Jurnal Sistem Dan Teknologi Informasi (JSTI)

https://journalversa.com/s/index.php/jsti Vol. 08, No. 1, Februari 2026

IMPLEMENTASI RANDOM FOREST PADA WEB FLASHCARD UNTUK MEMPREDIKSI TINGKAT KESULITAN KARTU

Setiyawan Purwanto[1], Rizqi Akbar Assamil[2], Muhammad Fatikha Rosikha Ilmi[3], Untung Rohwadi[4], Sigit Alsyukuri Wibawa[5]

Email: setiyawanpurwanto@gmail.com[1], rizqiakbarassamil@gmail.com[2], rosikha79@gmail.com[3], untung.unr@bsi.ac.id[4], sigit.stb@bsi.ac.id[5]

[1,2,3,4,5]Universitas Bina Sarana Informatika

ABSTRAK

Prediksi tingkat kesulitan kartu dalam penggunaan *flashcard* merupakan salah satu penerapan penting dalam pengembangan sistem pembelajaran adaptif dan kecerdasan buatan, khususnya untuk mendukung proses belajar yang lebih personal bagi pengguna. Namun, menentukan tingkat kesulitan secara akurat masih menjadi tantangan, terutama karena perbedaan gaya belajar pengguna dan variasi parameter yang dihasilkan dari interaksi mereka dengan kartu. Penelitian ini bertujuan untuk mengembangkan sistem prediksi tingkat kesulitan *flashcard* berbasis web menggunakan metode *Random Forest*, yang dikenal memiliki performa baik dalam klasifikasi data dengan banyak fitur. Sistem ini dirancang untuk mengumpulkan berbagai parameter penggunaan *flashcard*, seperti waktu menjawab, jumlah percobaan, lama baca kartu, dan selisih waktu antar sesi, kemudian mengolahnya menjadi dataset berlabel tingkat kesulitan. Model *Random Forest* dilatih menggunakan dataset tersebut melalui modul *Trainer3*, lalu diimplementasikan kembali sebagai fungsi *rule-based* berbentuk *if-else* dalam *JavaScript* sehingga dapat dijalankan secara langsung di *browser* tanpa memerlukan *server* atau model berat. Fungsi prediksi ini diterapkan pada halaman Final *Flashcard* untuk mengestimasi tingkat kesulitan setiap kartu secara otomatis berdasarkan parameter terbaru dari pengguna. Berdasarkan hasil pengujian, sistem mampu memberikan prediksi tingkat kesulitan yang konsisten dengan akurasi 88.64% pada dataset pengujian. Implementasi sistem ini diharapkan dapat menjadi solusi teknologi pembelajaran yang adaptif, ringan, dan efisien, serta berpotensi dikembangkan untuk meningkatkan pengalaman belajar berbasis personalisasi pada berbagai platform lainnya.

Kata Kunci: Flashcard Berbasis Web, Random Forest, Machine Learning, Pembelajaran Adaptif, Javascript.

ABSTRACT

Predicting the difficulty level of flashcards is an important application in the development of adaptive systems and artificial intelligence, particularly to support a more personalized learning process for users. However, accurately determining the difficulty level remains a challenge, primarily due to differences in user learning styles and the variation of parameters resulting from their interactions with the cards. This study aims to develop a web-based flashcard difficulty prediction system using the Random Forest method, which is known to perform well in classifying data with many features. This system is designed to collect various flashcard usage parameters, such as answer time, number of attempts, card reading time, and the time difference between sessions, then process them into a dataset labeled with difficulty levels. The Random Forest model is drilled using the dataset through the Trainer3 module, then reimplemented as a rule-based function in the form of an if-else in JavaScript so that It can be run directly in a browser without the need for servers or heavy models. This prediction function is implemented on the Final Flashcard page to automatically estimate the difficulty level of each card based on the latest user parameters. Based on test results, the system is able to provide consistent difficulty level predictions with an accuracy of 88.64% on the test dataset. The implementation of this system is expected to be an adaptive, lightweight, and efficient learning technology solution, and has the potential to be developed to enhance personalized learning experiences on various other platforms.

Keywords: *Web-Based Flashcards, Random Forest, Machine Learning, Adaptive Learning, Javascript.*

1. PENDAHULUAN

Pembelajaran mandiri berbasis digital semakin berkembang pesat seiring meningkatnya akses terhadap teknologi dan internet. Salah satu media pembelajaran yang banyak digunakan adalah *flashcard*, yaitu kartu yang berisi pasangan informasi seperti pertanyaan–jawaban atau konsep–definisi. *Flashcard* terbukti efektif dalam mendukung proses mengingat dan pengulangan (*spaced repetition*) karena dapat menyesuaikan ritme belajar pengguna serta memfasilitasi latihan berulang [7].

Namun, dalam praktiknya, setiap pengguna memiliki tingkat kesulitan yang berbeda terhadap setiap kartu. Sebagian kartu dianggap mudah, sementara yang lain memerlukan pengulangan lebih banyak. Pada banyak platform *flashcard*, penentuan tingkat kesulitan masih dilakukan secara manual oleh pengguna, sehingga dapat mengganggu konsentrasi, bersifat subjektif dan tidak konsisten. Ketiadaan sistem

otomatis yang mampu memprediksi tingkat kesulitan menyebabkan pengalaman belajar kurang optimal karena materi tidak dapat diprioritaskan sesuai kebutuhan masing-masing pengguna [1].

Perkembangan teknologi kecerdasan buatan memberikan peluang untuk mengatasi permasalahan tersebut. Salah satu algoritma machine learning yang banyak digunakan untuk klasifikasi adalah *Random Forest*, yaitu metode *ensemble learning* yang menggabungkan banyak *decision tree* untuk menghasilkan prediksi yang lebih stabil, akurat, dan tahan terhadap *overfitting* [3]. Kemampuannya dalam menangani data kompleks dan variabel beragam menjadikan algoritma ini sesuai untuk digunakan dalam sistem rekomendasi, termasuk prediksi tingkat kesulitan kartu pada platform belajar digital.

Dengan memanfaatkan interaksi pengguna terhadap *flashcard* seperti hasil tebakan (benar/salah), jumlah percobaan, waktu menjawab, lama membaca kartu, dan jeda waktu antar sesi model *Random Forest* dapat mempelajari karakteristik yang membuat suatu kartu dianggap mudah atau sulit. Prediksi tingkat kesulitan tersebut dapat membantu sistem mengatur jadwal pengulangan, memberikan rekomendasi latihan, serta mempersonalisasi pengalaman belajar bagi setiap pengguna.

Penelitian ini bertujuan mengimplementasikan algoritma *Random Forest* pada sebuah web *flashcard* untuk memprediksi tingkat kesulitan kartu secara otomatis. Sistem ini diharapkan mampu memberikan pengalaman belajar yang lebih adaptif, akurat, dan efisien, serta meningkatkan efektivitas pembelajaran mandiri. Selain itu, penelitian ini menjadi langkah awal dalam pengembangan platform *flashcard* cerdas yang mengintegrasikan machine learning sebagai dasar pengambilan keputusan.

2. TINJAUAN PUSTAKA

Penelitian Terdahulu

Penelitian yang mengkaji penerapan machine learning dalam sistem pembelajaran digital semakin berkembang pesat. Salah satunya adalah penelitian yang membahas penggunaan algoritma klasifikasi untuk memprediksi tingkat kesulitan materi belajar berdasarkan perilaku pengguna. Beberapa penelitian menunjukkan bahwa data interaksi pengguna, seperti waktu menjawab, tingkat kesalahan, dan pola pengulangan, dapat digunakan untuk memprediksi kesulitan suatu item pembelajaran secara akurat menggunakan algoritma *supervised learning* [4]. Temuan tersebut membuktikan bahwa

metode berbasis machine learning mampu meningkatkan personalisasi pembelajaran karena sistem dapat menyesuaikan materi berdasarkan kemampuan pengguna. Hal ini memberikan landasan bahwa pendekatan serupa dapat diterapkan pada platform *flashcard* untuk memprediksi tingkat kesulitan kartu secara otomatis.Selain itu, sejumlah penelitian juga menunjukkan bahwa metode *ensemble learning*, khususnya *Random Forest*, memiliki performa yang stabil dalam tugas klasifikasi karena mampu menggabungkan banyak *decision tree* untuk meningkatkan akurasi dan mengurangi *overfitting* [5]. Dibandingkan dengan algoritma tunggal seperti *decision tree*, *Random Forest* lebih tahan terhadap *noise* dan variasi data, sehingga banyak digunakan pada sistem rekomendasi dan prediksi tingkat kesulitan pada aplikasi pendidikan digital.

Random Forest

Random Forest adalah algoritma machine learning berbasis *ensemble learning*, yang menggabungkan sejumlah pohon keputusan (*decision tree*) untuk menghasilkan prediksi yang lebih akurat dan *robust*. Algoritma ini diperkenalkan oleh Leo Breiman sebagai pengembangan dari metode *bagging*, di mana setiap pohon dilatih menggunakan *subset* acak dari data dan fitur. Proses ini membuat setiap pohon memiliki variasi dan tidak saling bergantung, sehingga hasil voting dari seluruh pohon menghasilkan prediksi yang stabil [5].

Keunggulan utama *Random Forest* meliputi:

a. Kemampuan menangani data non-linear.
b. *Robust* terhadap *outlier* dan *noise*
c. Mampu mengukur pentingnya fitur (*feature importance*).
d. Cocok untuk dataset kecil hingga menengah.

Dalam konteks sistem *flashcard*, algoritma ini mampu belajar dari parameter seperti hasil tebakan, jumlah percobaan, waktu tebakan, lama baca kartu, dan selisih waktu antar sesi. Dengan demikian, *Random Forest* dapat memprediksi apakah sebuah kartu termasuk kategori "ulangi", "ragu", "cukup", atau "hafal" berdasarkan pola penggunaan.

Flashcard Digital dan Personalization Learning

Flashcard digital merupakan media pembelajaran yang banyak digunakan karena

efektivitasnya dalam meningkatkan retensi informasi melalui metode pengulangan bertahap (*spaced repetition*). Beberapa penelitian menunjukkan bahwa efektivitas *flashcard* akan meningkat apabila sistem mampu menyesuaikan tingkat kesulitan dan frekuensi pengulangan berdasarkan performa pengguna [5]. Platform modern seperti *Anki* atau *Quizlet* telah menerapkan mekanisme penilaian tingkat kesulitan, namun masih mengandalkan penilaian subjektif dari pengguna. Pendekatan berbasis machine learning dapat memperbaiki kekurangan tersebut karena prediksi tingkat kesulitan dilakukan secara otomatis melalui analisis pola interaksi pengguna.

Penerapan Machine Learning pada Sistem Pembelajaran

Penerapan machine learning pada sistem pendidikan (*Educational Data Mining*) banyak difokuskan pada prediksi performa belajar, rekomendasi materi, dan analisis perilaku pengguna. Hasil penelitian menunjukkan bahwa model klasifikasi mampu meningkatkan pengalaman belajar pengguna melalui adaptasi otomatis berdasarkan tingkat kesulitan materi [5]. Konsep ini menjadi dasar pengembangan sistem pada penelitian ini, di mana model *Random Forest* digunakan untuk memprediksi tingkat kesulitan *flashcard* secara otomatis berdasarkan dataset perilaku pengguna. Integrasi model ke dalam web *flashcard* memberi kemampuan adaptif yang lebih baik dan pengalaman belajar yang lebih personal.

3. METODE PENELITIAN

Metode penelitian ini menjelaskan seluruh tahapan yang dilakukan mulai dari proses pengumpulan data pengguna dan anotasi tingkat kesulitan pada kartu *flashcard*, pembangunan model prediksi menggunakan algoritma *Random Forest*, konversi model ke dalam bentuk fungsi *JavaScript*, hingga implementasi model pada platform web *flashcard*. Seluruh tahapan dilakukan secara sistematis untuk memastikan bahwa model mampu memberikan prediksi tingkat kesulitan kartu secara akurat dan dapat diintegrasikan dengan baik pada aplikasi berbasis web.

Tahapan Perancangan Sistem

a. Pengumpulan Data Pengguna

Tahap pertama penelitian adalah melakukan pengumpulan data interaksi pengguna dalam menggunakan *flashcard*. Data ini merupakan komponen penting untuk

membangun model prediksi. Dataset yang digunakan dalam penelitian ini dikumpulkan dari riwayat penggunaan *flashcard* bahasa Inggris-Indonesia dengan struktur sebagai berikut:

Fitur Input:

1. hasil_tebakan (*binary*: 0=salah, 1=benar)
2. jumlah_percobaan(*integer*)
3. waktu_lama_tebakan (detik)
4. lama_baca_kartu (detik)
5. selisih_waktu_sesi (jam antar sesi)

Label/Target:

tingkat_kesulitan dengan empat kelas:

1. "ulangi" (perlu pengulangan segera)
2. "ragu" (kurang yakin)
3. "cukup" (cukup dipahami)
4. "hafal" (sudah dikuasai)

Ukuran Dataset:

218 sampel dengan distribusi:

1. ulangi: 36 sampel
2. hafal: 123 sampel
3. cukup: 48 sampel
4. ragu: 11 sampel

b. Anotasi & Penentuan Label Tingkat Kesulitan

Tahap Kedua dalam pengolahan dataset adalah melakukan proses Pra-pemrosesan data (*data preprocessing*) untuk memastikan bahwa data siap digunakan dalam pelatihan model machine learning. Pra-pemrosesan ini dilakukan melalui beberapa langkah sebagai berikut:

1. Normalisasi

Data Proses normalisasi dilakukan untuk mengubah seluruh fitur yang masih dalam bentuk string atau kategori menjadi nilai numerik agar dapat diproses oleh algoritma *Random Forest*.

- Setiap label kategorikal seperti tingkat kesulitan ("ulangi", "ragu", "cukup", "hafal") dikonversi menjadi representasi numerik.
- Fitur-fitur lain yang sudah berupa angka dibiarkan dalam format aslinya, tetapi tetap dilakukan pengecekan konsistensi tipe data.

Normalisasi ini penting untuk menghindari kesalahan pemrosesan dan memastikan data dapat dipahami oleh algoritma.

2. Pembagian Dataset

Dataset dibagi menjadi dua bagian menggunakan komposisi standar:

- 80% data digunakan sebagai data training, yaitu untuk melatih model agar dapat mempelajari pola dari fitur input terhadap label.
- 20% data digunakan sebagai data testing**,** yaitu untuk menguji performa model secara objektif. Pembagian dilakukan dengan teknik *random shuffle* untuk memastikan sampel terdistribusi secara acak. Teknik ini membantu mencegah bias akibat urutan data dan memberikan hasil evaluasi yang lebih representatif.

3. **Handling *Missing Values***

Penanganan *Missing Values*

Pada tahap pemeriksaan data, tidak ditemukan adanya nilai kosong (*missing values*) dalam seluruh fitur maupun label. Dengan demikian, tidak diperlukan proses imputasi atau penanganan khusus lainnya. Seluruh sampel dapat digunakan dalam proses pelatihan dan pengujian model.

c. Pembuatan Model *Random Forest* di *Web-Based trainer*

Pengembangan model dilakukan menggunakan *web-based trainer* (*Trainer3.html*) yang mengimplementasikan algoritma *Random Forest* secara langsung di *browser*. Tahapan dalam pembuatan model:

1. Memuat dataset pengguna
2. Melakukan pembagian dataset (80% data latih, 20% data uji)
3. Membangun model *Random Forest* dengan parameter:
    - *n_estimators*: 5 pohon
    - *max_depth*: 12
    - *min_samples_split*: 3
    - *max_features*: 3
4. Melatih model menggunakan data latih

d. Pelatihan dan Validasi Model

Algoritma *Random Forest* dipilih karena:

1. Mampu bekerja dengan baik pada data kategorikal dan numerik
2. Kuat terhadap *overfitting*
3. Memberikan stabilitas hasil prediksi
4. Memiliki kemampuan melakukan klasifikasi *multi-class* (mudah, sedang, sulit)

Model dievaluasi menggunakan:

1. *Confusion Matrix*
2. *Precision*
3. *Recall*
4. *F1-Score*
5. Akurasi keseluruhan

Hasil evaluasi digunakan untuk memastikan bahwa model memiliki performa memadai sebelum diimplementasikan ke web.

e. Konversi Model *Random Forest* ke *JavaScript*

Agar dapat digunakan langsung di *browser*, model dikonversi dari struktur pohon keputusan menjadi fungsi *JavaScript*. Metode konversi yang digunakan adalah melalui web *trainer* yang menghasilkan kode *JavaScript* berbentuk fungsi *if-else* (*rule-based*).

Hasil akhir berupa:

1. *flashcard_model.js*` → file *JavaScript* yang berisi logika *Random Forest*

2. Fungsi `*predictDifficulty(row)*` → mengembalikan label "ulangi", "ragu", "cukup", atau "hafal"

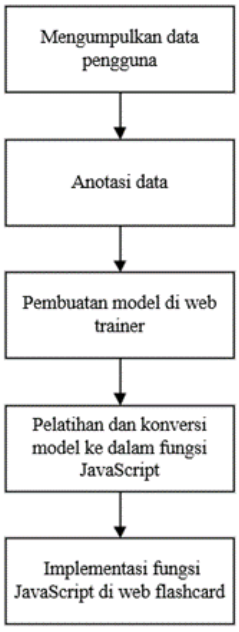
f. Implementasi Fungsi Prediksi di Web *Flashcard*

Setelah model dikonversi, tahap selanjutnya adalah penerapannya di web *flashcard*. Fungsi model dijalankan di *browser* untuk:

1. Mengambil data terbaru pengguna
2. Memilih fitur yang dibutuhkan model
3. Menjalankan prediksi kesulitan
4. Menampilkan hasil kepada pengguna
5. Mengatur ulang urutan *flashcard* berdasarkan tingkat kesulitan
    - Kartu "ulangi" tampil lebih sering
    - Kartu "hafal" dikurangi frekuensinya

Integrasi web dilakukan menggunakan:

1. *HTML*
2. *CSS*
3. *JavaScript*
4. Model *Random Forest* (converted version)



Gambar 1. Flowchart Sistem

Parameter Evaluasi

Guna mengukur efektivitas model *Random Forest* dalam mengestimasi tingkat kesulitan kartu pada sistem *web flashcard*, penelitian ini menerapkan sejumlah parameter evaluasi standar. Penilaian dilakukan melalui metrik klasifikasi yang lazim digunakan dalam riset pengembangan media edukasi berbasis data. Pendekatan ini dipilih karena selaras dengan metodologi pengujian pada penelitian *flashcard* digital sebelumnya yang menunjukkan hasil pengukuran performa secara komprehensif [3].

Precision **(Presisi)** Mengukur tingkat ketepatan model dalam memberikan prediksi suatu kelas.

$$\textit{Precision Rate} = \frac{f(TP)}{f(TP) + f(FP)}$$

- TP (*True Positive*): Model secara akurat memprediksi label “ulangi” sesuai data asli.
- FP (*False Positive*): Model keliru menetapkan label “ulangi” pada kartu yang yang label aslinya adalah 'hafal'.

Interpretasi : Presisi tinggi berarti model tidak sering salah menandai kartu sebagai "ulangi".

Relevansi : Jika banyak kartu dengan label asli "hafal" secara keliru diprediksi oleh model sebagai kelas "ulangi", maka nilai presisi pada kelas "ulangi" akan menurun. Secara praktis, ketidakakuratan ini akan mengganggu sistem adaptasi pembelajaran, karena pengguna akan dipaksa untuk terus mengulang materi yang sebenarnya sudah mereka kuasai sepenuhnya, sehingga proses belajar menjadi tidak efisien.

Recall Mengukur kemampuan model dalam menangkap seluruh data positif yang sebenarnya ada pada tiap kelas.

$$\textit{Recall} = \frac{f(TP)}{f(TP) + f(FN)}$$

FN (*False Negative*): Model memprediksi kelas 'hafal' atau 'cukup', padahal label asli kartu tersebut adalah 'ulangi'.

Interpretasi : *Recall* tinggi berarti model jarang melewatkan kartu yang benar-benar sulit dan mampu mendeteksi sebagian besar objek positif.

Relevansi Jika banyak kartu sulit diprediksi sebagai mudah/sedang, *recall* kelas "Sulit" menjadi rendah, sehingga model tidak mampu mengenali tingkat kesulitan yang sebenarnya.

F1-Score Rata-rata harmonis antara *Precision* dan *Recall*, cocok untuk data tidak seimbang.

$$F1\text{-}Score = 2 \times \frac{Precision\ Score \times Recall\ Score}{Precision\ Score + Recall\ Score}$$

Interpretasi : Skor F1 yang tinggi merepresentasikan keberhasilan model dalam menyeimbangkan antara ketepatan prediksi dan cakupan data yang berhasil dikenali. Sebaliknya, fluktuasi pada nilai ini mengindikasikan adanya ketidakkonsistenan model dalam memisahkan kategori tingkat kesulitan yang memiliki karakteristik serupa.

Relevansi : Mengingat adanya ketimpangan jumlah data (*class imbalance*) pada sistem *flashcard* ini, *F1-Score* menjadi tolok ukur yang lebih objektif dibandingkan akurasi biasa dalam memvalidasi keadilan performa model pada seluruh kelas.

Accuracy Mengukur jumlah prediksi benar dibanding keseluruhan data.

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

- TN (*True Negative*): Kondisi di mana model secara tepat memprediksi kelas non-target, misalnya memprediksi kelas 'hafal' pada data yang secara faktual adalah 'hafal'.

Interpretasi : Akurasi memberikan gambaran umum performa model, tetapi kurang ideal digunakan sendiri jika dataset tidak seimbang.

Relevansi: Pada sistem *flashcard,* kelas "Mudah" biasanya lebih banyak, sehingga akurasi perlu dipadukan dengan *F1-Score* dan *Confusion Matrix* agar evaluasi performa model lebih tepat.

Confusion Matrix

Memberikan evaluasi rinci mengenai jumlah prediksi benar dan salah untuk setiap kelas. Matriks ini menunjukkan pola kesalahan model dan membantu mengetahui kelas mana yang paling sering keliru diprediksi. *Confusion Matrix* berguna untuk menganalisis kelemahan model dan menjadi dasar perbaikan dataset maupun parameter model.

Macro-Average

Menghitung rata-rata metrik (*Precision, Recall, F1-Score*) untuk setiap kelas secara seimbang tanpa mempertimbangkan jumlah data pada masing-masing kelas. *Macro-Average* cocok digunakan ketika dataset tidak seimbang karena menilai performa tiap kelas secara adil.

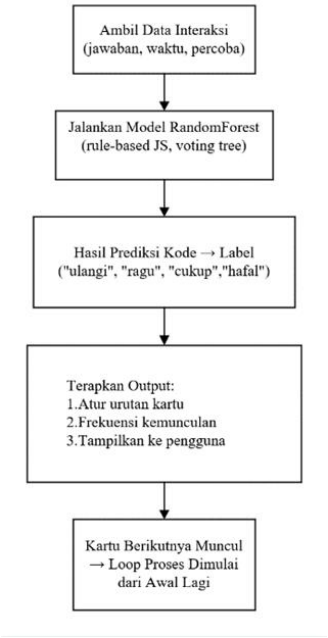
Output Prediksi Tingkat Kesulitan

Output prediksi dihasilkan secara langsung dari interaksi pengguna saat menggunakan web *flashcard*. Sistem memanfaatkan model *Random Forest* yang sudah dikonversi menjadi fungsi *JavaScript* berbasis *rule-based* (*if–else*). Dengan pendekatan ini, prediksi dapat berjalan cepat di sisi *frontend* tanpa membutuhkan *server* tambahan.

Alur Kerja Lengkap:

1. Pengambilan Parameter dari Interaksi Pengguna Web *flashcard* mengumpulkan data seperti waktu menjawab, jumlah kesalahan, riwayat jawaban, dan frekuensi kartu setiap kali pengguna berinteraksi. Parameter ini menjadi input bagi model.
2. Eksekusi Fungsi Prediksi *Random Forest* Model *Random Forest* yang sebelumnya dilatih diekspor menjadi aturan *if–else* di *JavaScript*. Setiap rule mewakili keputusan dari masing-masing tree dalam hutan. Saat parameter masuk, fungsi ini dijalankan untuk menentukan voting dari setiap tree. Hasil *Random Forest* berupa
3. Penyesuaian Tampilan dan Pengaturan Urutan Kartu Output prediksi langsung memengaruhi perilaku aplikasi dengan SRS (*spaced repetition system*), misalnya**:**
   - kartu “ulangi” dimunculkan lebih sering,
   - kartu “hafal” ditunda kemunculannya,
   - UI dapat menampilkan indikator sesuai tingkat kesulitan.
4. Pengelolaan Output Prediksi secara Real-Time Prediksi diberikan segera setelah pengguna menjawab kartu. Tidak ada *delay* karena pembacaan model dilakukan sepenuhnya di *JavaScript*.
5. Pengulangan Proses secara Kontinu Setiap kali kartu baru muncul dan pengguna berinteraksi, parameter terbaru diambil dan keseluruhan proses prediksi dijalankan kembali. Hal ini membuat sistem adaptif terhadap perubahan performa pengguna.

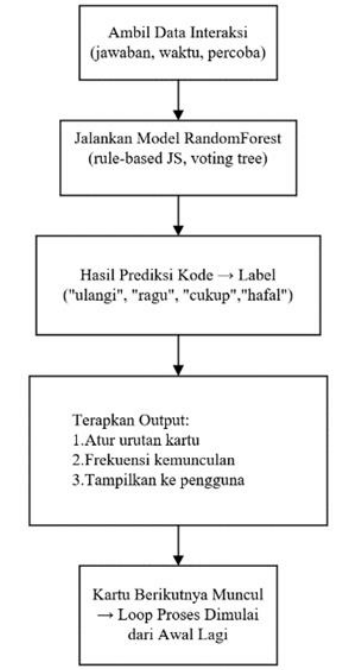
6. Pengulangan Proses secara Kontinu Setiap kali kartu baru muncul dan pengguna berinteraksi, parameter terbaru diambil dan keseluruhan proses prediksi dijalankan kembali. Hal ini membuat sistem adaptif terhadap perubahan performa pengguna.



Gambar 2. Flowchart Output Prediksi Tingkat Kesulitan (Versi Proyek Web *Flashcard*)

4. HASIL DAN PEMBAHASAN

Kinerja Model *Random Forest*



Berdasarkan hasil evaluasi model *Random Forest* yang telah dilatih menggunakan dataset interaksi pengguna *flashcard*, diperoleh performa klasifikasi untuk prediksi tingkat kesulitan kartu sebagai berikut:

1. Akurasi Keseluruhan

   Model *Random Forest* mencapai akurasi sebesar 88.64% pada data pengujian, dengan *error rate* sebesar 11.36%. Hasil ini menunjukkan bahwa model mampu memprediksi tingkat kesulitan kartu dengan cukup baik, meskipun menggunakan dataset yang relatif kecil (218 sampel).

2. Performa per Kelas

   Distribusi performa model untuk setiap kelas tingkat kesulitan disajikan pada Tabel 1 berikut:

Tabel 1. Performa Klasifikasi Model *Random Forest* per Kelas

Kelas	***Precision***	***Recall***	***F1-Score***	***Support***
hafal	0.857	0.960	0.906	25
ulangi	1	1	1	6
cukup	0.875	0.636	0.737	11
ragu	1	1	1	2

3. Rata-rata Makro (*Macro-Average*)

   Rata-rata makro dari metrik evaluasi menunjukkan:

   Precision: 0.933

   Recall: 0.899

   F1-Score: 0.910

Nilai *precision* yang tinggi (0.933) menunjukkan bahwa model jarang memberikan prediksi positif yang salah, sementara *recall* yang lebih rendah (0.899) mengindikasikan bahwa model terkadang melewatkan kasus positif yang sebenarnya.

Analisis *Confusion Matrix*

Confusion matrix memberikan gambaran rinci tentang pola kesalahan klasifikasi model, seperti yang ditunjukkan pada Tabel 2:

Tabel 2. *Confusion Matrix* Model *Random Forest*

Predicted / Actual	hafal	ulangi	ragu	cukup
hafal	24	0	0	1
ulangi	0	6	0	0
cukup	4	0	0	7
ragu	0	0	2	0

Dari *confusion matrix* dapat dianalisis bahwa:

- Kelas "ulangi" terklasifikasi dengan sempurna (6 benar dari 6 sampel)
- Kelas "hafal" menunjukkan performa yang cukup baik dengan 24 sampel terdeteksi benar, 1 sampel salah diklasifikasikan menjadi kelas “cukup”, dan ada 4 kasus dari kelas "cukup" yang salah diklasifikasi sebagai "hafal"
- Kelas "cukup" mengalami kesulitan dalam klasifikasi, di mana 7 dari 11 sampel yang terprediksi benar, sementara 4 sampel salah diklasifikasi sebagai "hafal"
- Kelas "ragu" walaupun dengan sedikitnya sampel (2 sampel) menunjukkan performa yang bagus karena dapat terklasifikasi dengan sempurna.

Analisis *Feature Importance*

Meskipun implementasi *Random Forest* dalam *JavaScript* tidak secara langsung menyediakan nilai *feature importance*, analisis terhadap struktur pohon keputusan yang dihasilkan menunjukkan bahwa fitur-fitur berikut memiliki pengaruh signifikan dalam prediksi:

- selisih_waktu_jam: Fitur ini muncul sebagai split utama pada pohon pertama dengan *threshold* 27.27 jam
- lama_baca_kartu: Memiliki peran kritis dalam membedakan berbagai tingkat kesulitan

- hasil_tebakan: Berpengaruh signifikan dalam mengidentifikasi kartu yang perlu diulang ("ulangi")
- jumlah_percobaan: Berkontribusi dalam menentukan level penguasaan kartu
- waktu_lama_tebakan: Memiliki pengaruh meskipun tidak sekuat fitur lainnya

Pembahasan Hasil

1. Kelebihan Model

   Model *Random Forest* yang dikembangkan menunjukkan beberapa kelebihan:

   - Stabilitas Klasifikasi: Akurasi 88.64% pada dataset kecil menunjukkan kemampuan generalisasi yang baik
   - Performa Cukup Optimal untuk Kelas Mayoritas: Model sangat akurat dalam mengklasifikasi kelas "hafal" (*recall* 96%) dan "ulangi" (*precision* dan *recall* 100%)
   - Efisiensi Komputasi: Konversi ke fungsi *JavaScript* memungkinkan prediksi real-time tanpa beban *server*
   - Ketahanan terhadap *Overfitting*: Struktur ensemble dengan lima pohon mengurangi risiko *overfitting* meskipun dataset kecil

2. Tantangan dan Keterbatasan

   1) Ketidakseimbangan Kelas (*Class Imbalance*):

      a. Distribusi dataset yang tidak merata (123 "hafal", 48 "cukup", 36 "ulangi", 11 "ragu") mempengaruhi performa model

      b. Kelas minoritas "ragu" hanya memiliki 11 sampel, sehingga sulit bagi model untuk mempelajari pola yang representatif

   2) Kesulitan Membedakan "cukup" dan "hafal":

      Berdasarkan hasil uji coba, model menghadapi tantangan signifikan dalam membedakan kategori "cukup" dan "hafal", di mana sebanyak 4 dari 11 kartu pada kategori "cukup" secara keliru diprediksi sebagai "hafal". Hal ini mengindikasikan adanya kemiripan karakteristik (*feature overlap*) yang sangat kuat antara kedua kelas tersebut dalam dataset yang digunakan. Kondisi tersebut terjadi karena parameter seperti waktu menjawab dan lama baca kartu pada kategori "cukup" memiliki kemiripan pola dengan kartu yang

sudah "hafal", sehingga pohon keputusan cenderung melakukan voting pada kelas mayoritas.

3) Ukuran Dataset Terbatas:
   a. Dengan hanya 218 sampel, model mungkin tidak menangkap seluruh variasi pola belajar pengguna
   b. Validasi silang (*cross-validation*) terbatas karena jumlah data yang kecil

3. Implikasi untuk Sistem Pembelajaran Adaptif

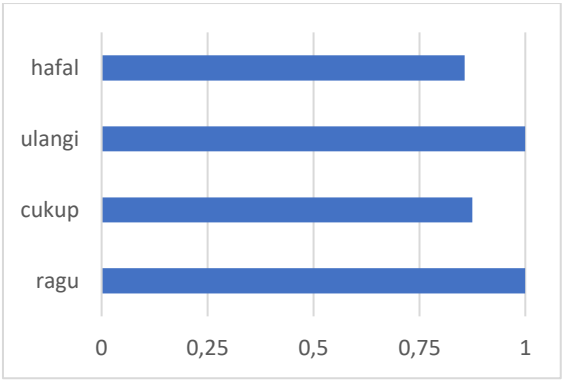
Hasil penelitian ini memiliki implikasi penting untuk pengembangan sistem pembelajaran adaptif:

1) *Personalized Learning* Path: Prediksi tingkat kesulitan yang akurat memungkinkan sistem untuk menyesuaikan alur belajar setiap pengguna
2) Optimasi *Spaced Repetition*: Kartu yang diprediksi "ulangi" dapat ditampilkan lebih sering, sementara kartu "hafal" dapat dikurangi frekuensinya
3) Efisiensi Waktu Belajar: Pengguna dapat fokus pada kartu yang benar-benar membutuhkan perhatian, mengoptimalkan waktu belajar

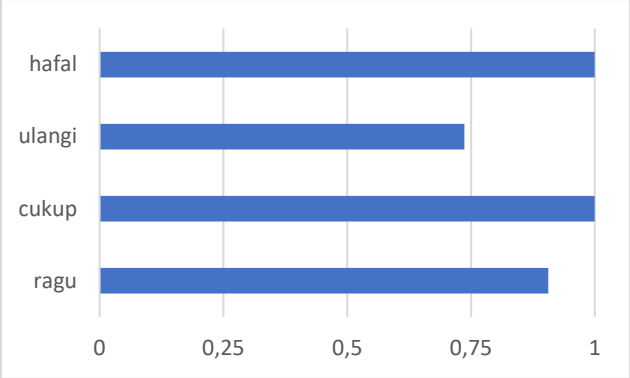
4. Analisis Visual Performa Model

Grafik berikut mengilustrasikan distribusi performa model:

Distribusi Akurasi per Kelas:



F1-Score per Kelas:



5. KESIMPULAN DAN SARAN

Model *Random Forest* yang dikembangkan dalam proyek ini berhasil memprediksi tingkat kesulitan kartu pada web *flashcard* berdasarkan parameter interaksi pengguna. Sistem mampu mengolah data yang dikumpulkan, melatih model melalui *Trainer3*, dan mengonversi hasilnya menjadi fungsi *JavaScript* berbasis *rule-based* (*if–else*) yang dapat dijalankan langsung di *browser*. Implementasi ini memungkinkan prediksi dilakukan secara cepat, ringan, dan tanpa ketergantungan *server*. Hasil prediksi membantu pengguna mengenali kartu mana yang tergolong mudah, sedang, atau sulit sehingga proses belajar menjadi lebih adaptif dan efektif. Pendekatan ini membuka peluang pengembangan lebih lanjut seperti penambahan fitur dataset, peningkatan akurasi model, dan pengembangan sistem pembelajaran adaptif berbasis perilaku pengguna.

Ketersediaan Data Dan Kode

Kode sumber, dataset, dan file implementasi dari penelitian ini tersedia di repositori *GitHub*:

https://github.com/projectganteng123/Random-Forest

Repositori ini berisi:

1. `*Trainer3.html*` - *Web-based* machine learning *trainer* untuk *Random Forest*
2. `*Final_FlashCard.html*` - Aplikasi web *flashcard* dengan model ML terintegrasi
3. `*flashcard_model.js*` - Model *Random Forest* dalam bentuk fungsi *JavaScript*
4. `*dataset.json*` - Dataset contoh untuk pelatihan model
5. `*evaluasi_model.json*` - Hasil evaluasi model lengkap

DAFTAR PUSTAKA

Badas, S. (2024). *Submitted Revised Published : June, 2*. 74–85.

Luh, N., Susantini, P., & Kristiantari, M. R. (2021). *Media Flashcard Berbasis Multimedia Interaktif untuk Pengenalan Kosakata Bahasa Inggris pada Anak Usia Dini*. *9*, 439–448.

Dini, F. S., Zainuddin, A., & Surakarta, U. M. (2025). *Pengembangan Flashcard Digital untuk Meningkatkan Penguasaan dan Pemahaman Kosakata Siswa Kelas II Sekolah Dasar ELSE (Elementary School Education*. *9*(2), 204–213.

Pendidikan, S., Informasi, T., Teknik, F., Surabaya, U. N., Pendidikan, S., Informasi, T., Teknik, F., Surabaya, U. N., & Industri, R. (n.d.). *TINJAUAN PUSTAKA SISTEMATIS TENTANG PENGGUNAAN FLASHCARD PADA MEDIA PEMBELAJARAN BERBASIS AUGMENTED REALITY Satria Bagus Wicaksana Yeni Anistyasari Abstrak*. 121–131.

Kusmiati, H., Rahmi, N., Saputra, B., Putri, K. P., & Sinaga, B. (2025). *Pelatihan Penggunaan M edia Pembelajaran Mandiri Siswa Berbasis Web Dengan Menerapkan Metode Podomoro dan Flashcard*. *4*(3), 553–559.

Media, P., Card, F., Pembelajaran, P., & Indonesia, B. (2024). *Flash Card Media Development in Grade 1 Indonesian Language Learning at Madrasah Ibtidaiyah Negeri 2 Jember*. *1*(2), 62–69.

Astuti, S., & Chandra, N. E. (2023). *Flashcard as Media in Teaching English : A Systematic Literature Review*. *13*(1), 395–409.

Justicia, R., Rahayu, A. K., Khaerunissa, F., & Herdiati, R. D. (2023). *Pelatihan Media Flashcard Voice Berbasis Teknologi Pada Guru PAUD*. *4*(2), 986–993.

Hanafi, M., Muhammadiyah, U., Rappang, S., & Haggar, E.-. (2023). *PENGUNAAN FLASHCARD BERBASIS DIGITAL*. *1*, 235–246.

Media, P., Card, F., Berbasis, L., Untuk, W. E. B., Kemampuan, M., Peserta, M., & Disleksia, D. (2025). *Pengembangan media flash card laca berbasis web untuk meningkatkan kemampuan membaca peserta didik disleksia*. *10*(September).