

Aplikasi Phyton Manajemen Analisa Strategi Serta Pemantauan Elektabilitas Untuk Pemenangan Calon DPRD Menggunakan Algoritma *Random Forest*

Python Application for Strategy Analysis Management and Electability Monitoring for DPRD
Candidate Winning Based on a Web Platform Using the Random Forest Algorithm

Nala Syakhiya Nasution¹, Fitriana Harahap²

^aProgram Studi Informatika Universitas Potensi Utama, Jl.K.L Yos Sudarso KM 6.5 Tj.Mulia, Medan, 20241, Indonesia

^bProgram Studi Informatika Universitas Potensi Utama, Jl.K.L Yos Sudarso KM 6.5 Tj.Mulia, Medan, 20241, Indonesia

E-mail: nsyakhiya213@gmail.com¹, fitrianaarahap1@gmail.com²

ABSTRAK

Pemilihan anggota DPRD merupakan wujud nyata pelaksanaan demokrasi di tingkat daerah. Namun, proses pemantauan elektabilitas calon legislatif masih didominasi oleh metode manual yang lambat dan kurang responsif terhadap dinamika pemilih. Penelitian ini bertujuan untuk merancang dan membangun aplikasi berbasis *web* yang mampu memantau dan menganalisis data elektabilitas secara real-time dengan menggunakan algoritma *Random Forest*. Sistem ini tidak hanya menyajikan data survei, tetapi juga menyediakan prediksi elektabilitas dan visualisasi interaktif untuk mendukung strategi kampanye yang adaptif dan berbasis data. Hasilnya diharapkan dapat membantu tim kampanye dalam pengambilan keputusan yang lebih cepat, akurat, dan strategis

Kata Kunci : Elektabilitas, DPRD, Strategi Kampanye, Random Forest, Aplikasi Web, Analisis Data

ABSTRACT

The election of Regional People's Representative Council (DPRD) members is a concrete manifestation of democracy at the regional level. However, the process of monitoring legislative candidate electability is still dominated by manual methods that are slow and less responsive to voter dynamics. This research aims to design and build a web-based application capable of monitoring and analyzing electability data in real time using the Random Forest algorithm. This system not only presents survey data but also provides electability predictions and interactive visualizations to support adaptive, data-driven campaign strategies. The results are expected to assist campaign teams in making faster, more accurate, and more strategic decisions.

Keywords : Electability, DPRD, Campaign Strategy, Random Forest, Web Application, Data Analysis.

Disubmit:

*Info Artikel
Direview:*

Diterima :

Copyright © 2025 – senadimu. All rights reserved.

1. PENDAHULUAN

Pemilihan anggota DPRD (Dewan Perwakilan Rakyat Daerah) merupakan salah satu bentuk pelaksanaan demokrasi di Indonesia yang memberikan ruang bagi masyarakat untuk memilih wakil rakyat yang akan menyuarakan aspirasi mereka di tingkat daerah. Dalam proses ini, keberhasilan seorang calon tidak hanya bergantung pada popularitas semata, tetapi juga pada strategi kampanye yang terstruktur, efektif, dan berbasis pada data yang akurat.

Selama ini, pemantauan elektabilitas calon legislatif umumnya dilakukan melalui survei manual yang membutuhkan waktu lama dalam pengumpulan, pengolahan, dan analisis data. Keterbatasan ini berdampak pada keterlambatan pengambilan keputusan oleh tim kampanye, serta kurangnya kemampuan untuk merespons dinamika pemilih secara cepat. Selain itu, banyak lembaga survei dan tim sukses yang masih mengandalkan pendekatan konvensional tanpa sistem terintegrasi yang mampu menyajikan data secara *real-time* dan menyeluruh.

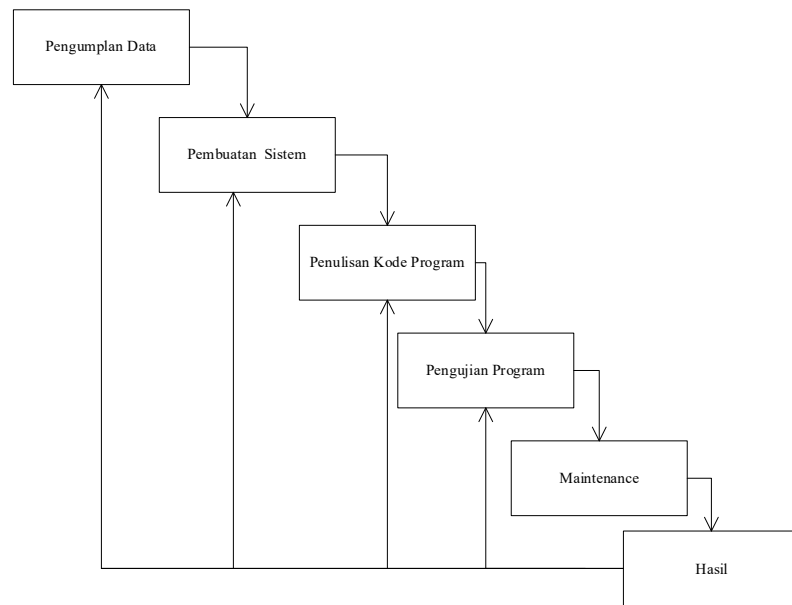
Di era *digital* dan *data-driven* saat ini, dibutuhkan suatu sistem informasi yang mampu mengelola dan menganalisis data elektabilitas secara cepat, akurat, dan terintegrasi. Salah satu teknologi yang potensial untuk mendukung analisis tersebut adalah algoritma ***Random Forest***, yang merupakan salah satu metode *machine learning* yang unggul dalam menangani data kompleks dan menghasilkan prediksi yang stabil. Algoritma ini mampu melakukan klasifikasi berdasarkan berbagai variabel seperti usia, jenis kelamin, wilayah, hingga preferensi politik pemilih. Dengan kekuatan *ensemble*-nya, *Random Forest* sangat cocok untuk menganalisis data survei yang bersifat multidimensional dan variatif.

Selain kemampuan analisis, aspek kemudahan akses dan kolaborasi juga menjadi kebutuhan utama dalam pengelolaan strategi kampanye. Oleh karena itu, pengembangan aplikasi berbasis ***platform web*** menjadi solusi yang ideal karena memungkinkan tim kampanye untuk memantau data elektabilitas secara *real-time*, berkolaborasi lintas perangkat, dan mengakses visualisasi data dalam bentuk grafik, tabel, peta wilayah dukungan, serta laporan strategis yang mudah dipahami.

Masalah lain yang sering dihadapi adalah kurangnya sistem yang terintegrasi dengan fitur prediksi elektabilitas dan analisis sentimen publik secara langsung. Hal ini menyebabkan tim kampanye kesulitan dalam menyesuaikan strategi secara cepat berdasarkan opini publik atau perubahan tren dukungan. Untuk mengatasi permasalahan tersebut, dibutuhkan suatu sistem teknologi yang tidak hanya mampu menampilkan data survei, tetapi juga menganalisisnya untuk menghasilkan wawasan strategis yang mendalam.

2. METODE

Dalam metode penelitian ini, penulis menerapkan berbagai teknik analisis dan klasifikasi untuk mengidentifikasi dan menyelesaikan permasalahan yang relevan. Langkah-langkah sistematis yang dilakukan untuk mencapai tujuan perancangan penelitian ditunjukkan pada Gambar 1 berikut,



Gambar 1. Kerangka Waterfall

Dalam pengembangannya metode *Waterfall* memiliki beberapa tahapan yaitu : *requirement* (analisis kebutuhan), *design* sistem (*system design*), *coding*, pengujian program, pemeliharaan sistem

1. Pengumpulan data

Peneliti melakukan pengumpulan data yang berisi tentang hal-hal yang harus ada pada hasil perancangan agar mampu menyelesaikan masalah yang ada sesuai tujuan. Data yang dibutuhkan dalam melakukan perancangan sistem adalah data calon DPRD dan data estimasi jawaban untuk tiap TPS, data pengguna dan bahasa pemrograman yang digunakan untuk membuat aplikasi adalah Phyton. Di dalam menyelesaikan penelitian ini penulis menggunakan 2 (dua) metode studi yaitu :

1. Studi Lapangan

Studi lapangan dilakukan oleh penulis dengan mengadakan studi langsung ke lapangan untuk mengumpulkan data yaitu peninjauan langsung ke lokasi studi. Adapun teknik pengumpulan data yang dilakukan penulis adalah :

a. Pengamatan (*Observation*)

Penulis melakukan pengamatan terhadap data produk dan data calon DPRD dan data estimasi jawaban untuk tiap TPS pada Kecamatan Binjai Selatan.

b. Wawancara (*Interview*)

Teknik ini secara langsung bertatap muka dengan pihak bersangkutan untuk mendapatkan penjelasan dari masalah-masalah yang sebelumnya kurang jelas yaitu tentang mekanisme sistem yang digunakan pada perusahaan dan juga untuk meyakinkan bahwa data yang diperoleh dikumpulkan benar-benar akurat.

2. Studi Kepustakaan (*Library Research*)

Penulis melakukan studi pustaka untuk memperoleh data-data yang berhubungan dengan penulisan skripsi dari berbagai sumber bacaan seperti buku panduan pembuatan aplikasi Phyton, Random Forest, manajemen data, dan buku atau jurnal konsep prediksi.

2. Pembuatan Sistem

Secara umum Manajemen Analisa Strategi Serta Pemantauan Elektabilitas Untuk Pemenangan Calon DPRD Berbasis *Platform Web* Menggunakan Algoritma *Random Forest* menggunakan model perancangan *Unified Modelling Language* yaitu *use case diagram*, *class diagram*, *activity diagram* dan *sequence diagram*.

3. Metode

Penulis memilih metode *Random Forest* dalam merancang Manajemen Analisa Strategi Serta Pemantauan Elektabilitas Untuk Pemenangan Calon DPRD menggunakan metode *Random Forest*.

4. Pengujian Program

Pada tahap ini dilakukan pengujian aplikasi secara menyeluruh, meliputi pengujian fungsional dan pengujian ketahanan sistem. Pengujian secara *black box (interface)* yaitu pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja.

5. Maintenace

Perangkat lunak yang susah disampaikan kepada pengguna sistem pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan baru.

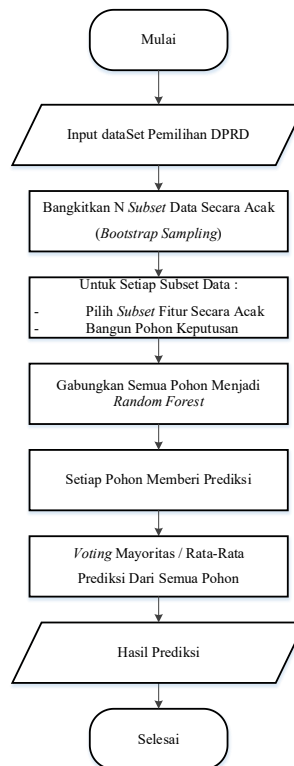
6. Hasil

Peneliti merupakan pengguna sistem pada Estimasi Manajemen Analisa Strategi Serta Pemantauan Elektabilitas Untuk Pemenangan Calon DPRD menggunakan metode *Random Forest*.

3. HASIL DAN PEMBAHASAN

3.1. Pembahasan

Flowchart ini menggambarkan alur kerja yang sistematis dari awal hingga akhir dalam membangun dan menggunakan model *Random Forest* untuk melakukan prediksi. Dapat dilihat pada Gambar 2 sebagai berikut:



Gambar 2. *Flowchart* Algoritma *Random Forest*

Bagian ini membahas secara komprehensif algoritma *Random Forest* yang diimplementasikan dalam sistem, mencakup prinsip kerjanya, metrik matematika yang digunakan, hingga simulasi perhitungan manual dengan sampel data riil untuk menunjukkan proses pengambilan keputusan.

1. Data Calon DPRD

Berikut ini merupakan data calon DPRD Pada kecamatan Binjai Selatan adalah sebagai berikut :

Tabel 1. Data Calon

No	Nama Calon	Partai	Asal Daerah
----	------------	--------	-------------

1	Benny Aulia Sanjaya	Demokrat	Rambung Barat
2	M. Ali Imron rhRP	Golkar	Tanah Seribu
3	M. Attan	Pdip	Binjai Estate
4	Gamatra Satria	PPP	Rambung Timur
5	M. Zimmy Daemawan	Nasdem	Tanah Merah
6	Efendi Ginting	Pks	Tanah Merah

2. Cara Kerja Metode *Random Forest*

Random Forest adalah sebuah algoritma *ensemble learning* yang berbasis pada *Decision Trees*. Konsep utamanya adalah membangun “hutan” dari banyak Decision Tree individual yang berbeda dan menggabungkan prediksi mereka. Tujuannya adalah untuk mendapatkan hasil prediksi yang lebih akurat, stabil, dan robust (kuat terhadap variasi data) dibandingkan dengan menggunakan satu Decision Tree tunggal. Prosesnya dibagi menjadi dua fase utama:

A. Fase Pelatihan (Membangun Hutan)

1. Bootstrap *Aggregating (Bagging)*:

- Dari dataset pelatihan asli yang berisi N baris data (sampel), algoritma akan secara independen membuat k buah dataset baru, yang disebut *bootstrap samples*.
- Setiap bootstrap sample ini juga akan memiliki N baris data, yang dibentuk dengan cara mengambil sampel secara acak dengan pengembalian (*sampling with replacement*) dari dataset asli.
- Implikasi: Karena pengambilan sampel dengan pengembalian, setiap bootstrap sample akan memiliki beberapa baris data dari dataset asli yang muncul berulang kali, sementara beberapa baris lainnya mungkin tidak muncul sama sekali dalam sampel tersebut. Hal ini secara sengaja memperkenalkan variasi dan keragaman pada data yang digunakan untuk melatih setiap *Decision Tree*.

2. Pembangunan Hutan (*Forest Building*) dengan *Random Feature Selection*:

- Untuk setiap bootstrap sample yang telah dibuat, algoritma membangun sebuah *Decision Tree* secara independen. Jadi, jika telah ditentukan untuk membangun *estimators* (jumlah pohon) sebanyak k, maka akan ada k Decision Trees yang dilatih.
- Inovasi Kunci (*Random Feature Selection*): Saat membangun setiap *Decision Tree*, pada setiap *node* (titik percabangan) di dalam tree, algoritma tidak mengevaluasi *semua* fitur yang tersedia untuk menemukan pemisah terbaik. Alih-alih, ia hanya akan memilih subset fitur secara acak (misalnya, jumlah fitur yang dipilih secara acak seringkali akar kuadrat dari total fitur atau sejumlah fitur acak lainnya).
- Dari subset fitur acak yang telah dipilih tersebut, algoritma kemudian mencari pemisah terbaik (fitur dan nilai ambang) yang paling efektif dalam membagi data. Pemilihan pemisah terbaik ini didasarkan pada metrik kemurnian node.

3. Kriteria Pemisahan Node: *Gini Impurity* (Matematika Inti)

- Pada setiap node *Decision Tree*, algoritma perlu memutuskan fitur mana dan nilai ambang mana yang akan digunakan untuk membelah data menjadi dua cabang yang lebih murni (*homogen*). Salah satu metrik yang paling umum digunakan untuk mengukur “ketidakmurnian” (atau *heterogenitas*) sebuah node adalah Gini Impurity.

Rumus *Gini Impurity* (G):

$$G = 1 - \sum_{i=1}^C (p_i)^2$$

Di mana:

- C adalah jumlah kelas unik yang ada dalam node tersebut (misalnya, dalam kasus ini ada 2 kelas: 'Dipilih' dan 'Tidak Dipilih').
- p_i adalah proporsi (fraksi) sampel yang termasuk dalam kelas i pada node tersebut.

- c. Tujuan Algoritma: Algoritma selalu mencari pemisah (fitur dan nilai ambang) yang menghasilkan penurunan Gini Impurity terbesar setelah pemisahan, atau dengan kata lain, yang paling meningkatkan kemurnian node anak.
- d. Gini Impurity dari Pemisahan (*Weighted Average*): Jika sebuah node dipisah menjadi cabang kiri dan kanan, Gini Impurity gabungan dari pemisahan tersebut dihitung sebagai rata-rata tertimbang:

$$G_{split} = \frac{N_{left}}{N} \times G_{left} + \frac{N_{right}}{N} \times G_{right}$$

Di mana:

- N_{left} dan N_{right} adalah jumlah sampel di cabang kiri dan kanan setelah pemisahan.
 - N adalah total jumlah sampel di node sebelum pemisahan.
 - G_{left} dan G_{right} adalah Gini Impurity dari masing-masing cabang.
- e. Penurunan Gini Impurity (Information Gain): Peningkatan kemurnian yang diperoleh dari pemisahan disebut Gini Gain:

$$Gain = G_{parent} - G_{split}$$

Algoritma akan memilih pemisah yang menghasilkan nilai *Gain* terbesar.

- f. Proses rekursif pembangunan tree berlanjut hingga *tree* mencapai kedalaman maksimum yang ditentukan (max depth), atau jumlah sampel minimum di daun (min samples _leaf), atau semua daun murni.

B. Fase Prediksi (Melakukan *Voting*)

1. *Input Data Baru*: Sebuah baris data baru yang karakteristiknya ingin diprediksi (misalnya, apakah pemilih akan memilih calon tertentu atau tidak) dimasukkan ke dalam model *Random Forest*.
2. *Prediksi Independen*: Data baru tersebut dilewatkan ke setiap *Decision Tree* yang ada di dalam "hutan". Setiap tree akan memproses data tersebut dari root hingga mencapai *leaf node* dan menghasilkan prediksinya sendiri secara independen.
3. *Agregasi Hasil (Voting)*:
 - Semua prediksi individu dari k *Decision Trees* dikumpulkan.
 - Untuk tugas klasifikasi, prediksi akhir dari *Random Forest* ditentukan oleh mayoritas suara (*majority voting*). Kelas yang paling banyak dipilih oleh tree-tree tersebut akan menjadi prediksi akhir. Contohnya, jika dari 100 tree, 70 tree memprediksi 'Dipilih' dan 30 tree memprediksi 'Tidak Dipilih', maka hasil akhir *Random Forest* adalah 'Dipilih'.

C. Simulasi Perhitungan Manual dengan Sampel Data

Langkah-langkah kerja *Random Forest* menggunakan sampel data pemilih yang diberikan, dengan fokus pada perhitungan Gini Impurity.

Langkah 1: Sampel Data Asli

Berikut adalah data yang akan digunakan sebagai dasar simulasi. Setiap baris memiliki fitur-fitur seperti usia, kelurahan, pendidikan, pekerjaan, isu utama, tingkat pengenalan, menerima _bant calon id, dan target is chosen.

Tabel 2. Sampel Data Pemilih

No	Nama	Jk	Usia	Desa/Kel	Rt	Rw	Tps
1	Abdul Azez	L	34	Pujidadi	0	0	Tps 1
2	Abdur Rahman	L	38	Pujidadi	0	0	Tps 1
3	Ade Dewi	P	32	Pujidadi	0	0	Tps 1
4	Adi Syahputra	L	53	Pujidadi	0	0	Tps 1
5	Agus Ariyanto	L	42	Pujidadi	0	0	Tps 1
6	Agus Gunawan Perangin-Angin	L	42	Pujidadi	0	0	Tps 1
7	Agus Sahputra	L	36	Pujidadi	0	0	Tps 1

No	Nama	Jk	Usia	Desa/Kel	Rt	Rw	Tps
8	Ahmad Iqbal Nasution, Se	L	45	Pujidadi	0	0	Tps 1
9	Aili Aida	P	57	Pujidadi	0	0	Tps 1
10	Ainal Safitri	P	42	Pujidadi	0	0	Tps 1
11	Airin Nurul Hasanah, Amk	P	42	Pujidadi	0	0	Tps 1
12	Aisah	P	36	Pujidadi	0	0	Tps 1
13	Ajeng Sunarsih	P	23	Pujidadi	0	0	Tps 1
14	Ajuarun	L	59	Pujidadi	0	0	Tps 1
15	Alam Subuh	L	55	Pujidadi	0	0	Tps 1
16	Aldi Ardiansyah	L	18	Pujidadi	0	0	Tps 1
17	Ananda Foransyah Nasution	L	24	Pujidadi	0	0	Tps 1
18	Andi Irawan	L	32	Pujidadi	0	0	Tps 1
19	Andini Febrianty	P	17	Pujidadi	0	0	Tps 1
20	Andriansyah	L	31	Pujidadi	0	0	Tps 1

- Singkatan Kolom: no: no Pemilih, nama pemilih, JK: Jenis Kelamin, U: Usia, KEL: Kelurahan, TPS: Tempat Pemungutan Suara, RT dan RW.
- Singkatan Jenis Kelamin: L: Laki-laki, P: Perempuan
- Singkatan Kelurahan: BE: Binjai Estate, RD: Rambung Dalam, RB: Rambung Barat, RT: Rambung Timur, TS: Tanah Seribu, TM: Tanah Merah, P: Pujidadi
- Ringkasan Target pada Dataset Asli:
- Jumlah kelas 'Dipilih' (1): 9
- Jumlah kelas 'Tidak Dipilih' (0): 28
- Total Sampel: 37

Langkah 2: Membuat *Bootstrap Samples (Bagging)*

Untuk ilustrasi, kita akan mengasumsikan membangun sebuah *Random Forest* dengan 3 *Decision Trees*. Setiap *Decision Tree* akan dilatih pada *bootstrap sample* yang terdiri dari 37 baris data yang diambil secara acak dengan pengembalian dari dataset asli.

- *Bootstrap Sample* untuk *Tree 1* (hasil acak):
Baris nomor: 2, 15, 5, 2, 23, 9, 1, 11, 24, 7, 19, 14, 2, 8, 20, 1,16, 25, 4, 13, 22, 10, 6, 17, 24, 30, 3, 35, 37, 26, 31, 12, 18, 27,28, 33, 34
Setelah proses *sampling* ini, *Bootstrap Sample 1* secara efektif memiliki:
 - 27 sampel kelas 'Tidak Dipilih' (0)
 - 10 sampel kelas 'Dipilih' (1)
 - Total sampel di node ini: 37
- *Bootstrap Sample* untuk *Tree 2* (hasil acak yang berbeda):
Baris nomor: 21, 5, 12, 18, 9, 3, 14, 25, 17, 1, 6, 11, 23, 20, 8,15, 4, 10, 22, 19, 5, 13, 7, 16, 2, 36, 32, 29, 37, 30, 26, 31,33, 34, 35, 27, 28
Sampel ini memiliki 25 kelas 'Tidak Dipilih' dan 12 kelas 'Dipilih'.
- *Bootstrap Sample* untuk *Tree 3* (hasil acak yang berbeda lagi):
Baris nomor: 10, 1, 1, 24, 8, 15, 19, 3, 21, 12, 5, 18, 25, 7, 4,11, 17, 23, 6, 9, 14, 2, 20, 22, 13, 36, 37, 26, 31, 32, 29, 28,33, 34, 35, 30, 27
Sampel ini memiliki 29 kelas 'Tidak Dipilih' dan 8 kelas 'Dipilih'.

Langkah 3: Membangun *Decision Tree* 1 secara Detail (Termasuk Perhitungan Gini Impurity)

Bootstrap Sample 1 yang memiliki 27 sampel 'Tidak Dipilih' dan 10 sampel 'Dipilih' (Total 37 sampel).

1. *Gini Impurity* Node Induk (*Root Node* dari *Tree 1*):

Proporsi Kelas 0 (p_0): $27/37 \approx 0.730$

Proporsi Kelas 1 (p_1): $10/37 \approx 0.270$

Gini Impurity (G_{parent}):

$$\begin{aligned} G_{parent} &= 1 - (0.730)^2 - (0.270)^2 \\ &= 1 - 0.5329 - 0.0729 \\ &= 0.3942 \end{aligned}$$

2. Pemilihan Fitur Acak di *Root Node*:

Dari 10 fitur yang tersedia, algoritma secara acak memilih sejumlah fitur untuk dipertimbangkan di node ini (3 fitur). Misalkan yang terpilih adalah: menerima bantuan, tingkat pengenalan dan usia.

3. Mencari Pemisah Terbaik dari Fitur yang Dipilih Secara Acak:

Algoritma akan mengevaluasi setiap fitur terpilih dan ambang batasnya untuk menemukan pemisahan yang menghasilkan *Gini Gain* terbesar.

a. Kandidat Split A: Fitur menerima _bantuan (menerima _bantuan == 1?)

– *Simulasi Pembagian Data*:

* Cabang Kiri (menerima bantuan = 1): 11 sampel (9 'Dipilih', 2 'Tidak Dipilih')

$$\begin{aligned} p_0 &= 2/11 \approx 0.182 \\ p_1 &= 9/11 \approx 0.818 \\ G_{left} &= 1 - (0.182)^2 - (0.818)^2 \\ &= 1 - 0.0331 - 0.6691 \\ &= 0.2978 \end{aligned}$$

* Cabang Kanan (menerima bantuan = 0): 26 sampel (1 'Dipilih', 25 'Tidak Dipilih')

$$\begin{aligned} p_0 &= 25/26 \approx 0.962 \\ p_1 &= 1/26 \approx 0.038 \\ G_{right} &= 1 - (0.962)^2 - (0.038)^2 \\ &= 1 - 0.9254 - 0.0014 \\ &= 0.0732 \end{aligned}$$

– *Gini Impurity* Pemisahan ($G_{split A}$):

$$G_{split_A} = \frac{11}{37} \cdot 0.2978 + \frac{26}{37} \cdot 0.0732 = 0.0886 + 0.0515 = 0.1401$$

– *Gini Gain* untuk Split A:

$$Gain_A = G_{parent} - G_{split A} = 0.3942 - 0.1401 = 0.2541$$

b. Kandidat Split B: Fitur tingkat pengenalan (tingkat _pengenalan == 'Paham Program'?)

– *Simulasi Pembagian Data*:

* Cabang Kiri ('Paham Program'): 8 sampel (4 'Dipilih', 4 'Tidak Dipilih')

$$\begin{aligned} p_0 &= 4/8 = 0.5 \\ p_1 &= 4/8 = 0.5 \\ G_{left} &= 1 - (0.5)^2 - (0.5)^2 = 0.500 \end{aligned}$$

* Cabang Kanan (Lainnya): 29 sampel (6 'Dipilih', 23 'Tidak Dipilih')

$$\begin{aligned} p_0 &= 23/29 \approx 0.793 \\ p_1 &= 6/29 \approx 0.207 \\ G_{right} &= 1 - (0.793)^2 - (0.207)^2 = 0.3284 \end{aligned}$$

– *Gini Impurity* Pemisahan ($G_{split B}$)

$$\begin{aligned} &= \frac{8}{37} \times 0.500 + \frac{29}{37} \times 0.3284 \\ &= 0.1081 + 0.2572 \\ &= 0.3653 \end{aligned}$$

– *Gini Gain* untuk Split B:

$$Gain_B = G_{parent} - G_{split B} = 0.3942 - 0.3653 = 0.0289$$

Keputusan Pemisah *Root Node*: Karena Gini Gain untuk Split A (0.2541) lebih besar dari Gini Gain Split B (0.0289), algoritma akan memilih menerima bantuan = 1 sebagai pemisah terbaik di *root node* dari Tree 1.

4. Membangun Cabang Berikutnya Secara Rekursif:

Cabang Kiri (menerima bantuan = 1): Node ini sekarang berisi 11 sampel (9 'Dipilih', 2 'Tidak Dipilih') dengan Gini Impurity $G_{\text{left}} = 0.2978$. Algoritma akan melanjutkan proses yang sama:

- 1) Memilih subset fitur acak baru (misal, usia, calon id, kelurahan).
- 2) Mencari pemisah terbaik (misal, usia ≤ 35) untuk node ini, menghitung Gini Gain, dan membuat percabangan lebih lanjut.
- 3) Proses ini berlanjut hingga kedalaman maksimum tercapai (max depth = 5 seperti di kode implementasi), atau node menjadi murni, atau jumlah sampel terlalu kecil.

Cabang Kanan (menerima bantuan = 0): Node ini berisi 26 sampel (1 'Dipilih', 25 'Tidak Dipilih') dengan Gini Impurity $G_{\text{right}} = 0.0732$. Karena Gini Impurity-nya sudah sangat rendah (mendekati 0), node ini kemungkinan besar akan menjadi *leaf node* yang memprediksi 'Tidak Dipilih' dengan kepercayaan diri tinggi, atau hanya memerlukan satu pemisah lagi untuk menjadi murni sepenuhnya.

Langkah 4: Ringkasan Pembangunan Decision Tree 2 dan Tree 3

Proses pembangunan Decision Tree 2 dan Tree 3 mengikuti alur yang sama seperti Tree 1, namun dengan dua perbedaan krusial yang menciptakan keragaman:

- 1) Dataset Bootstrap yang Berbeda: Setiap tree dilatih pada subset data yang sedikit berbeda.
- 2) Subset Fitur Acak yang Berbeda: Pada setiap node dari setiap tree, algoritma memilih subset fitur acak yang berbeda untuk dipertimbangkan.

Kedua faktor ini memastikan bahwa setiap Decision Tree dalam hutan memiliki "perspektif" dan aturan keputusan yang unik, yang sangat penting untuk mengurangi *variance* dan meningkatkan robustness keseluruhan Random Forest.

Langkah 5: Melakukan Prediksi untuk Data Baru

Sebuah data pemilih baru yang belum pernah terlibat dalam pelatihan (data dari daftar yang belum digunakan).

Data Pemilih Baru untuk Prediksi:

id_pemilih: 9999, jenis_kelamin: P, usia: 40, kelurahan: BINJAI ESTATE, tps: TPS 23, pendidikan: SMP, pekerjaan: Swasta, isu_utama: Ekonomi, calon_id: 4, tingkat_pengenalan: Tahu Nama, menerima_bantuan: 0

Data ini akan dilewatkan ke setiap Decision Tree yang telah dilatih:

1. Prediksi dari Tree 1 (berdasarkan struktur yang di bangun):
 - a) Mulai dari *root node*: menerima bantuan = 1? Nilai untuk data baru ini adalah 0 (Tidak).
 - b) Maka, data ini akan mengikuti cabang 'Tidak'.
 - c) Karena cabang 'Tidak' ini (pada asumsi kita) mayoritas mutlak adalah 'Tidak Dipilih',
 - d) Hasil Prediksi Tree 1: 0 (Tidak Dipilih).
2. Prediksi dari Tree 2 (berdasarkan struktur hipotetisnya):
 - Tree 2 memiliki aturan utama: "Jika isu utama == 'Ekonomi' DAN calon id == 4, maka 'Dipilih'".
 - Untuk data baru: isu_utama == 'Ekonomi'? → Ya. calon id == 4? → Ya.
 - Hasil Prediksi Tree 2: 1 (Dipilih).
3. Prediksi dari Tree 3 (berdasarkan struktur hipotetisnya):
 - a) Anggap Tree 3 memiliki aturan utama: "Jika kelurahan == 'BINJAI ESTATE' DAN usia > 35 DAN tingkat pengenalan == 'Tahu Nama', maka 'Dipilih'".
 - b) Untuk data baru: kelurahan == 'BINJAI ESTATE'? → Ya. usia > 35 (yaitu 40)? → Ya. tingkat pengenalan == 'Tahu Nama'? → Ya.
 - c) Hasil Prediksi Tree 3: 1 (Dipilih).

Langkah 6: *Agregasi Hasil Akhir (Voting)*

- Kumpulkan semua suara dari Decision Trees:
Tree 1: 0 (Tidak Dipilih)
Tree 2: 1 (Dipilih)
Tree 3: 1 (Dipilih)
- Lakukan Voting: Terdapat 2 suara untuk 'Dipilih' dan 1 suara untuk 'Tidak Dipilih'.
- Prediksi Final Random Forest: Kelas dengan suara mayoritas adalah 1 (Dipilih).

Kesimpulan

Simulasi perhitungan manual ini menunjukkan bagaimana Random Forest bekerja secara matematis:

- Keragaman melalui Bootstrap Sampling: Setiap tree melihat versi data yang sedikit berbeda, menciptakan perspektif yang beragam.
- Keacakan dalam Pemilihan Fitur: Pada setiap node, algoritma hanya mempertimbangkan subset fitur acak, bukan semua fitur, yang mengurangi overfitting.
- Optimasi Berbasis Gini Impurity: Setiap pemisahan dipilih berdasarkan kriteria matematika yang jelas untuk memaksimalkan kemurnian.
- Agregasi Demokratis: Keputusan akhir diambil berdasarkan voting mayoritas dari semua tree, mengurangi bias individu.

Kekuatan Random Forest terletak pada kombinasi ini: setiap tree individual mungkin tidak sempurna, tetapi kombinasi dari banyak tree yang beragam menghasilkan prediksi yang lebih robust dan akurat dibandingkan model tunggal.

3.2. Teknik Evaluasi

Tabel 3. Laporan Klasifikasi

	Precision	Recall	F1-score	Support
Tidak Dipilih	0.85	0.63	0.72	4143
Dipilih	0.20	0.46	0.28	828
Accuracy			0.60	4971
Macro avg	0.53	0.54	0.54	4971
Weighted avg	0.74	0.60	0.65	4971

Perhitungan Metrik Evaluasi Model

Berdasarkan laporan klasifikasi, berikut perhitungan akurasi, presisi, recall, dan F1-score untuk kelas "Dipilih":

1. Akurasi

Rumus:

$$\text{Akurasi} = (TP + TN) / (TP + TN + FP + FN)$$

Keterangan:

TP (*True Positive*): Prediksi benar untuk kelas positif (Dipilih).

TN (*True Negative*): Prediksi benar untuk kelas negatif (Tidak Dipilih).

FP (*False Positive*): Prediksi salah sebagai positif.

FN (*False Negative*): Prediksi salah sebagai negatif.

Langkah Perhitungan:

Total data = 4971 (dari laporan).

Akurasi = 0.5989 (dari laporan).

Maka, jumlah prediksi benar = Akurasi $\times$ Total data = $0.5989 \times 4971 \approx 2976$.

Akurasi = $2976 / 4971 \approx 0.5989$ atau 59.89%.

Hasil*: Akurasi = 59.89%.

2. Presisi (Precision)

Rumus:

$\text{Presisi} = \text{TP} / (\text{TP} + \text{FP})$

Langkah Perhitungan:

Untuk kelas "Dipilih":

- Presisi = 0.1976 (dari laporan).
- Support kelas "Dipilih" = 828 (jumlah data aktual positif).
- Recall = 0.4601 (dari laporan), sehingga $\text{TP} = \text{Recall} \times \text{Support} = 0.4601 \times 828 \approx 381$.
- Presisi = $\text{TP} / (\text{TP} + \text{FP}) = 381 / (381 + \text{FP}) = 0.1976$.
- $\text{FP} = (381 / 0.1976) - 381 \approx 1928 - 381 = 1547$.

$\text{Presisi} = 381 / (381 + 1547) \approx 0.1976$ atau 19.76%.

Hasil: Presisi = 19.76%.

3. Recall

$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$

Langkah Perhitungan:

Untuk kelas "Dipilih":

- Recall = 0.4601 (dari laporan).
- Support = 828 (jumlah data aktual positif).
- $\text{TP} = \text{Recall} \times \text{Support} = 0.4601 \times 828 \approx 381$.
- $\text{FN} = \text{Support} - \text{TP} = 828 - 381 = 447$.

Verifikasi: $\text{Recall} = 381 / (381 + 447) \approx 0.4601$ atau 46.01%.

Hasil: Recall = 46.01%.

4. F1-Score

$\text{F1-Score} = 2 \times (\text{Presisi} \times \text{Recall}) / (\text{Presisi} + \text{Recall})$

Langkah Perhitungan*:

Untuk kelas "Dipilih":

- Presisi = 0.1976, Recall = 0.4601.
- $\text{F1-Score} = 2 \times (0.1976 \times 0.4601) / (0.1976 + 0.4601)$.
- Hitung: $0.1976 \times 0.4601 \approx 0.0909$.
- $2 \times 0.0909 = 0.1818$.
- $\text{Presisi} + \text{Recall} = 0.1976 + 0.4601 = 0.6577$.
- $\text{F1-Score} = 0.1818 / 0.6577 \approx 0.2765$ atau 27.65%.

Hasil: F1-Score = 27.65%.

Kesimpulan

Akurasi: 59.89%

Presisi (kelas Dipilih): 19.76%

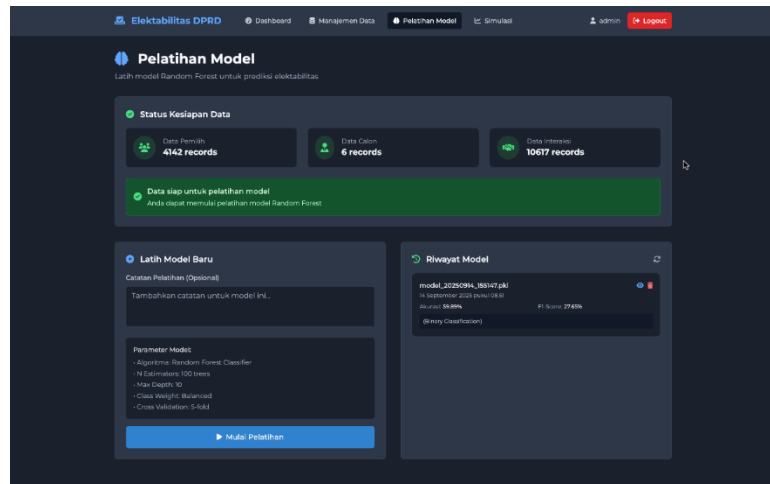
Recall (kelas Dipilih): 46.01%

F1-Score (kelas Dipilih): 27.65%

3.3. Hasil

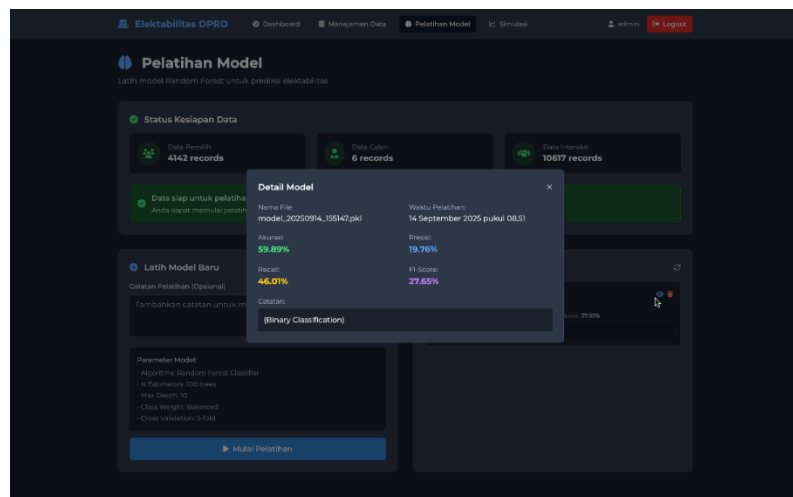
1. Tampilan *Form* Data Pelatihan Random Forest

Tampilan ini merupakan tampilan *form* input data pelatihan random forest yang berfungsi untuk menampilkan data-data pelatihan random forest. Berikut gambar *form* data pelatihan random forest ditunjukkan pada gambar 3:

Gambar 3. Tampilan *Form* Data Pelatihan Random Forest

2. Tampilan *Form* Data Panel Model

Tampilan ini merupakan tampilan *form* input data panel model yang berfungsi untuk menampilkan data-data panel model. Berikut gambar *form* data panel model ditunjukkan pada gambar 4 :

Gambar 4. Tampilan *Form* Data Panel Model

3. Tampilan *Form* Data Simulasi

Tampilan ini merupakan tampilan *form* simulasi yang berfungsi untuk menampilkan data-data simulasi. Berikut gambar *form* data simulasi ditunjukkan pada gambar 5 :

Gambar 5. Tampilan *Form* Data Simulasi

4. Tampilan *Form* Data Hasil Elektabilitas Keseluruhan

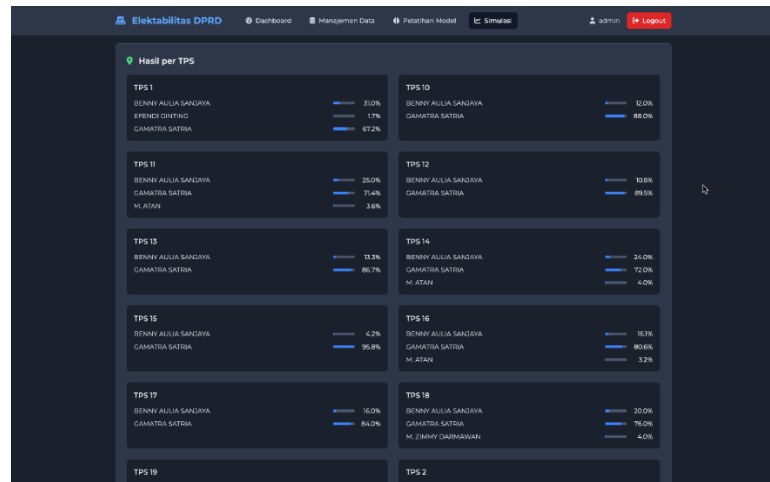
Tampilan ini merupakan tampilan *form* hasil elektabilitas keseluruhan yang berfungsi untuk menampilkan data-data hasil elektabilitas keseluruhan. Berikut gambar *form* data hasil elektabilitas keseluruhan ditunjukkan pada gambar 6 :



Gambar 6. Tampilan *Form* Data Hasil Elektabilitas Keseluruhan

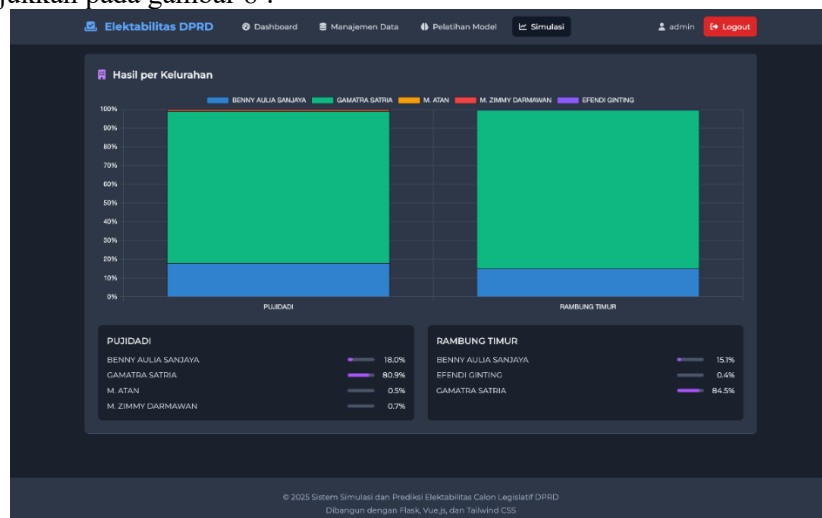
5. Tampilan *Form* Data hasil elektabilitas TPS

Tampilan ini merupakan tampilan *form* hasil elektabilitas TPS yang berfungsi untuk menampilkan data-data hasil elektabilitas TPS. Berikut gambar *form* data hasil elektabilitas TPS ditunjukkan pada gambar 7 :

Gambar 7. Tampilan *Form* Data hasil elektabilitas TPS

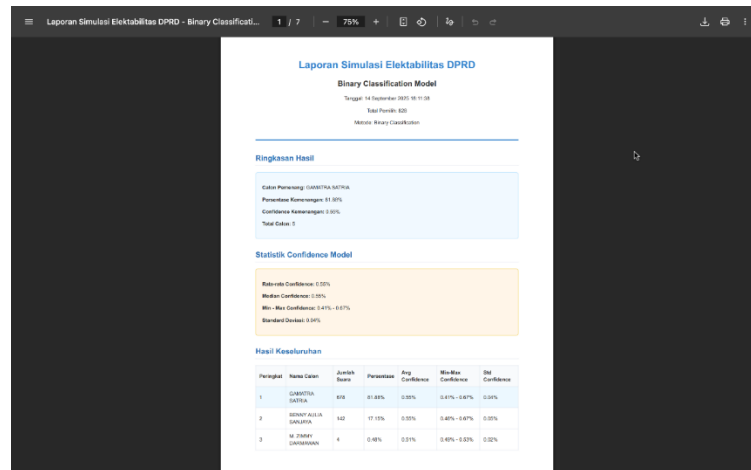
6. Tampilan *Form* Data hasil elektabilitas Kelurahan

Tampilan ini merupakan tampilan *form* hasil elektabilitas kelurahan yang berfungsi untuk menampilkan data-data hasil elektabilitas kelurahan. Berikut gambar *form* data hasil elektabilitas kelurahan ditunjukkan pada gambar 8 :

Gambar 8 Tampilan *Form* Data hasil elektabilitas Kelurahan

7. Tampilan *Form* Laporan Ringkasan Hasil

Form ini menampilkan laporan data ringkasan hasil, ketika *admin* memilih laporan pada option laporan ringkasan hasil maka program akan menampilkan laporan ringkasan hasil. Gambar tampilan *form* laporan ringkasan hasil dapat pada gambar 9 :



Laporan Simulasi Elektabilitas DPRD
Binary Classification Model
Tanggal: 14 Desember 2025 08:11:08
Total Peserta: 678
Metode: Binary Classification

Ringkasan Hasil

Kelompok Pemeringkat: GAMBARITA SATRIA
Partisipasi Keterserangan: 81.88%
Confidence Keterserangan: 0.88%
Total Skor: 5

Statistik Confidence Model

Kelompok Confidence: 0.88%
Min-Max Confidence: 0.88%
Min-Max Confidence: 0.88%
Standard Deviasi: 0.04%

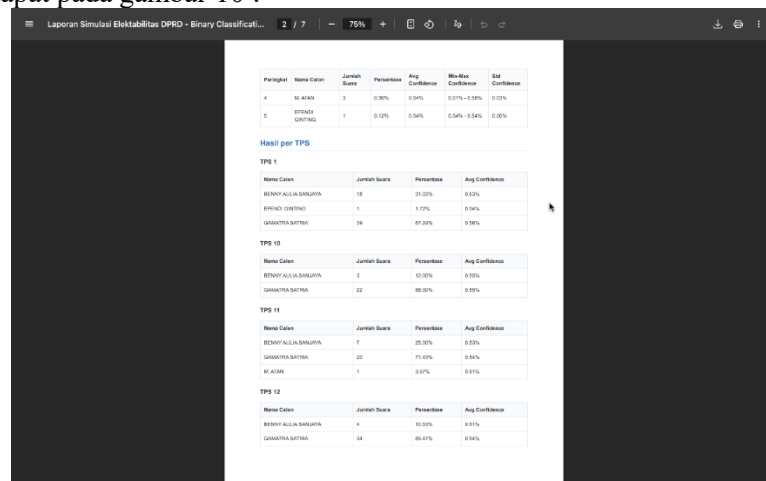
Hasil Keseluruhan

Partisipan	Nama Calon	Jumlah Skor	Persentase	Avg. Confidence	Min-Max Confidence	Std. Confidence
1	GAMBARITA SATRIA	678	81.88%	0.88%	0.88% - 0.88%	0.04%
2	BERNYAT ALLIA SANJAYA	142	17.15%	0.88%	0.88% - 0.88%	0.05%
3	SI. JEMBATI (GAMBARITA)	4	0.48%	0.88%	0.88% - 0.88%	0.02%

Gambar 9. Tampilan *Form* Laporan Ringkasan Hasil

8. Tampilan *Form* Laporan Breakdown TPS

Form ini menampilkan laporan data Breakdown TPS, ketika *admin* memilih laporan pada option laporan Breakdown TPS maka program akan menampilkan Breakdown TPS. Gambar tampilan *form* Breakdown TPS dapat pada gambar 10 :



Hasil per TPS

TPS 1

Nama Calon	Jumlah Skor	Persentase	Avg. Confidence
BERNYAT ALLIA SANJAYA	18	25.00%	0.88%
BERNYAT GEMENDI	1	1.77%	0.84%
GAMBARITA SATRIA	39	87.24%	0.96%

TPS 10

Nama Calon	Jumlah Skor	Persentase	Avg. Confidence
BERNYAT ALLIA SANJAYA	3	10.00%	0.88%
GAMBARITA SATRIA	22	80.00%	0.88%

TPS 18

Nama Calon	Jumlah Skor	Persentase	Avg. Confidence
BERNYAT ALLIA SANJAYA	7	25.00%	0.88%
GAMBARITA SATRIA	20	71.43%	0.84%
SI. JEMBATI	1	3.57%	0.81%

TPS 12

Nama Calon	Jumlah Skor	Persentase	Avg. Confidence
BERNYAT ALLIA SANJAYA	4	10.00%	0.81%
GAMBARITA SATRIA	34	80.00%	0.84%

Gambar 10. Tampilan *Form* Breakdown TPS

9. Tampilan *Form* Laporan Breakdown Kelurahan

Form ini menampilkan laporan data Breakdown kelurahan, ketika *admin* memilih laporan pada option laporan Breakdown kelurahan maka program akan menampilkan Breakdown kelurahan. Gambar tampilan *form* Breakdown kelurahan dapat pada gambar 11 :

The screenshot displays a web application interface for 'Laporan Simulasi Elektabilitas DPRD - Binary Classification'. The main content area is divided into three sections, each containing a table of election results:

- TPS 9:**

Nama Calon	Jumlah Suara	Persentase	Avg Confidence
DEWIAT NALA IN SARAJANA	15	80.00%	0.95%
CAHAYATIA GAYATIA	31	80.00%	0.95%
NI JENYAT CUMBERAWA	1	0.00%	0.95%
- PILKADA:**

Nama Calon	Jumlah Suara	Persentase	Avg Confidence
DEWIAT NALA IN SARAJANA	407	17.00%	0.95%
CAHAYATIA GAYATIA	483	80.00%	0.95%
NI JENYAT CUMBERAWA	1	0.00%	0.95%
NI JENYAT CUMBERAWA	4	0.00%	0.95%
- KABUPATEN TAMBORA:**

Nama Calon	Jumlah Suara	Persentase	Avg Confidence
DEWIAT NALA IN SARAJANA	30	10.00%	0.95%
DEWIAT NALA IN SARAJANA	1	0.00%	0.95%
CAHAYATIA GAYATIA	100	80.00%	0.95%

Below the tables, there are filters for 'Jenis Kabupaten' (Kabupaten) and 'Tipe TPS' (TPS). A section titled 'Penjelasan Model Binary Classification' provides a brief description of the model's purpose and accuracy.

Gambar 11. Tampilan *Form Breakdown* kelurahan

4. KESIMPULAN

Berdasarkan hasil penelitian dan implementasi sistem, dapat disimpulkan bahwa:

1. Telah berhasil dirancang dan dibangun sebuah aplikasi berbasis *web* yang mampu memantau dan mengelola data elektabilitas calon anggota DPRD secara terstruktur dan *real-time*. Sistem ini memungkinkan integrasi data survei dari berbagai wilayah dengan tampilan *dashboard* yang informatif dan mudah digunakan oleh tim kampanye dan analis politik.
2. Algoritma *Random Forest* telah berhasil diimplementasikan dalam sistem untuk melakukan analisis dan prediksi elektabilitas calon berdasarkan data survei yang telah di *input*. Algoritma ini mampu memproses data dengan banyak variabel dan menghasilkan prediksi dengan tingkat akurasi yang cukup tinggi dibandingkan metode klasifikasi konvensional lainnya.
3. Hasil pengujian sistem menunjukkan bahwa kinerja dan akurasi algoritma *Random Forest* dalam mengklasifikasikan data elektabilitas mencapai tingkat presisi yang baik. Algoritma mampu menangani data dalam jumlah besar, meminimalkan kesalahan prediksi, dan memberikan *insight* strategis yang dapat digunakan oleh tim kampanye dalam merumuskan langkah-langkah selanjutnya.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas Potensi Utama yang telah banyak memberikan masukan dan saran dalam penyelesaian penelitian ini.

REFERENSI

- [1] Abdul Rosid, U., & Maulana, H. (2024). Rancang Bangun Aplikasi Evaluasi Pengangkatan Kepala Desa Pada Badan Pemberdayaan Masyarakat Dan Pemerintah Desa Kabupaten Garut. *Jurnal Sistem Informasi Galuh*, 2(2), 93–102. <https://doi.org/10.25157/jsig.v2i2.3998>
- [2] Aggi Nurhapipah, Shiva Steviana, & Reza Mauldy Raharja. (2024). Strategi Kampanye Di Era Digital: Dengan Memanfaatkan Aplikasi X Dalam Pemilu 2024. *ALADALAH: Jurnal Politik, Sosial, Hukum Dan Humaniora*, 2(3), 272–281. <https://doi.org/10.59246/aladalah.v2i3.874>
- [3] Alexander Putra, J., Hermawan, N. A., Salsabilla, H., Annisa Wolio, N., Haykal Ramadhan, S., Erlangga, E., & Kusnendar, J. (2024). Rancang Bangun Aplikasi Berbasis Mobile Untuk Pendaftaran Pasien Klinik dan Penjualan Obat Dengan Model Waterfall. *Simkom*, 9(2), 152–163. <https://doi.org/10.51717/simkom.v9i2.393>
- [4] Alfiansyah, R. M., & Priono, H. (2024). Implementasi Sikadeka untuk Meningkatkan Efektivitas Audit Dana Kampanye Pemilu 2024 pada Partai Politik Kota X. *Jurnal Ilmiah Ekonomi Dan Manajemen*, 2(7), 615–623. <https://doi.org/10.61722/jiem.v2i7.2141>

- [5] Alfina, O., & Harahap, F. (2019). Pemodelan Uml Sistem Pendukung Keputusan Dalam Penentuan Kelas Siswa Siswa Tunagrahita 1. *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, 3(2), 143–150. <https://doi.org/10.46880/jmika.Vol3No2.pp143-150>
- [6] Fahrozi, W., Fitriana Harahap, Adhar, D., & Rasaki, F. I. (2023). Penerapan Metode Des dalam Memprediksi Persediaan Barang Berbasis Web. *Jurnal Unitek*, 16(2), 196–206. <https://doi.org/10.52072/unitek.v16i2.680>
- [7] Isanto, A. P., Wibowo, A. T., & Milad, M. K. (2020). Perancangan Sistem Informasi Monitoring Tim Sukses untuk Strategi Pemenangan Caleg Menggunakan Framework ITIL Versi 3. *Techno.Com*, 19(1), 34–44. <https://doi.org/10.33633/tc.v19i1.2758>
- [8] Mallisza, D., Hadi, H. S., & Aulia, A. T. (2022). Implementasi Model Waterfall Dalam Perancangan Sistem Surat Perintah Perjalanan Dinas Berbasis Website Dengan Metode SDLC. *Jurnal Teknik, Komputer, Agroteknologi Dan Sains*, 1(1), 24–35. <https://doi.org/10.56248/marostek.v1i1.9>
- [9] Priyambodo, A., Prihati, P., & Nurdianto, K. (2022). Perancangan Sistem Informasi Wisata Kopeng Berbasis Web dengan Metode Waterfall. *Jurnal Cakrawala Informasi*, 2(1), 59–68. <https://doi.org/10.54066/jci.v2i1.176>
- [10] Rahayu, T., & Wibisono, M. B. (2020). Rancang Bangun Sistem Pendukung Keputusan Menentukan Kandidat Calon Kepala Desa Dengan Metode Fanp Berbasis Web. *Jurnal Ilmiah Matrik*, 22(3), 350–356. <https://doi.org/10.33557/jurnalmatrik.v22i3.1134>
- [11] Rahmat Hidayat, Y., Haryanti, T., & Kurniawati, L. (2020). Sistem Informasi Pemilihan Umum Kepala Daerah Online Berbasis Android. *Information System for Educators and Professionals*, 4(2), 124–133. <https://ejournal-binainsani.ac.id/index.php/ISBI/article/view/1296/1121>
- [12] Samsoleh, E., & Rubiani, H. (2021). Rancang Bangun Sistem Penilaian Kinerja Pemerintah Daerah. *INFOTECH Journal*, 49–56. <https://doi.org/10.31949/infotech.v7i2.1503>
- [13] Sirait, A. (2021). Pemanfaatan Teknologi Berbasis Web Mobile Pada Aplikasi Survei Pilkada Asahan Online Sektor Media Sosial. *JOISIE (Journal Of Information Systems And Informatics Engineering)*, 5(2), 129–137. <https://doi.org/10.35145/joisie.v5i2.1242>
- [14] Supit, M., Lopian, M., & Tulung, T. (2022). PERAN MEDIA SOSIAL DALAM MENINGKATKAN ELEKTABILITAS CALON ANGGOTA LEGISLATIF DALAM PEMILIHAN UMUM TAHUN 2019 (Studi di Desa Mobuya Kecamatan Passi Timur Kecamatan Bolaang). *Eksekutif*, 2(1), 1–10.
- [15] Umma, S., & Holilah, H. (2023). Studi Popularitas, Akseptabilitas, dan Elektabilitas Muhaimin Iskandar pada Pilpres 2024 di Kalangan Alumni Pesantren Mamba'ul Ma'arif Denanyar Jombang. *JRP (Jurnal Review Politik)*, 13(1), 89–123. <https://doi.org/10.15642/jrp.2023.13.1.89-123>