdeteksi. Pengujian aplikasi dengan metode SUS dilakukan dengan hasil seperti yang ditunjukkan Tabel 3 berikut.

Analisis Hasil Pengujian Aplikasi

Nilai SUS yang diperoleh menunjukkan tingkat penerimaan pengguna terhadap aplikasi FishQ yang termasuk dalam kategori *acceptable* dengan *Grade A* seperti yang ditunjukkan pada Tabel 4 dan Gambar 4 [20].

Kesimpulan

Penelitian ini berhasil mengembangkan aplikasi Android FishQ untuk mengidentifikasi kesegaran ikan cakalang menggunakan model *deep learning* YOLOv8 yang di-host di *Google Cloud Platform*. Hasil pengujian menunjukkan bahwa aplikasi FishQ mampu memberikan hasil identifikasi kesegaran ikan serta mendapatkan *feedback* positif dari pengguna mengenai pengalaman penggunaannya. Dengan demikian, aplikasi FishQ dapat memberikan solusi praktis dan efisien untuk meningkatkan kualitas dan keamanan produk perikanan.

Daftar Pustaka

- Duarte AM, Silva F, Pinto FR, Barroso S, Gil MM. Quality Assessment of Chilled and Frozen Fish-Mini Review. *Foods*. 2020 November;9(12):1739. Accessed: Jul. 6, 2024.
- Franceschelli L, Berardinelli A, Dabbou S, Ragni L, Tartagni M. Sensing Technology for Fish Freshness and Safety: A Review. *Sensors*. 2021 February;21(4):1373. Accessed: Jul. 6, 2024.
- Alzubaidi L, Zhang J, Humaidi AJ, et al. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *Journal of Big Data*. 2021 April;8(1):53. Accessed: Jul. 6, 2024.
- Shorten C, Khoshgoftaar TM. A Survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*. 2019 December;6(1):60. Accessed: Jul. 6, 2024.
- Xiangdong T. Image Recognition Algorithm Based on Hybrid Deep Learning. *International Journal of System Assurance Engineering*

Table 3. Hasil Kuesioner SUS

No.	Responden	Result of SUS Score Calculation										Total Score
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	
1	Responden 1	3	2	5	1	5	2	5	1	5	2	87,5
2	Responden 2	5	1	5	1	5	1	5	1	5	1	100
3	Responden 3	4	2	4	2	4	2	4	2	4	2	75
4	Responden 4	5	1	5	1	5	1	5	1	5	1	100
5	Responden 5	4	1	5	1	5	2	5	1	4	1	92.5
...
103	Responden 103	5	2	4	1	5	1	4	2	4	2	85
Nilai Rata-rata											86,87	

Table 4. Grading SUS

SUS Score	Letter Grade
≥ 80,3	A
≤ 74	B
≤ 68	C
≤ 51	D
< 51	E

and Management. 2023. Accessed: Jul. 6, 2024.

6. Wang Y, Cheng W, Sufi F, Fang Q, Mahmoud SS. A Systematic Review of Using Deep Learning in Aphasia: Challenges and Future Directions. *Computers*. 2024 May;13(5):117. Accessed: Jul. 6, 2024.
7. Mendez E, Ochoa O, Olivera-Guzman D, et al. Integration of Deep Learning and Collaborative Robot for Assembly Tasks. *Applied Sciences*. 2024 January;14(2):839. Accessed: Jul. 6, 2024.
8. Sari RP, Henim SR. The Application of System Usability Scale Method to Measure the Usability of Electronic Learning System (E-learning) of Politeknik Caltex Riau. *ILKOM Jurnal Ilmiah*. 2021 December;13(3):266-71. Accessed: Jul. 6, 2024.
9. Mateus BG, Martinez M. An Empirical Study on Quality of Android Applications Written in Kotlin Language. *Empirical Software Engineering*. 2019;24:3356-93. Accessed: Jul. 6, 2024.
10. Muhyidin MA, Sulhan MA, Sevtiana A. Perancangan UI/UX Aplikasi My CIC Layanan Informasi Akademik Mahasiswa Menggunakan Aplikasi Figma. *Jurnal Digital*. 2020 December;10(2):208. Accessed: Jul. 6, 2024.
11. Mungoli N. Scalable, Distributed AI Frameworks: Leveraging Cloud Computing for Enhanced Deep Learning Performance and Efficiency. *arXiv preprint arXiv:230413738*. 2023. Accessed: Jul. 6, 2024. Available from: <https://arxiv.org/abs/2304.13738>.
12. Eshratifar AE, Pedram M. Energy and Performance Efficient Computation Offloading for Deep Neural Networks in a Mobile Cloud Computing Environment. In: *Proceedings of the 2018 Great Lakes Symposium on VLSI*. New York, NY, USA: Association for Computing Machinery; 2018. p. 111-6. Accessed: Jul. 6, 2024.
13. Koubaa A, Ammar A, Alahdab M, Kanhouh A, Azar AT. DeepBrain: Experimental Evaluation of Cloud-Based Computation Offloading and Edge Computing in the Internet-of-Drones for Deep Learning Applications. *Sensors*. 2020;20(18):5240. Accessed: Jul. 6, 2024.
14. Zhang Q, Li X, Che X, Ma X, Zhou A, Xu M, et al. A Comprehensive Benchmark of Deep Learning Libraries on Mobile Devices. In: *Proceedings of the ACM Web Conference 2022*. New York, NY, USA: Association for Computing Machinery; 2022. p. 3298-307. Accessed: Jul. 6, 2024.
15. Sohan M, Ram TS, Reddy CVR. A Review on YOLOv8 and Its Advancements. In: *ICDICI 2023. Algorithms for Intelligent Systems*. Springer, Singapore; 2024. Accessed: Jul. 6, 2024.
16. Miled WS, Akhloufi MA, Asker HB. A Real-Time Deep UAV Detection Framework Based on a YOLOv8 Perception Module. In: *ISPR 2023. Communications in Computer and Information Science*. vol. 1941. Springer, Cham; 2024. Accessed: Jul. 6, 2024.
17. Reis D, Kupec J, Hong J, Daoudi A. Real-time Flying Object Detection with YOLOv8. *arXiv preprint arXiv:230509972*. 2023. Accessed: Jul. 6, 2024. Available from: <https://arxiv.org/abs/2305.09972>.
18. Nasional BS. *Ikan Segar*; 2021. SNI 2729:2021.
19. Badan Standardisasi Nasional. *Ikan Beku*; 2014. SNI 4110:2014.
20. Luthfi AZA. Analisa Pengukuran Usability Sistem Menggunakan Metode Use Questionary pada Aplikasi Duolingo. *JTIK*. 2021;5(2):267-75. Accessed: Jul. 6, 2024. Available from: <https://doi.org/10.59697/jtik.v5i2.553>.
21. Lathifah HM, Novamizanti L, Rizal S. Fast and accurate fish classification from underwater video using you only look once. In: *IOP Conference Series: Materials Science and Engineering*. vol. 982. IOP Publishing; 2020. p. 012003.
22. Akhyar F, Novamizanti L, Putra T, Furqon EN, Chang MC, Lin CY. Lightning YOLOv4 for a surface defect detection system for sawn lumber. In: *2022 IEEE 5th International Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE; 2022. p. 184-9.
23. Akhyar F, Novamizanti L, Riantiarni T. Sistem inspeksi cacat pada permukaan kayu menggunakan model deteksi obyek YOLOv5. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*. 2022;10(4):990.
24. Aziz AK, Maulana MD, Adawiyah RF, Firdaus RF, Novamizanti L, Ramdhon F. Comparative analysis of YOLOv8 models in skip-jack fish quality assessment system. In: *2023 3rd International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*. IEEE; 2023. p. 237-42.
25. Azizah S, Wahidin, Padang M, Novamizanti L, Saidah S. Identifying the Ripeness and Quality Level of Strawberries Based on YOLOv7-EfficientNet. In: *The 5th International Conference on Data Science and Its Applications (ICoDSA)*. IEEE; 2024. .
26. Wicaksono BA, Novamizanti L, Ibrahim N. Tea leaf maturity levels based on ycbcr color space and clustering centroid. In: *Journal of Physics: Conference Series*. vol. 1367. IOP Publishing; 2019. p. 012028.