Implementasi Model Ramalan Prophet Untuk Mendukung Keputusan Pada Website Monitoring Stok Produk

1st Muhammad Fathul Mubiin
Sistem Informasi
Telkom University Purwokerto
Banyumas, Indonesia
fathulmubin@student.telkomuniversity.ac.id

2nd Toni Anwar
Sistem Informasi
Telkom University Purwokerto
Banyumas, Indonesia
tonianwar@telkomuniversity.ac.id

3rd Abednego Dwi Septiadi
Software Engineering
Telkom University Purwokerto
Banyumas, Indonesia
abednego@telkomuniversity.ac.id

***Abstrak* — Toko Kripik Tempe Inti Sari masih menghadapi permasalahan dalam penentuan jumlah stok karena proses perencanaan produksi dan persediaan dilakukan berdasarkan intuisi pemilik serta pencatatan transaksi yang masih manual. Terdapat risiko menimbulkan kerugian penjualan akibat penentuan jumlah produksi yang berlebihan. Penelitian ini bertujuan mengembangkan sistem monitoring stok berbasis website yang mampu mengintegrasikan pengelolaan data stok, transaksi penjualan, dan peramalan permintaan produk untuk mendukung pengambilan keputusan penyediaan stok. Metode pengembangan sistem yang digunakan adalah Rapid Application Development, sedangkan pengujian fungsional dilakukan menggunakan Black-box Testing. Sistem dibangun menggunakan Laravel untuk pengelolaan operasional dan Flask sebagai layanan peramalan berbasis model Prophet. Hasil penelitian menunjukkan bahwa sistem berhasil mengolah data historis penjualan menjadi informasi prediksi permintaan stok dan menyajikannya dalam bentuk dasbor analitik. Pada studi kasus produk tempe ungkep bumbu kuning, sistem menghasilkan prediksi sebanyak 30 unit dibandingkan penjualan aktual 28 unit, sehingga memperkecil produk berlebih menjadi 2 unit dari 9 unit sebelum diteapkannya sistem. Seluruh fungsi sistem berjalan sesuai kebutuhan pengguna dan mampu mendukung digitalisasi pengelolaan stok serta penyediaan informasi berbasis data untuk membantu perencanaan persediaan yang lebih terukur.**

***Kata kunci*— SPK, situs web, peramalan, prophet, laravel, flask**

I. PENDAHULUAN

Limbah produk masih menjadi salah satu permasalahan utama dalam pengelolaan persediaan produk karena produk dengan umur simpan terbatas harus dikelola dengan tingkat ketepatan yang tinggi agar jumlah stok tetap seimbang dengan permintaan [1]. Perusahaan yang menyimpan stok terlalu banyak, risiko pemborosan meningkat, sedangkan stok yang terlalu rendah dapat menyebabkan kehilangan penjualan, sehingga pengambilan keputusan persediaan pada produk selalu berada pada pertimbangan antara kekurangan dan pemborosan [2]. Berbagai penelitian menunjukkan bahwa pengelolaan persediaan yang lebih adaptif dan berbasis data dapat menekan limbah secara signifikan, termasuk penurunan expired stock sebesar 69.6% [3]. Histori transaksi dan jumlah produk terjual merupakan sumber data penting untuk mengidentifikasi tren penjualan dan meramal permintaan stok pada periode berikutnya, sehingga sistem yang merangkum data tersebut dan menghasilkan ramalan permintaan stok relevan sebagai alat bantu keputusan [4].

Usaha Mikro, Kecil, dan Menengah (UMKM) toko Kripik Tempe Inti Sari merupakan usaha produsen sekaligus toko yang memasarkan berbagai produk makanan, khususnya keripik tempe dan olahan tempe siap masak lainnya. Proses penentuan volume produksi dan pengadaan stok di toko ini sepenuhnya bertumpu pada pengalaman dan intuisi pemilik. Pemilik menentukan jumlah produk yang harus disediakan berdasrakan ketahanan produk, proses penentuan hanya berdasarkan intuisi, tanpa melibatkan perhitungan matematis atau analisis tren deret waktu (time-series). Pendekatan berbasis intuisi ini sangat rentan terhadap bias perkiraan manusia.

Berdasarkan catatan produksi periode 30 Mei–5 Juni 2026, pemilik toko menyediakan stok tempe ungkep bumbu kuning sebanyak 40 unit berdasarkan intuisi. Data penjualan aktual yang dicatat karyawan menunjukkan produk tersebut hanya terjual 31 unit hingga 5 Juni 2026. Selisih 9 unit antara perkiraan intuitif dan permintaan aktual ini menyebabkan pemborosan modal kerja, karena sisa stok tersebut telah melewati masa simpan dan tidak dapat dijual. Permasalahan ini diperparah oleh manajemen data operasional yang belum terstruktur, sebab pencatatan transaksi penjualan harian dan mutasi stok masih dilakukan secara manual menggunakan buku log fisik dan nota kertas.

Penelitian ini bertujuan untuk mengurangi risiko produksi berlebih di toko Kripik Tempe Inti Sari melalui penerapan model Prophet dalam menentukan jumlah produksi. Model tersebut diimplementasikan pada sistem berbasis website yang mengintegrasikan pencatatan transaksi, pengelolaan stok, dan penyajian informasi analitik. Pengujian fungsional merupakan metode pengujian yang dilakukan dari sudut pandang pengguna tanpa memerlukan pengetahuan tentang struktur internal kode program. Evaluasi sistem difokuskan pada fitur peramalan dalam memprediksi permintaan produk tempe ungkep bumbu kuning untuk periode 27 Juni–3 Juli 2026. [10 pts]

II. KAJIAN TEORI

A. Sistem Pendukung Keputusan (SPK)

SPK merupakan sistem berbasis komputer yang dirancang untuk membantu pengambil keputusan dalam memanfaatkan data dan model tertentu guna memecahkan

permasalahan yang bersifat semi-terstruktur maupun tidak terstruktur. SPK tidak dimaksudkan untuk menggantikan peran pengambil keputusan, melainkan sebagai alat bantu yang memperluas kemampuan pengambil keputusan dalam mengolah informasi tanpa menggantikan penilaian (*judgment*) manusia dalam proses pengambilan keputusan akhir [5].

B. Flask

Flask terdiri dari dua komponen utama Werkzeug dan Jinja. Werkzeug menangani routing, debugging, dan website server gateway interface (WSGI), sementara Flask menggunakan Jinja2 sebagai mesinnya. Flask memungkinkan integrasi ekstensi untuk menambahkan fungsionalitas tersebut, menjadikan Flask sebagai microframework yang tetap cocok untuk penggunaan produksi saat mengembangkan aplikasi dan layanan website. Aplikasi Flask dasar dapat ditempatkan dalam satu berkas Python atau diorganisir secara modular untuk membentuk aplikasi yang siap produksi. Konsep Flask adalah menyediakan fondasi yang kokoh untuk semua aplikasi, sementara fitur tambahan diimplementasikan melalui ekstensi [6].

C. *Application Programming Interface* (API)

API hadir sebagai antarmuka khusus yang memungkinkan interaksi antar-perangkat lunak. API menyediakan mekanisme kontrak yang terstandardisasi untuk menghubungkan, mengintegrasikan, serta memperluas kapabilitas suatu sistem informasi. Dalam arsitektur sistem terdistribusi, API memegang peran krusial dalam membentuk komponen-komponen sistem yang saling lepas, sehingga perubahan pada satu komponen tidak akan mengganggu stabilitas komponen lainnya .

D. Prophet

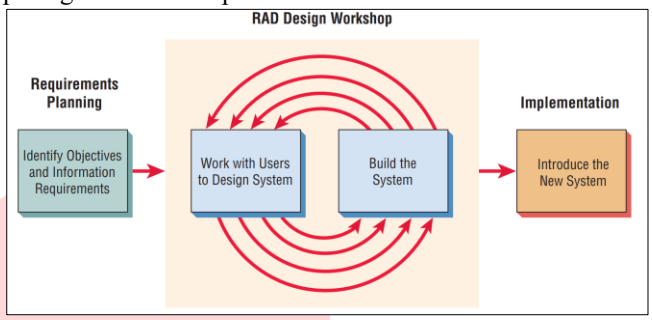
Prophet, sebuah fremework peramalan deret waktu yang dikembangkan dan dirilis sebagai perangkat lunak sumber terbuka oleh Facebook, tidak berfungsi sebagai satu model tunggal, melainkan sebagai suatu prosedur otomatis untuk membangun model forecasting. Prophet dibatasi pada pemodelan peramalan univariat, artinya hanya dapat menghasilkan ramalan berdasarkan satu variabel dependen. Kebutuhan peramalan untuk multivariat, pengguna harus membangun model prophet terpisah untuk setiap variabel yang akan diramal. Prophet memerlukan dataset dengan struktur kolom khusus, yakni ds sebagai penanda tanggal dan y sebagai variabel dependen [7].

E. *Black-box testing*

Black-box testing, sering disebut sebagai behavioral testing atau functional testing, berfokus pada pengujian kesesuaian perangkat lunak terhadap kebutuhan fungsional yang telah ditetapkan. Teknik pengujian ini memungkinkan perancang uji untuk menyusun sekumpulan kondisi input yang mampu menguji seluruh fungsi perangkat lunak tanpa memerhatikan struktur internal kode. Black-box testing berperan sebagai pendekatan yang mampu mengungkap jenis kesalahan berbeda dari yang ditemukan melalui pengujian berbasis logika internal [8].

III. METODE

Prosedur penelitian ini menggunakan pendekatan metode *Rapid Application Development* (RAD). RAD terdiri atas tiga fase utama yang melibatkan kolaborasi aktif antara pengguna dan analis sistem, yaitu tahap perencanaan kebutuhan, perancangan, dan implementasi. Pendekatan ini menekankan keterlibatan pengguna secara intensif dalam setiap tahapan pengembangan, khususnya pada proses perancangan yang berkaitan langsung dengan kebutuhan bisnis [9]. Visualisasi pada gambar 1 merupakan alur fase metode RAD.



GAMBAR 1
(Fase RAD)

Pada tahap *Requirements Planning* peneliti meninjau lingkup sistem dan permasalah yang ingin diselesaikan untuk menentukan bagaimana kebutuhan fungsional dan non fungsional sistem informasi dapat membantu mencapai kondisi operasional yang ideal.

Sub-proses *design system* pada siklus design *workshop melanjutkan* rancangan kerangka konseptual representasi kelas, operasi, dan asosiasi model struktural melalui pembuatan *class* diagram, sebagai visualisasi perancangan struktur data pada sistem yang akan dikembangkan

Sub-proses *build system* pada siklus *design workshop*, merancang atau menyempurnakan prototipe yang meliputi antarmuka, pengalaman, serta fungsionalitas modul yang dikembangkan. Prototipe yang sudah dikembangkan akan dilakukan pengujian fungsional modul sebelum demonstrasi prototipe kepada pengguna agar melibatkan pengujian dan evaluasi pengguna secara langsung.

Proses *implementation* melibatkan penerapan sistem pada lingkungan operasional (*deployment*) agar dapat digunakan secara nyata. Sebagai evaluasi implementasi, kinerja sistem akan diukur melalui observasi catatan transaksi penjualan produk tempe ungkep bumbu kuning serta identifikasi selisih antara hasil ramalan sistem dan penjualan aktual pada periode 27 Juni–3 Juli 2026.

IV. HASIL DAN PEMBAHASAN

Kebutuhan fungsional secara spesifik menguraikan kemampuan sistem, mencakup batasan operasional, interaksi antarmuka, hingga logika pemrosesan. Pada tabel 1 menyajikan rincian kebutuhan fungsional sistem yang telah dirancang.

TABEL 1
(Kebutuhan Fungsional *System*)

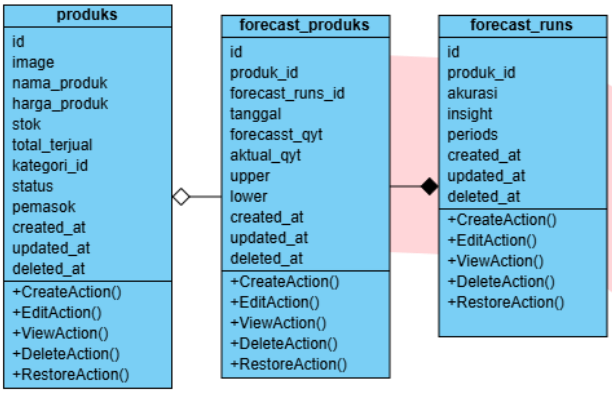
Fungsional	Penjelasan
Peramalan penjualan stok	Memberikan informasi akurasi, insight, dan ramalan

Kebutuhan non-fungsional menitikberatkan pada bagaimana sistem tersebut beroperasi. Aspek ini mencakup batasan teknis, parameter kinerja, dan standar kualitas yang harus dipenuhi ditunjukkan pada tabel 2.

TABEL 2
(Kebutuhan Fungsional *System*)

Fungsional	Penjelasan
Performa	Memvisualisasikan data dalam waktu kurang dari 3 detik

Representasi kelas, operasi, dan asosiasi model struktural ditunjkan pada gambar 2, sebagai visualisasi perancangan struktur data pada sistem yang akan dikembangkan di penelitian ini.



GAMBAR 2
(*Class* Diagram Struktur Data Terkait Modul Peramalan)

Diagram *class* pada gambar 4.46 memvisualisasikan struktur data sistem dengan tambahan *class forecast_produks*, dan *forecast_runs*. *Class* produks ini terhubung melalui relasi agregasi dengan *class forecast_produks* dengan klausul multiplisitas 1 ke ∗. *class forecast_produks* berfungsi mencatat hasil prediksi individual yang meliputi atribut id, produk_*id*, *forecast_runs_id*, tanggal, *forecasst_qyt*, aktual_*qyt*, *upper*, dan *lower*. *class forecast_runs* yang merekam jejak setiap proses eksekusi peramalan dengan atribut berupa *id*, produk_*id*, akurasi, *insight*, dan *periods*. *Class forecast_runs* ini terhubung ke kelas *forecast_produks* menggunakan relasi komposisi dengan klausul multiplisitas 1 ke ∗.

Implementasi model Prophet akan menggunakan alat berupa *library* Flask. Aplikasi Flask akan dugunakan untuk memisah layanan peramalan untuk kebutuhan skalabilitas di masa mendatang. Dokumentasi kontrak API yang dihasilkan dari pengembangan memberikan teknis mengenai cara kerja ramalan sistem sekaligus menjadi acuan bagi pengembangan atau integrasi sistem yang di tunjukan pada table 3.

TABEL 3
(Kontrak API)

Komponen	Keterangan
HTTP Method	POST
URL Endpoint	api/forecast
Headers	Content-Type: application/json
Authorization	Bearer
Request Body	{ "data": [{"string", "integer"}, …], "periods": "integer" }
Response Sukses	{ "status": "success", "data": { "validation": […], "forecst": […], "akurasi": "integer", "insight": { "reason": "string", "summary": "string"} } }
Response Gagal	{ "status": "error", "message": "string" }

Pengujian ini dilakukan untuk memastikan bahwa implementasi dapat merespons dengan baik. Rincian scenario pengujian tersebut disajikan pada Tabel 4 uji *Black-box.*

TABEL 4
(Pengujian *Black-Box* Implementasi Prophet)

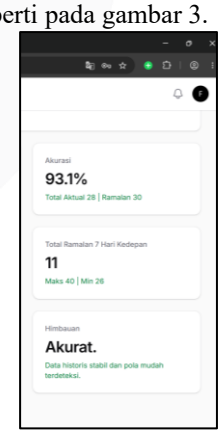
Skenario	Hasil yang Diharapkan	Hasil Pengujian	Status
Pengiriman data ke API Flask	Data berhasil dikirim ke API	Data berhasil dikirim ke API	Sesuai
Proses prediksi Prophet berjalan	Model menghasilkan prediksi	Model menghasilkan prediksi	Sesuai
Penanganan data kosong	Sistem tidak crash dan menampilkan log error	Sistem tidak crash dan menampilkan log error	Sesuai
Penanganan error API	Sistem menangani error (retry/log)	Sistem menangani error (retry/log)	Sesuai
Visualisasi hasil prediksi	Grafik prediksi tampil dengan benar	Grafik prediksi tampil dengan benar	Sesuai

Hasil evaluasi kesesuaian sistem dirangkum secara rinci pada Tabel 5, berdasarkan umpan balik pengguna akhir terhadap prototipe yang telah dibangun. Umpan balik tersebut mencakup *user reaction*, *user suggestion*, dan *innovation revision plan*.

TABEL 5
(Formulir Evaluasi *Prototipe*)

Pengguna n	Pengguna 1
Nama pengguna	Pak Untung
Periode Observasi	19-03-2026
User reaction	Penjeasannya mudah dimengerti
User suggestion	Bisa dilanjutkan
Inovation	
Revision Plans	

Evaluasi implementasi model Prophet menghasilkan ramalan permintaan stok produk tempe ungkep bumbu kuning berdasarkan histori transaksi penjualan yang telah tercatat sebelumnya.. Hasil ramalan dari implementasi model Prophet terlihat seperti pada gambar 3.



GAMBAR 3
(Hasil Ramalan Sistem)

Pengaruh implementasi sistem dalam menekan angka produksi berlebih, dilakukan analisis komparatif terhadap ramalan permintaan tempe ungkep bumbu kuning untuk periode 27 juni sampai 3 Juli 2026. Analisis ditujukan untuk membandingkan seberapa besar selisih yang dihasilkan antara perkiraan berbasis intuisi manual pemilik toko dan hasil angka alternatif yang dikeluarkan oleh sistem

berdasarkan data deret waktu. Sajian data perbandingan selisih unit tersebut ditunjukkan pada tabel 6.

TABEL 6
(Formulir Evaluasi Prototipe)

Pendekatan Analisis	Hasil ramalan	Penjualan Aktual	Selisih Aktual
Intuisi Manual Pemilik	40 Unit	31 Unit	9 Unit
Hasil Peramalan Sistem	30 Unit	28 Unit	2 Unit

Berdasarkan Tabel tabel 6, terlihat adanya perbedaan selisi kerugian produksi. Pengambilan keputusan penyediaan produk yang murni mengandalkan intuisi manual menghasilkan jarak kelebihan sebanyak 9 unit dari permintaan aktual. Jarak selisih berpotensi meningkatkan risiko kerugian produksi. Hasil angka yang dikeluarkan oleh sistem memberikan pengaruh yang berbeda terhadap angka kerugian tersebut. Melalui pemrosesan implementasi Prophet, sistem mengeluarkan angka ramalan sebesar 30 unit, yang memperkecil jarak selisih menjadi hanya 2 dari penjualan aktual 28 unit. Angka selisih dari sistem ini kemudian digunakan sebagai penentu insight dinamis dari sistem. Insight dinamis menjadi rekomendasi keputusan dari sistem ini.

V. KESIMPULAN

Penggunaan model Prophet pada sistem berbasis website mampu mengurangi produksi berlebih yang berpotensi menyebabkan kerugian bagi Toko Kripik Tempe Inti Sari. Sistem menghasilkan rekomendasi jumlah produksi sebanyak 30 unit berdasarkan pemrosesan data deret waktu masa lalu. Hasil rekomendasi tersebut memiliki selisih 2 unit dibandingkan dengan penjualan aktual sebanyak 28 unit pada periode 27 Juni sampai 3 Juli. Toko Kripik Tempe Inti Sari mengalami produksi berlebih sebanyak 9 unit sebelum sistem diterapkan. Sistem mampu menurunkan produksi berlebih sebesar 77,78% sehingga dapat membantu mengurangi risiko kerugian akibat kelebihan produksi. Capaian ini tidak lepas dari fungsionalitas sistem yang sudah teruji. Black-box digunakan sebagai uji fungsionalitas sistem. Dari 5 skenario uji menghasilkan 100% kesesuaian.

REFERENSI

[1] M. E. Staff and N. Mustafee, “Discrete-event simulation for effective perishable inventory management: a review,” *Simulation*, vol. 101, no. 9, pp. 981–1000, Sep. 2025.

[2] M. Motamedi, N. Li, and D. G. Down, “Optimal Ordering Policy for Perishable Products by Incorporating Demand Forecasts.” [Online]. Available: https://ssrn.com/abstract=4807963

[3] A. Zare, S. Kalantari, S. Rasouli, X. Wang, C. A. Huerta-Aguilar, and N. R. Smith, “Optimizing Order Decisions for Perishable Product Inventory Management: A Policy-Based Dynamic Programming Approach,” in *2025 IEEE Mexican Humanitarian Technology Conference (MHTC)*, IEEE, pp. 166–172, Jun. 2025.

[4] V. K. Masiyare and A. K. Sharma, “A Mathematical Inventory Optimization Framework with AI- Based Demand Forecasting for Perishable Agricultural Products,” *International Journal of Drug Delivery Technology*, vol. 16, no. 17s, Apr. 2026.

[5] P. B. Keenan and C. Heavin, “Scientific Authorship in DSS Research: Past Trends and Future Opportunities,” 2023, pp. 287–301.

[6] Kunal. Relan, *Building REST APIs with Flask : create Python web services with MySQL*. Apress, 2019.

[7] Joos. Korstanje, *Advanced forecasting with Python : with state-of-the-art-models including LSTMs, Facebook’s Prophet, and Amazon’s DeepAR*. Apress, 2021.

[8] R. S. . Pressman and B. R. . Maxim, *Software engineering : a practitioners approach*. McGraw-Hill Education, 2020.

[9] K. E. . Kendall and J. E. . Kendall, *Systems analysis and design*. Pearson Education Limited, 2020.