

Rancang Bangun *System Predictive Lead Scoring (PROSPECTA)* untuk Menentukan Prospek Nasabah Deposito Berjangka Menggunakan Next.js

Azizun Fathin Wahyu Lestari¹, Firdhaus Hari S A H², Diyah Ruswanti³

^{1,2,3}Program Studi Informatika, Fakultas Sains Teknologi dan Kesehatan,
Universitas Sahid Surakarta

Email: ¹azizunfathin@gmail.com, ²pakedozuss@usahidsolo.ac.id, ³dyahruswanti@usahidsolo.ac.id

(Naskah masuk: 1 Juni 2026, diterima untuk diterbitkan: 3 Juli 2026)

Abstrak: Perkembangan teknologi informasi di sektor perbankan mendorong pemanfaatan data nasabah secara optimal untuk mendukung pengambilan keputusan berbasis data. Salah satu permasalahan yang dihadapi adalah kondisi *lead overload*, di mana jumlah calon nasabah tidak sebanding dengan kemampuan tim sales dalam melakukan tindak lanjut, sehingga pemasaran menjadi kurang efektif. Penelitian ini bertujuan untuk mengembangkan sistem *Predictive Lead Scoring (PROSPECTA)* berbasis web yang mampu membantu proses prioritas prospek nasabah deposito berjangka. Metode penelitian yang digunakan meliputi observasi, studi literatur, dan metode pengembangan sistem *Waterfall*. Proses pengolahan data dilakukan menggunakan *preprocessing Robust Scaler* untuk menstabilkan skala data, serta penerapan algoritma *Random Forest* dalam membangun model prediksi. Sistem diimplementasikan dalam bentuk dashboard menggunakan *Next.js* yang menyajikan hasil prediksi probabilitas secara informatif. Hasil pengujian menunjukkan bahwa model memiliki tingkat akurasi sebesar 93%, serta seluruh fitur berjalan dengan baik berdasarkan pengujian *blackbox* dan API menggunakan *Postman*. Dengan demikian, *prospecta* mampu mendukung efektivitas kerja tim sales dalam menentukan prioritas prospek serta meningkatkan efisiensi pemasaran berbasis data.

Kata Kunci – *Predictive Lead Scoring; Machine Learning; Random Forest; Dashboard; Perbankan*

Design and Development of Predictive Lead Scoring System (PROSPECTA) for Time Deposit Customers Using Next.js

Abstract: Advancements in information technology in the banking sector have encouraged the optimal utilization of customer data to support data-driven decision making. One of the challenges faced is *lead overload*, where the number of potential customers is not proportional to the sales team's capacity to follow up, resulting in less effective marketing processes. This study aims to develop a web-based *Predictive Lead Scoring* system (PROSPECTA) to assist in prioritizing potential customers for term deposit products. The research methodology includes observation, literature study, and the *Waterfall* system development method. The proposed method includes data preprocessing using *Robust Scaler* to stabilize data scaling, as well as the implementation of the *Random Forest* algorithm to build the prediction model. The system is implemented in the form of a dashboard using *Next.js*, which presents prediction results in the form of probability scores in an informative manner. The evaluation results show that the model achieves an accuracy of 93%, and all system features function properly based on *black-box* and API testing using *Postman*. Therefore, PROSPECTA can support the effectiveness of sales teams in determining prospect priorities and improve the efficiency of data-driven marketing.

Keywords – *Predictive Lead Scoring; Machine Learning; Random Forest; Dashboard; Banking*

1. PENDAHULUAN

Berkembangnya teknologi informasi di sektor perbankan mendorong pemanfaatan data nasabah secara lebih optimal untuk mendukung pengambilan keputusan berbasis data [1][2]. Dalam praktiknya institusi keuangan mulai mengadopsi pendekatan *Machine Learning* untuk menganalisis perilaku nasabah dan memprediksi ketertarikannya terhadap produk perbankan, seperti deposito berjangka [3][4].

Masalah utama yang sering dihadapi adalah *lead overload* dan kampanye pemasaran yang tidak efisien karena tim sales masih bekerja tanpa dukungan analisis data yang mendalam [5][6]. Untuk mengatasi hal tersebut, penggunaan *Predictive Lead Scoring* berbasis algoritma prediktif dinilai lebih efektif dibandingkan metode konvensional karena mampu menangani data dengan kompleksitas tinggi [7][8].

Pemilihan algoritma yang tepat seperti *Random Forest* menjadi sangat relevan dalam meningkatkan kualitas prediksi karena kemampuannya melakukan klasifikasi melalui mekanisme *bagging* yang stabil [7]. Selain aspek algoritma, penelitian terbaru sangat menekankan pentingnya tahap *preprocessing* data, khususnya penggunaan *Robust Scaler* untuk menstabilkan skala fitur yang memiliki variasi nilai ekstrem atau *outlier* [9]. Tahapan *preprocessing* yang tepat serta pemilihan Teknik klasifikasi yang sesuai terbukti mampu memberikan hasil yang lebih seimbang dalam penargetan potensi nasabah [10][9]. Optimasi model *Random Forest* mampu menghasilkan nilai error yang lebih kecil secara konsisten dan mempercepat waktu pelatihan model dibandingkan hanya menggunakan metode tunggal [11].

Penerapan metode peramalan berbasis komputasi juga telah terbukti mampu menghasilkan estimasi yang akurat pada objek penelitian Sheda Florist untuk meminimalkan risiko kerugian [12]. Meskipun penerapan *Machine Learning* terbukti efektif dalam mendukung keputusan strategis, Sebagian besar penelitian terdahulu masih berfokus pada pengujian performa algoritma secara terpisah [4].

Berdasarkan hal tersebut, penelitian ini mengembangkan sistem *Predictive Lead Scoring* (PROSPECTA) menggunakan Next.js yang menyajikan hasil prediksi dalam bentuk *dashboard*. Sistem ini diharapkan dapat membantu tim sales dalam melakukan prioritas prospek secara lebih efektif serta mendukung pengambilan keputusan berbasis data di sektor perbankan.

2. METODE PENELITIAN

2.1. Observasi

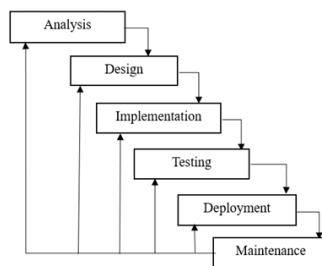
Observasi dalam penelitian ini dilakukan dengan mengumpulkan dan menganalisis data yang digunakan sebagai dasar pengembangan model *Machine Learning*. Data yang digunakan berasal dari *Bank Marketing Dataset* pada UCI Machine Learning Repository. Pada tahap ini dilakukan *Exploratory Data Analysis* untuk memahami karakteristik dan pola data, yang kemudian digunakan sebagai dasar dalam proses pengolahan data dan pembangunan model *Predictive Lead Scoring*.

2.2. Studi Literatur

Studi literatur dilakukan dengan mengumpulkan dan mempelajari berbagai referensi seperti jurnal ilmiah, buku, dan penelitian terdahulu yang relevan. Pada tahap ini, dipelajari konsep *Machine Learning*, *Predictive Lead Scoring*, serta teknologi pengembangan sistem seperti Next.js dan *dashboard* sebagai dasar dalam perancangan sistem.

2.3. Metode Pengembangan Sistem

Metode pengembangan sistem pada penelitian ini menggunakan *Software Development Life Cycle* (SDLC) dengan pendekatan Waterfall, yaitu model pengembangan perangkat lunak yang dilakukan secara bertahap dan sistematis mulai dari analisis kebutuhan, perancangan, implementasi, pengujian, *deployment*, hingga pemeliharaan [13]. Ilustrasi metode Waterfall ditunjukkan pada Gambar 1.



Gambar 1 Metode Waterfall [13]

2.3.1. Analisis Kebutuhan Sistem

Tahap analisis kebutuhan dilakukan berdasarkan permasalahan pemasaran produk deposito berjangka di sektor perbankan yang kemudian dibagi menjadi kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional meliputi pengelolaan akun pengguna (admin dan sales), penyajian *dashboard*, daftar prospek berdasarkan hasil prediksi, serta fitur pencarian, filter, dan pengelolaan status prospek.

Sementara itu, kebutuhan non-fungsional mencakup keamanan melalui autentikasi, performa sistem yang stabil, antarmuka yang responsif, serta kemudahan akses pada berbagai perangkat.

2.3.2. Desain Sistem

Tahap desain dilakukan untuk merancang arsitektur sistem serta antarmuka pengguna sesuai dengan kebutuhan yang telah dianalisis. Perancangan antarmuka dilakukan menggunakan Figma dengan fokus pada kemudahan penggunaan (*user-friendly*), konsistensi tampilan, serta penyajian visualisasi data yang informatif.

2.3.3. Implementasi

Tahap implementasi dilakukan dengan mengembangkan sistem sesuai desain yang telah dibuat, dengan frontend menggunakan Next.js, React.js, TypeScript, dan Tailwind CSS untuk menghasilkan antarmuka yang responsif, serta backend menggunakan Express.js dan MongoDB untuk pengelolaan data. Model *Machine Learning* menggunakan algoritma *Random Forest* yang telah melalui tahap *preprocessing* dengan *Robust Scaler* untuk menghasilkan probabilitas calon nasabah, yang kemudian disajikan melalui *dashboard* tanpa proses pelatihan ulang secara langsung di dalam sistem.

2.3.4. Pengujian

Tahap pengujian dilakukan untuk memastikan sistem berjalan sesuai dengan kebutuhan yang telah ditentukan melalui pengujian fungsional pada fitur utama seperti autentikasi pengguna, peran admin dan sales, pengelolaan data prospek, serta tampilan *dashboard*. Selain itu, pengujian API menggunakan Postman dilakukan untuk memastikan komunikasi antara *frontend* dan *backend* berjalan dengan baik.

Evaluasi model *Machine Learning* juga dilakukan menggunakan metrik *accuracy* untuk mengukur performa dalam memprediksi kemungkinan nasabah berlangganan deposito berjangka, sehingga sistem yang dikembangkan dapat digunakan secara optimal sebagai alat bantu prioritasasi prospek.

2.3.5. Deployment

Tahap *deployment* dilakukan untuk mempublikasikan sistem agar dapat diakses secara online, dengan *backend* diimplementasikan menggunakan platform Railway dan *frontend* menggunakan Vercel. Pemilihan platform ini didasarkan pada kemampuannya dalam mendukung pengembangan aplikasi web modern serta memudahkan integrasi antara *frontend* dan *backend*.

2.3.6. Maintenance (Pemeliharaan)

Tahap pemeliharaan dilakukan setelah sistem melalui proses *deployment*, yang meliputi perbaikan bug, peningkatan performa, serta penyempurnaan antarmuka berdasarkan hasil evaluasi penggunaan. Selain itu, dilakukan monitoring sistem untuk memastikan aplikasi tetap berjalan stabil dan dapat digunakan sebagai prototipe dalam mendukung proses prioritas prospek oleh tim sales perbankan.

3. HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil pengembangan sistem serta evaluasi model *Machine Learning* yang digunakan dalam penelitian. Pembahasan difokuskan pada proses pengolahan data, penerapan algoritma *Random Forest*, serta analisis kinerja model dalam memprediksi potensi nasabah berlangganan deposito berjangka. Selain itu, ditampilkan pula implementasi sistem dalam bentuk *dashboard* sebagai media visualisasi hasil prediksi untuk mendukung proses prioritas prospek.

3.1. Pre-Processing Data (Robust Scaller)

Tahap *preprocessing* data menggunakan metode *Robust Scaler* untuk menormalkan data dengan mempertimbangkan nilai median dan *Interquartile Range* (IQR), sehingga lebih tahan terhadap pengaruh nilai ekstrem (*outliers*). Metode ini bertujuan untuk menstabilkan skala fitur agar proses pembelajaran model menjadi lebih optimal. Transformasi *Robust Scaler* dirumuskan pada persamaan (1):

$$X_{scaled} = \frac{X - median(X)}{Q3 - Q1} \quad (1)$$

Di mana :

- Q1 merupakan kuartil pertama
- Q3 merupakan kuartil ketiga
- IQR merupakan hasil dari Q3-Q1

Data yang digunakan dalam penelitian ini disajikan pada Tabel 1

Tabel 1. Data Penelitian

No	Age	Day	Campaign
1.	25	3	2
2.	29	5	2
....			
20169.	30	28	1
20170.	37	17	2

Selanjutnya, dilakukan perhitungan nilai *scaled* untuk setiap variabel menggunakan persamaan (1). Sebelum itu, terlebih dahulu dihitung nilai median dan IQR dari masing masing variabel. Berdasarkan hasil perhitungan, diperoleh nilai median untuk variable *age* sebesar 39 dan nilai IQR sebesar 16.

Sebagai contoh, untuk data *age* = 25, maka hasil normalisasi dapat dihitung sebagai berikut :

$$X_{scaled} = \frac{25 - 39}{16} = -0,875$$

Hasil tersebut menunjukkan bahwa nilai *age* berada dibawah *median*, sehingga menghasilkan nilai *scaled negative*. Proses yang sama dilakukan pada seluruh data dan variabel lainnya, seperti *day* dan *campaign*, sehingga diperoleh hasil normalisasi menggunakan *Robust Scaler* yang disajikan pada Tabel 2

Tabel 2. Hasil Perhitungan Robust Scaler

No	Age_Scaled	Day_Scaled	Campaign_Scaled
1.	-0,875	-5.5	0
2.	-0.625	-4.5	0
...			
20169.	-0.5625	7	-0.5
20170.	-0.125	1.5	0

Hasil perhitungan tersebut menunjukkan bahwa proses normalisasi telah berhasil menstabilkan skala data tanpa dipengaruhi secara signifikan oleh nilai ekstrem. Dengan demikian, data yang telah dinormalisasi menggunakan *Robust Scaler* menjadi lebih siap digunakan pada tahap pemodelan *Machine Learning*.

3.2. Algoritma Random Forest

Random Forest merupakan algoritma yang bekerja dengan membangun banyak pohon keputusan (*decision tree*) secara paralel untuk menghasilkan prediksi yang lebih stabil dan akurat. Keunggulan utamanya terletak pada mekanisme *bagging* yang efektif mengurangi resiko *overfitting* serta kemampuannya dalam menangani dataset berskala besar dengan fitur yang beragam [4].

Dalam proses klasifikasi, *Random Forest* menggunakan metode *majority voting*, yaitu memilih kelas yang paling sering muncul dari seluruh hasil prediksi pohon keputusan. Secara sistematis, prediksi akhir dapat dirumuskan pada Persamaan (2):

$$\hat{y} = \text{mode}(h_1(x), h_2(x), \dots, h_n(x)) \quad (2)$$

Di mana:

- $h_i(x)$ merupakan hasil prediksi dari pohon ke- i terhadap data input x
- n adalah jumlah pohon yang digunakan dalam model.
- Fungsi *mode* digunakan untuk menentukan kelas yang paling sering muncul sebagai hasil akhir prediksi

Selain itu, probabilitas suatu kelas (misalnya kelas “yes”) dihitung berdasarkan jumlah pohon yang memprediksi kelas tersebut dibandingkan dengan total pohon yang digunakan. Perhitungan probabilitas dirumuskan pada persamaan (3):

$$P(\text{yes}) = \frac{\sum_{i=1}^n [\text{Tree}_i = \text{"yes"}]}{n} \quad (3)$$

Contoh perhitungan manual *Random Forest* menggunakan 3 pohon keputusan ditunjukkan pada Tabel 3. Setiap data diprediksi oleh masing – masing pohon (*Tree 1*, *Tree 2*, *Tree 3*), kemudian dilakukan proses *majority voting* untuk menentukan hasil akhir.

Tabel 3. Perhitungan Random Forest

age	day	campaign	y	Tree 1	Tree 2	Tree 3	Prediksi Random Forest	Benar/Tidak	Probabilitas
25	3	2	yes	yes	yes	yes	yes	yes	100%
29	5	1	yes	yes	yes	yes	yes	yes	100%
56	6	2	no	no	yes	yes	yes	no	67%
38	18	3	yes	yes	no	yes	yes	yes	67%
33	26	1	no	yes	no	yes	yes	no	67%
60	25	1	yes	no	no	yes	no	no	33%
57	5	1	no	no	yes	yes	yes	no	67%
54	7	3	no	no	yes	yes	yes	no	67%
34	21	2	no	yes	no	yes	yes	no	67%
63	14	2	yes	no	yes	yes	yes	yes	67%

Sebagai contoh pada salah satu data diperoleh hasil prediksi dari tiga pohon yaitu (*yes, yes, no*), sehingga hasil akhir adalah *yes* berdasarkan fungsi *mode*. Probabilitas kelas "*yes*" dihitung sebagai berikut :

$$P(\text{yes}) = \frac{2}{3} = 0.67 \text{ (67\%)}$$

Sedangkan jika seluruh pohon menghasilkan prediksi yang sama, misalnya (*yes, yes, yes*), maka probabilitasnya adalah :

$$P(\text{yes}) = \frac{3}{3} = 1 \text{ (100\%)}$$

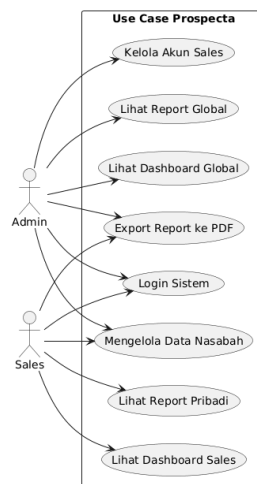
Hasil ini menunjukkan bahwa semakin banyak pohon yang memprediksi suatu kelas, maka semakin tinggi tingkat keyakinan (probabilitas) terhadap hasil prediksi tersebut.

3.3. Implementasi Dashboard

Tahap implementasi *dashboard* dilakukan untuk menyajikan hasil prediksi *Machine Learning* dalam bentuk antarmuka yang mudah digunakan oleh pengguna, yaitu admin dan sales. Sistem dikembangkan berbasis web menggunakan Next.js sehingga dapat diakses secara fleksibel melalui berbagai perangkat.

3.3.1. Use Case Diagram

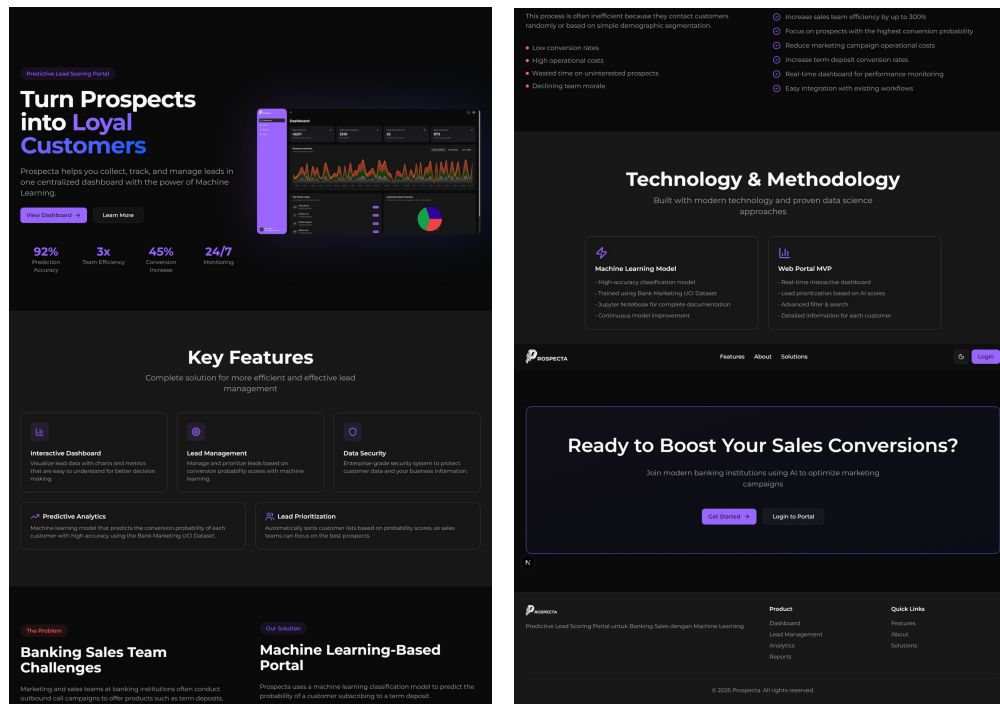
Alur interaksi pengguna dengan sistem ditunjukkan pada *use case diagram* pada Gambar 2, yang menggambarkan dua peran utama yaitu admin dan sales. Admin memiliki akses untuk mengelola akun, melihat laporan dan *dashboard* secara *global*, serta melakukan ekspor data. Sementara itu, sales memiliki akses untuk mengelola data nasabah, melihat laporan pribadi, serta menggunakan hasil prediksi dalam aktivitas tindak lanjut prospek.



Gambar 2. Use Case Diagram

3.3.2. Landing Page Prospects

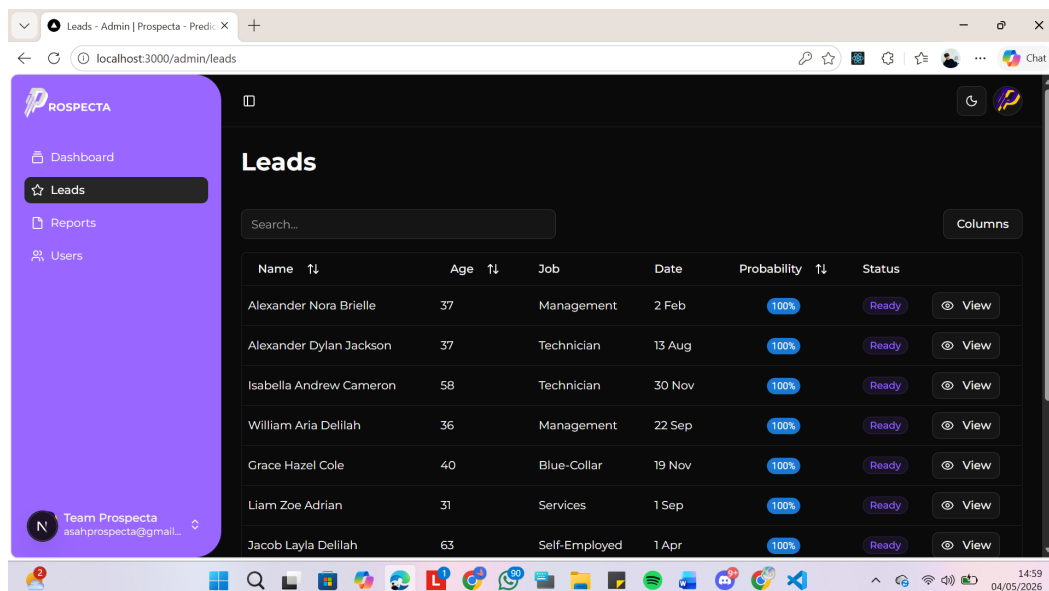
Implementasi antarmuka diawali dengan halaman utama (*landing page*) yang ditunjukkan pada Gambar 3, yang berfungsi sebagai tampilan awal *dashboard* dan akses menuju fitur utama melalui proses *autentikasi* pengguna.



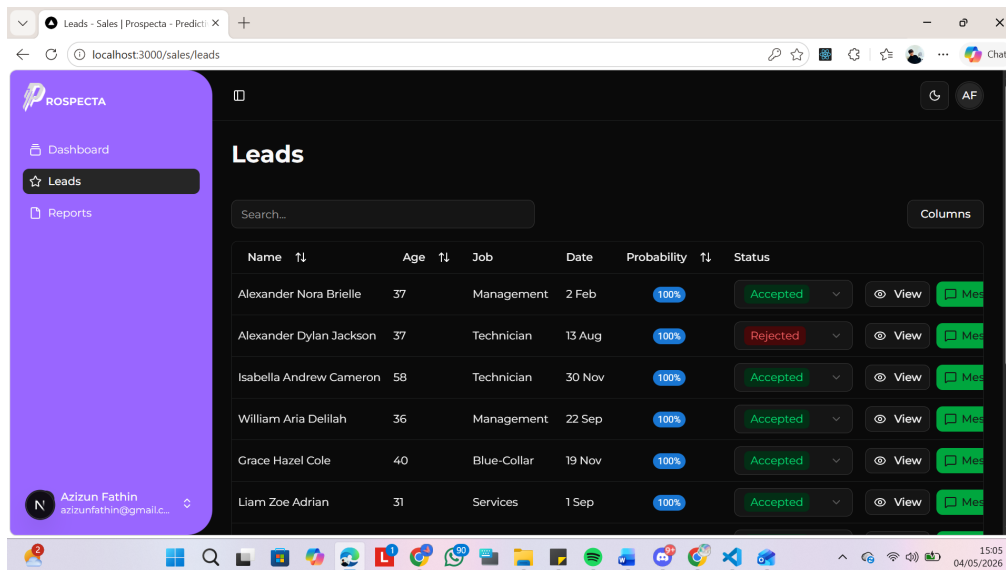
Gambar 3. Landing Page

3.3.3. Halaman Leads (Tampilan Hasil Probabilitas)

Halaman leads ditampilkan dalam dua versi sesuai dengan peran pengguna. Pada Gambar 4 ditunjukkan halaman leads untuk admin yang berfungsi untuk melihat data nasabah berserta hasil prediksi probabilitas tanpa fitur tindak lanjut. Sementara itu pada Gambar 5 ditunjukkan halaman leads untuk sales yang tidak hanya menampilkan data dan hasil prediksi, tetapi juga menyediakan fitur tindakan seperti pengiriman email dan panggilan telepon kepada nasabah



Gambar 4 Halaman Leads Admin



Gambar 5. Halaman Leads Sales

Perbedaan fitur pada kedua peran tersebut menunjukkan bahwa *dashboard* dirancang untuk mendukung kebutuhan masing – masing pengguna, di mana admin berfokus pada monitoring data secara keseluruhan, sedangkan sales berfokus pada tindak lanjut prospek berdasarkan hasil prediksi yang ditampilkan.

3.4. Pengujian Sistem

Tahap pengujian sistem dilakukan untuk memastikan seluruh komponen yang dikembangkan dapat berjalan sesuai dengan fungsi dan kebutuhan yang telah ditentukan. Dalam penelitian ini, pengujian sistem dibagi menjadi tiga bagian, yaitu pengujian model *Machine Learning* menggunakan metrik *accuracy*, pengujian fungsional sistem untuk memastikan setiap fitur berjalan dengan baik, serta pengujian API menggunakan Postman untuk memvalidasi komunikasi antara *frontend* dan *backend*.

3.4.1. Pengujian Model (Accuracy)

Pengujian model *Machine Learning* dilakukan untuk mengevaluasi kinerja algoritma *Random Forest* dalam memprediksi kemungkinan nasabah berlangganan deposito berjangka. Pada penelitian ini, evaluasi model menggunakan metrik *accuracy* yang mengukur tingkat ketepatan prediksi model terhadap keseluruhan data. Secara matematis, *accuracy* dirumuskan pada persamaan (4):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

Dimana :

- TP (*True Positive*) merupakan jumlah data *positif* yang di prediksi benar
- TN (*True Negative*) adalah jumlah data *negatif* yang di prediksi benar
- FP (*False Positive*) adalah jumlah data *negatif* yang salah di prediksi sebagai *positif*
- FN (*False Negative*) merupakan jumlah data *positif* yang di prediksi sebagai *negatif*

Berdasarkan hasil *confusion matrix* pada Gambar 8, diperoleh nilai *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), dan *True Positive* (TP) yang dirangkum pada Tabel 4.

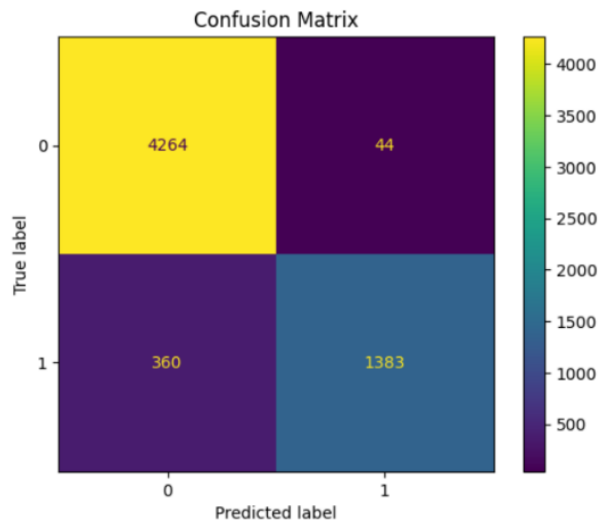
Tabel 4. Nilai Confusion Matrix

Kategori	Nilai
<i>True Negative</i> (TN)	4264
<i>False Positive</i> (FP)	44
<i>False Negative</i> (FN)	360
<i>True Positive</i> (TP)	1383

Maka nilai *accuracy* dapat dihitung sebagai berikut :

$$Accuracy = \frac{1383+4264}{1383+4264+44+360} = \frac{5647}{6051} = 0,93$$

Hasil perhitungan tersebut menunjukkan bahwa model memiliki tingkat akurasi sebesar 93%, yang berarti model mampu memprediksi data dengan tingkat ketepatan yang cukup tinggi.



Gambar 6. Confusion Matrix Random Forest

Selain itu, hasil evaluasi model juga ditunjukkan melalui *classification report* pada Gambar 6, yang menampilkan nilai *precision*, *recall*, dan *f1-score* untuk masing – masing kelas. Berdasarkan hasil tersebut, model menunjukkan performa baik, khususnya pada kelas negatif (0) dengan *recall* sebesar 0,99, serta tetap mampu mengidentifikasi kelas *positif* (1) dengan cukup baik.

--- Laporan Klasifikasi ---					
	precision	recall	f1-score	support	
0	0.92	0.99	0.95	4308	
1	0.97	0.79	0.87	1743	
accuracy			0.93	6051	
macro avg	0.95	0.89	0.91	6051	
weighted avg	0.94	0.93	0.93	6051	
ROC-AUC Score: 0.9234					

Gambar 9. Classification Report

3.4.2. Pengujian Blackbox Testing

Blackbox testing dilakukan untuk memastikan bahwa setiap fitur pada *dashboard* dapat berjalan sesuai dengan kebutuhan yang telah ditentukan. Pengujian ini berfokus pada kesesuaian *input* dan *output* tanpa memperhatikan struktur kode program yang digunakan. Setiap fitur diuji berdasarkan skenario tertentu untuk memastikan bahwa sistem memberikan respon yang sesuai. Hasil *blackbox testing* dirangkum pada Tabel 5.

Tabel 5. Hasil Blackbox Testing

No	Fitur/Halaman yang Diuji	Role	Skenario Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
1.	Login	Admin	Input email dan <i>password</i> yang <i>valid</i>	Berhasil masuk ke <i>dashboard</i> prospecta	Sesuai	Berhasil
2.	Login	Sales	Input email dan <i>password</i> yang <i>valid</i>	Berhasil masuk ke <i>dashboard</i> prospecta	Sesuai	Berhasil
3.	Dashboard	Admin	Mengakses halaman <i>dashboard</i>	Data analitik tampil dengan benar (<i>Dashboard global</i>)	Sesuai	Berhasil
4.	Dashboard	Sales	Mengakses halaman <i>dashboard</i>	Data tampil sesuai akun sales (<i>Dashboard pribadi</i>)	Sesuai	Berhasil
5.	Leads	Admin	Melihat data nasabah dan hasil prediksi	Data dan probabilitas tampil	Sesuai	Berhasil
6.	Leads	Sales	Melihat data nasabah dan melakukan tindak lanjut	Data, probabilitas fitur email, dan <i>call</i> tersedia	Sesuai	Berhasil

Berdasarkan hasil pengujian yang telah dilakukan, seluruh fitur pada *dashboard* prospecta berjalan sesuai dengan fungsinya. Hal ini menunjukkan bahwa *dashboard* telah memenuhi kebutuhan fungsional yang ditentukan dan dapat digunakan dengan baik oleh pengguna.

3.4.3. Pengujian Postman

Pengujian API dilakukan untuk memastikan bahwa komunikasi antara *frontend* dan *backend* berjalan dengan baik. Pengujian ini dilakukan menggunakan postman dengan menguji beberapa *endpoint* utama seperti autentikasi pengguna, pengambilan data nasabah, serta penampilan hasil prediksi.

Hasil pengujian menunjukkan bahwa setiap endpoint mampu memberikan respon yang sesuai dengan permintaan. Dengan demikian, dapat disimpulkan bahwa integrasi antara *frontend* dan *backend* pada *dashboard* prospecta telah berjalan dengan baik dan mendukung fungsi *dashboard* secara keseluruhan.

4. KESIMPULAN

Kesimpulan Berdasarkan hasil penelitian dan pembahasan, dapat disimpulkan bahwa:

- 1) Sistem *Predictive Lead Scoring* (PROSPECTA) berhasil dikembangkan untuk membantu prioritisasi prospek nasabah deposito berjangka melalui pendekatan berbasis *Machine Learning*.
- 2) Algoritma *Random Forest* yang dioptimasi dengan preprocessing *Robust Scaler* mampu menghasilkan performa yang baik dengan tingkat akurasi sebesar 93%.
- 3) Implementasi *dashboard* berbasis Next.js mampu menampilkan hasil prediksi dalam bentuk probabilitas, sehingga memudahkan tim sales dalam mengidentifikasi prospek potensial secara lebih cepat dan tepat.
- 4) Hasil pengujian sistem menunjukkan bahwa seluruh fitur berjalan dengan baik, dibuktikan melalui pengujian fungsional (*blackbox*) dan pengujian API menggunakan Postman.
- 5) Penelitian selanjutnya disarankan untuk menambahkan variasi model serta penggunaan metrik evaluasi lain agar hasil prediksi menjadi lebih komprehensif.

DAFTAR PUSTAKA

- [1] M. Rasikh, A. Riyyasy, W. Nouval Aghniya, and H. Tantyoko, "LEDGER: Journal Informatic and Information Technology Penerapan Algoritma Machine Learning Untuk Memprediksi Term Deposit Nasabah Perbankan," 2023.
- [2] M. Pamuk and M. Schumann, "Towards AI Dashboards in Financial Services: Design and Implementation of an AI Development Dashboard for Credit Assessment," *Mach. Learn. Knowl. Extr.*, vol. 6, no. 3, pp. 1720–1761, Sep. 2024, doi: 10.3390/make6030085.

- [3] E. A. Apriadi and M. Bisri, "Bank Customer Decision Prediction on Term Deposit Products Using Random Forest Algorithm on Bank Marketing Campaign Data," *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 7, no. 2, pp. 534–543, May 2025, doi: 10.47709/cnahpc.v7i2.5801.
- [4] A. Jadli, M. Hain, and A. Hasbaoui, "Artificial intelligence-based lead propensity prediction," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 3, pp. 1281–1290, Sep. 2023, doi: 10.11591/ijai.v12.i3.pp1281-1290.
- [5] A. M. Zaki, N. Khodadadi, W. H. Lim, and S. K. Towfek, "Predictive Analytics and Machine Learning in Direct Marketing for Anticipating Bank Term Deposit Subscriptions," *American Journal of Business and Operations Research*, vol. 11, no. 1, pp. 79–88, 2024, doi: 10.54216/AJBOR.110110.
- [6] M. Wu, P. Andreev, and M. Benyoucef, "Association for Information Systems Association for Information Systems Smart Sales: Amplifying the Power of Predictive Lead Scoring in Smart Sales: Amplifying the Power of Predictive Lead Scoring in B2B Sales B2B Sales," 2024. [Online]. Available: <https://aisel.aisnet.org/pacis2024>
- [7] Sindhu, "Stacking Ensemble Learning: Combining XGBoost, LightGBM, CatBoost, and AdaBoost with Random Forest Meta Model," Oct. 30, 2025. doi: 10.21203/rs.3.rs-7944070/v1.
- [8] I. I. Deshmukh and A. Ansari, "International Journal of Innovative Research in Science Engineering and Technology (IJIRSET) Intelligent Marketing Campaign Response Prediction: A Comparative Evaluation of Logistic Regression, Random Forest, and XG Boost with SMOTE Augmentation," *Certified Journal |*, vol. Volume 15, Mar. 2026, doi: 10.15680/IJIRSET.2026.1503259.
- [9] T. Gori, A. Sunyoto, and H. Al Fatta, "Preprocessing Data dan Klasifikasi untuk Prediksi Kinerja Akademik Siswa," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 1, pp. 215–224, Feb. 2024, doi: 10.25126/jtiik.20241118074.
- [10] N. Nuraeni, "Klasifikasi Data Mining untuk Prediksi Potensi Nasabah dalam Membuat Deposito Berjangka Data Mining Classification for Predicting Customer Potential in Making Term Deposits," *Jurnal Ilmiah Intech: Information Technology Journal of UMUS*, vol. 3, no. 01, pp. 65–75, 2021.
- [11] D. Susilo, Diah Ruswanti, Supriyanta, and Wawan Nugroho, "Comparison Of Principal Component Analysis And Random Forest Algorithm For Predicting Housing Prices," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 11, no. 2, pp. 353–361, Nov. 2025, doi: 10.33480/jitk.v11i2.7256.
- [12] W. C. F. Saputri, H. Khusnuliawati, and F. Fitriyadi, "Pembangunan Sistem Forecasting Berbasis Web Menggunakan Metode Holt-Winters Exponential Smoothing Untuk Meningkatkan Akurasi Kebutuhan Jumlah Panen Bunga Potong (Studi Kasus: Shenda Florist Surabaya)," 2024. [Online]. Available: <http://dx.doi.org/>
- [13] A. Nurseptaji, "Implementasi Metode Waterfall Pada Perancangan Sistem Informasi Perpustakaan," *Jurnal Dialektika Informatika (Detika)*, vol. 1, no. 2, pp. 49–57, May 2021, doi: 10.24176/detika.v1i2.6101.