

Desain dan Implementasi *Website* Harvest Lens Prediksi Harga Beras Menggunakan *Framework* Streamlit

1st Muhammad Billy Julyano

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

billyjulyano@student.telkomuniversity.ac.id

2nd Suryo Adhi Wibowo

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

suryoadhiwibowo@telkomuniversity.ac.id

3rd Koredianto Usman

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

korediantousman@telkomuniversity.ac.id

Pergerakan harga pangan di Indonesia memiliki dampak signifikan terhadap ekonomi dan kesejahteraan masyarakat. Fluktuasi harga dan ketidaksesuaian pasokan sering menjadi pemicu inflasi yang mengganggu stabilitas ekonomi. Oleh karena itu, pemahaman yang akurat tentang dinamika harga pangan sangat penting untuk mendukung kebijakan yang efektif dalam mengendalikan inflasi dan memastikan ketersediaan pangan yang memadai. Program prediksi harga pangan yang dikembangkan oleh Badan Pangan Nasional, Telkom University, dan Badan Riset Inovasi Nasional bertujuan untuk meningkatkan pemahaman ini. Telkom University berperan dalam mengembangkan prediksi harga pangan berbasis web yang disebut Harvest Lens. *Website* ini dikembangkan menggunakan *Use Case Diagram* dan *Activity Diagram* untuk perancangan, serta Python dan *Framework* Streamlit untuk implementasi. Pengujian menggunakan *System Usability Scale* (SUS) menunjukkan rata-rata nilai 87,08, mencerminkan kategori "*Best Imaginable*" menurut skala SUS. Hasil ini menunjukkan bahwa *website* berhasil memenuhi kebutuhan pengguna dengan baik dan memberikan pengalaman pengguna yang memuaskan.

Kata kunci— *Website*, Python, Streamlit, SUS

I. PENDAHULUAN

Pergerakan harga pangan memiliki dampak yang signifikan pada ekonomi dan kesejahteraan masyarakat di Indonesia. Data dari Badan Pusat Statistik menunjukkan bahwa kontribusi Garis Kemiskinan Makanan (GKM) terhadap garis kemiskinan sangat signifikan mencapai 73%. Fluktuasi harga dan ketidaksesuaian pasokan seringkali menjadi masalah utama yang memicu inflasi dan mengganggu stabilitas ekonomi secara keseluruhan. Harga pangan yang tidak stabil dapat mengakibatkan kenaikan angka kemiskinan dan penurunan daya beli masyarakat. Oleh karena itu, pemahaman yang akurat tentang pergerakan harga pangan sangat penting untuk mendukung pengambilan kebijakan yang efektif, yang bertujuan untuk mengendalikan inflasi dan memastikan ketersediaan pangan yang memadai bagi masyarakat.

Badan Pangan Nasional (BAPANAS), Telkom University, dan Badan Riset Inovasi Nasional (BRIN) bekerja sama dalam pengembangan program prediksi harga pangan di Indonesia. Program ini bertujuan untuk meningkatkan pemahaman tentang dinamika harga pangan sehingga BAPANAS dapat merencanakan kebijakan yang lebih efektif dalam mengendalikan inflasi, menjaga ketersediaan pangan, dan memastikan stabilitas ekonomi serta kesejahteraan masyarakat. Selain itu, program ini membantu dalam mengidentifikasi periode kritis di mana harga pangan cenderung naik, sehingga pemerintah dapat mempersiapkan langkah-langkah yang tepat. Telkom University berperan dalam pengembangan algoritma prediksi harga pangan yang akurat berbasis *website*, dengan harapan dapat memberikan kontribusi signifikan dalam mengatasi permasalahan harga pangan dan memperkuat ketahanan pangan nasional.

II. KAJIAN TEORI

Kajian teori yang digunakan dalam perancangan *website* Harvest Lens meliputi *Use Case Diagram* dan *Activity Diagram*. Aplikasi ini dibangun menggunakan bahasa pemrograman Python dan *Framework* Streamlit untuk mengimplementasikan *website*. Pengujian aplikasi dilakukan menggunakan *System Usability Scale* (SUS). Berikut merupakan penjelasan dari kajian teori dalam pembuatan *website* Harvest Lens:

A. *Use Case Diagram*

Use case diagram merupakan alat visual dalam rekayasa perangkat lunak yang digunakan untuk mengilustrasikan interaksi antara aktor eksternal (seperti pengguna atau sistem lain) dengan sistem yang dikembangkan [1]. *Use Case Diagram* menampilkan berbagai kasus penggunaan (*use cases*) yang mengidentifikasi fungsi atau layanan yang disediakan oleh sistem bagi para aktor. *Use case diagram* membantu dalam memahami kebutuhan sistem dari sudut pandang pengguna dan memastikan bahwa semua fungsi yang diperlukan telah dikenali dan dianalisis dengan baik.

B. Activity Diagram

Activity Diagram adalah jenis diagram yang digunakan untuk memodelkan alur kerja atau aktivitas dalam sistem. Diagram ini menggambarkan urutan aktivitas, keputusan, dan paralelisme, membantu pemangku kepentingan memahami dan mengoptimalkan proses yang kompleks [2]. Aktivitas ditunjukkan dengan elemen diagram seperti kotak atau oval, panah yang menunjukkan aliran antar aktivitas, dan keputusan.

C. Pemrogramanbahasa Python

Python adalah bahasa pemrograman yang populer karena sintaksisnya yang sederhana dan mudah dipelajari, serta kemampuannya untuk digunakan dalam berbagai jenis pengembangan, termasuk *website*, *data science*, dan otomatisasi [3]. Python mendukung pemrograman berorientasi objek dan prosedural, serta memiliki ekosistem pustaka yang luas, seperti NumPy dan Pandas untuk analisis data dan pengembangan web. Bahasa ini juga dikenal dengan komunitas yang besar dan aktif, yang terus berkontribusi pada peningkatan dan pemeliharaan alat dan pustaka yang ada.

D. Framework Streamlit

Streamlit adalah *framework open-source* yang digunakan untuk membangun aplikasi web interaktif berbasis data secara cepat dan mudah menggunakan Python. *Framework* ini sangat cocok untuk *data scientists* dan *machine learning engineers* karena memungkinkan pembuatan antarmuka pengguna (UI) dengan sedikit kode dan tanpa memerlukan pengetahuan mendalam tentang pengembangan web [4]. Dengan sintaksis yang sederhana dan intuitif, Streamlit memungkinkan pengguna untuk menampilkan visualisasi data, model machine learning, dan elemen interaktif lainnya secara real-time, sehingga mempermudah dalam eksplorasi dan presentasi data.

E. System Usability Scale

System Usability Scale (SUS) adalah alat untuk mengevaluasi kegunaan sistem atau produk. SUS memberikan gambaran umum mengenai persepsi pengguna terhadap sistem yang diuji, yang berguna dalam proses pengembangan untuk meningkatkan pengalaman pengguna. Pengujian dengan metode SUS menggunakan google formulir dengan menyebar kuesioner yang terdiri dari sepuluh pertanyaan dengan lima pilihan jawaban dari "Sangat Tidak Setuju" hingga "Sangat Setuju". SUS digunakan untuk mengukur sejauh mana suatu sistem dapat digunakan dengan mudah dan memuaskan oleh pengguna.

Nilai yang telah di isi dari setiap responden akan dihitung total nilai setiap responden menggunakan rumus (1) :

$$x = (P1 - 1) + (5 - P2) + (P3 - 1) + (5 - P4) + (P5 - 1) + (5 - P6) + (P7 - 1) + (5 - P8) + (P9 - 1) + (5 - P10) . * 2,5 (P) \tag{1}$$

Keterangan:

x = total nilai responden

P = nilai setiap pertanyaan

Setelah penilaian total nilai setiap responden didapatkan, Kemudian terdapat perhitungan nilai rata-rata SUS dengan rumus (2):

$$\bar{X} = \frac{\sum X}{n} \tag{2}$$

Keterangan:

\bar{X} = nilai rata-rata SUS

$\sum X$ = total nilai responden

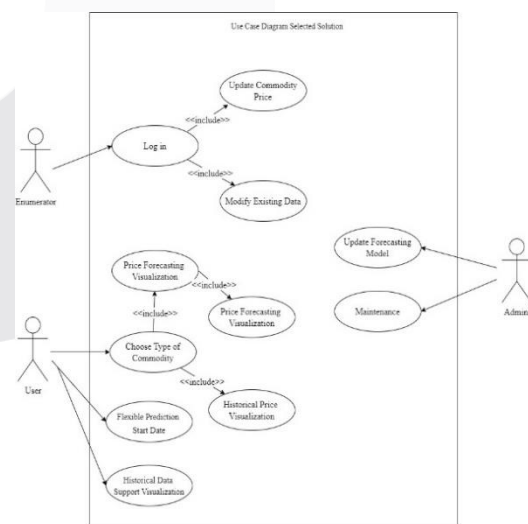
n = jumlah responden

III. METODE

Perancangan sistem yang dilakukan pada *website* Harvest Lens meliputi perancangan *use case diagram*, *activity diagram*, dan perancangan *website* dari pemrosesan desain sampai implementasi kode.

A. Use Case Diagram

Pada Gambar 1, terlihat terdapat tiga aktor yang mewakili entitas yang akan berinteraksi dengan aplikasi yaitu *user*, *enumerator*, dan *admin*. Enumerator dapat mengakses fitur *login* dimana fitur dibutuhkan untuk enumerator melakukan pembaruan data harga. Lalu *user* dapat menggunakan aplikasi untuk melihat pergerakan harga pangan berdasarkan jenis pangan yang dipilih dengan tanggal awal prediksi yang fleksibel, dan melihat pergerakan historis data support. Lalu terdapat juga fitur untuk memberitahu *user* bagaimana tren harga di masa depan terhadap harga pangan di masa sekarang. Sementara itu, administrator bertanggung jawab atas *maintenance* aplikasi baik secara keamanan maupun pembaruan model *forecasting* secara berkala, hal ini dilakukan untuk menjaga keandalan performa model dan kinerja aplikasi secara keseluruhan. Kolaborasi ketiga aktor ini membentuk ekosistem yang seimbang dalam manajemen data harga pangan.



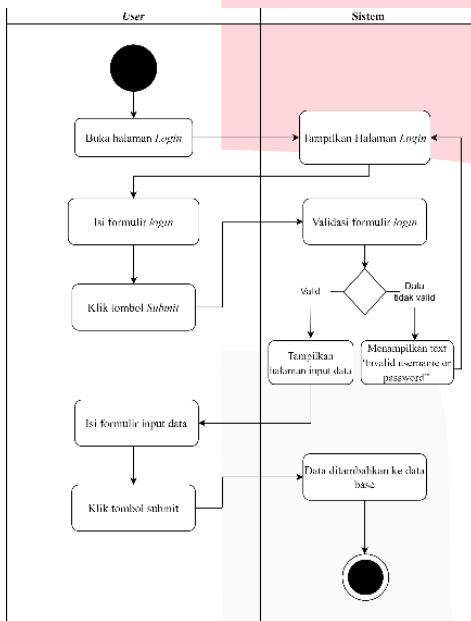
GAMBAR 1 Use Case Diagram

B. Activity Diagram

Activity diagram *login* input data mengilustrasikan langkah-langkah yang terlibat dalam proses *login* ke sistem. Pada Gambar 2, langkah-langkah dimulai dengan *user* membuka halaman *login*, mengisi formulir *login*, dan

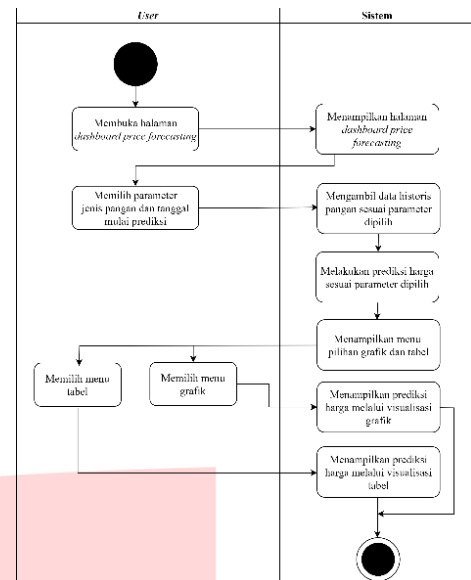
menekan tombol *submit*. Sistem kemudian memvalidasi kredensial yang dimasukkan oleh *user*. Jika kredensial valid, sistem menampilkan halaman formulir input data. Jika kredensial tidak valid, sistem menampilkan pesan "*Invalid username or password*". Setelah berhasil *login*, *user* akan diarahkan untuk mengisi formulir input data. Setelah *user* mengisi formulir input data dan menekan tombol *submit*, sistem kemudian akan menambahkan data ke dalam basis data.

Activity diagram ini memberikan representasi visual dari serangkaian aktivitas yang terjadi dalam sistem, mulai dari *login* hingga memasukkan data. Diagram ini menunjukkan bahwa untuk mengakses formulir input data, *user* harus melakukan *login* terlebih dahulu sebelum data dapat ditambahkan ke dalam basis data.



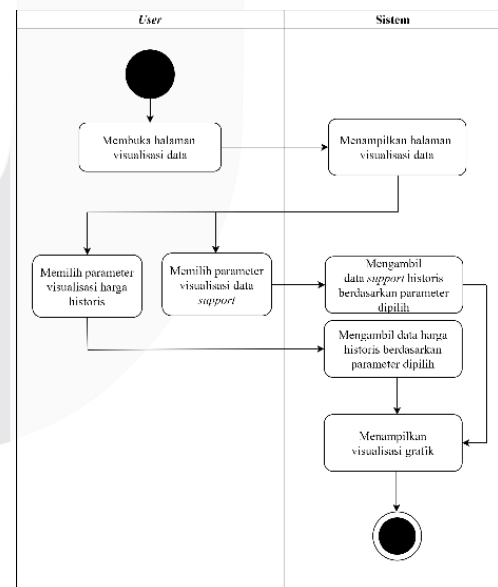
GAMBAR 2
Activity Diagram 1

Activity Diagram *dashboard price forecasting* pada Gambar 3 memberikan gambaran mengenai cara sistem menampilkan prediksi harga pangan secara visual. *User* dapat memilih jenis pangan dan tanggal mulai prediksi. Sistem kemudian mengambil data historis sesuai parameter yang dipilih, melakukan prediksi, dan menampilkan hasilnya dalam bentuk grafik dan tabel. *User* dapat menjelajahi data secara interaktif untuk mendapatkan wawasan tentang tren harga pangan yang mendukung pengambilan keputusan. Activity diagram ini menunjukkan langkah-langkah dari menampilkan prediksi hingga penyajian hasil melalui tabel dan grafik.



GAMBAR 3
Activity Diagram 2

Activity diagram visualisasi data historis berfungsi untuk memberikan gambaran mengenai cara pengguna melakukan visualisasi data historis serta cara kerja sistem dalam menampilkan visualisasi data. Diagram ini, seperti yang ditunjukkan pada Gambar 4, memiliki dua bagian menu: visualisasi data harga historis dan visualisasi data *support* historis. Ketika pengguna membuka halaman visualisasi data historis, sistem akan menampilkan formulir di mana pengguna dapat memilih parameter seperti tipe data dan rentang tanggal. Setelah pengguna memilih parameter, sistem akan mengambil data sesuai parameter yang dipilih dan menampilkan visualisasi data dalam bentuk grafik.



GAMBAR 4
Activity Diagram 3

C. Perancangan Website

Pengembangan *website* adalah proses merancang dan membangun situs web agar berfungsi dengan baik di internet. Tahap pertama adalah analisis kebutuhan, di mana kebutuhan dan persyaratan *website* ditentukan. Tahap kedua adalah desain visual dan tata letak, yang dirancang menggunakan aplikasi Figma. Tahap ketiga adalah implementasi kode, di

mana *website* dibangun menggunakan bahasa pemrograman Python dengan *framework* Streamlit dan aplikasi Visual Studio Code digunakan untuk mengimplementasikan desain menjadi kode. Selanjutnya, tahap pengujian dilakukan untuk menguji dan memastikan fungsionalitas.

Desain *website* adalah proses merancang tampilan visual dan struktur situs web agar menarik, mudah digunakan, dan memenuhi kebutuhan pengguna. Ini melibatkan pengaturan elemen seperti tata letak, warna, font, grafik, dan interaksi pengguna. Desain UI/UX yang kreatif sangat penting untuk efektivitas [5]. Aspek penting meliputi kerapian, fungsionalitas, responsivitas, konsistensi, dan kesesuaian dengan kebutuhan. Desain yang baik harus sederhana, mudah dipahami, dan tetap memiliki fitur yang dibutuhkan. Setelah melakukan analisis, *website* harus memiliki desain sederhana namun fungsional, memberikan pengalaman yang intuitif bagi pengguna. Aplikasi Figma akan digunakan untuk mendesain *website*, memungkinkan implementasi mudah dengan fitur-fiturnya.

Implementasi kode adalah langkah berikutnya, di mana desain yang dirancang diubah menjadi kode program. Pengembangan web dilakukan menggunakan Visual Studio Code. Visual Studio Code digunakan untuk menulis, mengedit, dan mengelola kode dalam berbagai bahasa pemrograman. Dalam proyek ini, bahasa pemrograman yang digunakan adalah Python, yang digunakan dalam berbagai aplikasi, termasuk pengembangan perangkat lunak, analisis data, dan pengembangan web. Untuk membangun aplikasi web interaktif, *framework* Streamlit digunakan bersama Python. Streamlit adalah *framework open-source* yang memungkinkan pengembang untuk membuat aplikasi web interaktif dengan cepat dan mudah menggunakan Python. Proses implementasi kode melibatkan penulisan dan pengujian kode komputer untuk memastikan situs web yang efektif dan fungsional. Kode yang dirancang memastikan bahwa setiap elemen dan fitur bekerja sesuai dengan desain yang telah ditetapkan.

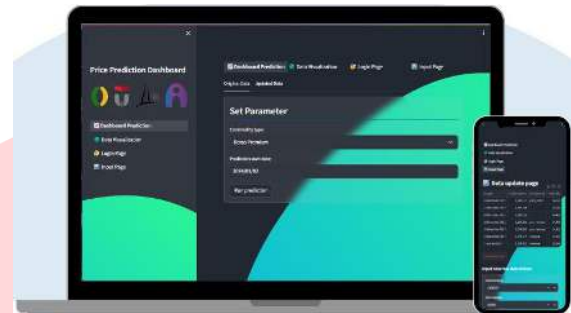
Setelah mengimplementasikan kode, tahap berikutnya adalah tahap *deployment*. Peluncuran atau *deployment website* merupakan tahap pengembangan situs web di mana situs yang telah selesai dikembangkan ditempatkan di server agar dapat diakses secara online. Proses ini melibatkan transfer *file* dan data ke server serta pengujian akhir sebelum situs siap digunakan. Selama *deployment*, pengembang memastikan semua aspek diatur dengan benar agar situs beroperasi optimal. Setelah berhasil di-*deploy*, situs dapat diakses melalui URL yang ditentukan, dan pengembang dapat memantau serta memperbarui situs sesuai kebutuhan.

Tahap selanjutnya merupakan tahap pengujian. Pengujian *website* adalah langkah dalam pengembangan situs web untuk memastikan semua fitur berjalan sesuai harapan dan memberikan pengalaman pengguna optimal. Proses ini mencakup pengujian fungsionalitas dan kinerja. Pengujian fungsionalitas memeriksa elemen seperti tombol, formulir, dan tautan untuk memastikan semuanya berfungsi dengan benar.

IV. HASIL DAN PEMBAHASAN

A. Hasil Implementasi

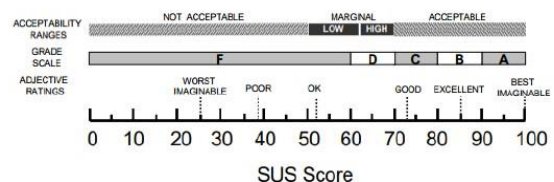
Hasil akhir dari penelitian ini adalah sebuah *website* yang menampilkan fitur-fitur yang telah dirancang. Fitur-fitur tersebut mencakup prediksi harga pangan, visualisasi data, input data, dan *login*. Selain fitur, desain yang sederhana dan intuitif juga diperhatikan agar dapat digunakan oleh pengguna berusia di atas 30 tahun. *Website* ini juga dibuat responsif agar dapat digunakan pada semua perangkat. Gambar 6 menunjukkan hasil akhir dari *website* tersebut.



GAMBAR 5
Hasil Implementasi

B. Pengujian System Usability Scale

Setelah *website* telah diimplementasikan, pengujian *website* dilakukan untuk memastikan fitur berjalan dan memberikan pengalaman pengguna yang optimal. Pengujian dilakukan dengan metode *System Usability Scale*. Dari perhitungan SUS dengan menyebarkan kuisioner kepada responden, didapatkan nilai total SUS sebesar 2612,5 dari 30 responden. Nilai rata-rata diperoleh dengan membagi nilai total SUS dengan jumlah responden, menghasilkan rata-rata 87,08. Dengan nilai rata-rata ini, parameter penilaian dapat diukur menggunakan Gambar 6 [6].



GAMBAR 6
Parameter penilaian SUS

Hasil nilai rata-rata nilai SUS sebesar 87,08 menunjukkan bahwa pengujian *website* telah mendapatkan kategori "Best Imaginable" menurut skala SUS. Skor ini berada pada *grade* B, yang mengindikasikan bahwa pengalaman pengguna *website* tersebut baik dan memenuhi harapan pengguna secara umum. Dengan demikian, *website* ini dinilai cukup memuaskan, menunjukkan bahwa sebagian besar pengguna merasa nyaman dan puas saat menggunakannya. Nilai ini mencerminkan bahwa desain, fungsionalitas, dan responsivitas *website* telah berhasil memenuhi kebutuhan dan preferensi pengguna.

V. KESIMPULAN

Perencanaan dan implementasi *website* terdiri dari beberapa tahap, mulai dari analisis kebutuhan, desain *website*, implementasi kode, *deployment*, hingga pengujian. Hasil akhir dari penelitian ini adalah sebuah *website* yang memiliki fitur prediksi harga, visualisasi data, dan input data serta desain yang sederhana dan mudah digunakan.

Hasil nilai rata-rata SUS dari total 30 responden adalah 87,08. Skor ini masuk dalam kategori “*Best Imaginable*” dengan rentang nilai 100-80. Pada skala *grade*, nilai 87,08 masuk dalam *grade* “B” dengan rentang nilai 90-80. *Website* ini menunjukkan bahwa desain dan fungsionalitasnya telah memenuhi harapan pengguna dan memberikan pengalaman pengguna yang baik. Dengan demikian, *website* ini dinilai cukup memuaskan dan dapat digunakan secara efektif oleh pengguna.

REFERENSI

- [1] M. A. Basuki, “Analisa Website Universitas Muria Kudus,” *Jurnal sains*, vol. 2, no. 2, hlm. 1–16, 2009.
- [2] J. S. Kurnia dan F. Risyda, “Rancang Bangun Penerapan Model Prototype Dalam Perancangan Sistem Informasi Pencatatan Persediaan Barang Berbasis Web,” *JSI (Jurnal sistem Informasi) Universitas Suryadarma*, vol. 8, no. 2, hlm. 223–230, 2021.
- [3] M. R. S. Alfarizi, M. Z. Al-farish, M. Taufiqurrahman, G. Ardiansah, dan M. Elgar, “Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning,” *Karimah Tauhid*, vol. 2, no. 1, hlm. 1–6, 2023.
- [4] N. Nurhapiza, N. S. Harahap, M. Fikry, dan M. Affandes, “Penerapan Chatbot pada Aplikasi Web Tanya Jawab Tentang Fiqih Jual Beli Islam Menggunakan LangChain,” *Journal of Computer System and Informatics (JoSYC)*, vol. 5, no. 3, hlm. 548–557, 2024.
- [5] A. A. Andryadi dan N. H. Fatonah, “Analisis User Experience Dan User Interface (Ui/Ux) Pada Website Menggunakan Metodegoogle Design Sprint,” *Jurnal Teknologi dan Bisnis*, vol. 3, no. 2, hlm. 137–144, 2021.
- [6] I. Salamah, “Evaluasi usability website polsri dengan menggunakan system usability scale,” *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, vol. 8, no. 3, hlm. 176–183, 2019.