Pengembangan Sistem Klasifikasi Kekuatan Kata Sandi Berbasis Web Menggunakan Machine Learning



Disusun Oleh:

N a m a	: Adi Prasetyo
NIM	: 22523182

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2026

HALAMAN PENGESAHAN DOSEN PEMBIMBING

Pengembangan Sistem Klasifikasi Kekuatan Kata Sandi Berbasis Web Menggunakan Machine Learning

TUGAS AKHIR

Disusun Oleh:

N a m a : Adi Prasetyo

NIM : 22523182

Yogyakarta, 5 Juli 2026

Pembimbing 1,	Pembimbing 2,
(Fayruz Rahma, S.T., M.Eng.)	(Erika Ramadhani, S.T., M.Eng.)

HALAMAN PENGESAHAN DOSEN PENGUJI

Pengembangan Sistem Klasifikasi Kekuatan Kata Sandi Berbasis Web Menggunakan Machine Learning

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 5 Juli 2026

Tim Penguji

Fayruz Rahma, S.T., M.Eng. _____

Anggota 1

Hari Setiaji, S.Kom., M.Kom. _____

Anggota 2

Moh. Idris, S.Kom., M.Kom. _____

Mengetahui,
Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Adi Prasetyo

NIM : 22523182

Tugas akhir dengan judul:

Pengembangan Sistem Klasifikasi Kekuatan Kata Sandi Berbasis Web Menggunakan Machine Learning

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 5 Juli 2026



(Adi Prasetyo)

HALAMAN PERSEMBAHAN

Alhamdulillahi Robbil 'Alamin puji dan syukur atas limpahan nikmat Allah *Subhanahu wa Ta'ala* atas seluruh nikmat dan rahmat dari-Nya hingga penulis dapat menyelesaikan tugas akhir dengan baik. Tulisan ini saya persembahkan untuk kedua orang tua saya Bapak Kadar Atmo dan Ibu Rominah serta seluruh keluarga yang telah mendukung penuh dan mendoakan penulis. Terima kasih atas segala bentuk dukungan secara fisik maupun materi, doa yang tidak pernah berhenti dipanjatkan, serta kesabaran untuk selalu menasehati penulis.

HALAMAN MOTO

"To be ballin you gotta b-all-in"
(Penulis)

"If you know the enemy and know yourself,
you need not fear the result of a hundred battles"
(Sun Tzu)

"I need a room full of mirrors so I can be surrounded by winners."
(Kanye West)

"Live as if you were to die tomorrow. Learn as if you were to live forever"
(Mahatma Gandhi)

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh. Puji dan syukur penulis panjatkan ke hadirat Allah Subhanahu wa Ta'ala atas limpahan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan laporan tugas akhir yang berjudul "Pengembangan Sistem Klasifikasi Kekuatan Kata Sandi Berbasis Web Menggunakan Machine Learning". Sholawat dan salam semoga senantiasa tercurah kepada Nabi Muhammad saw., yang selalu menjadi teladan utama bagi saya dan seluruh umat manusia.

Laporan tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Informatika Program Sarjana, Fakultas Teknologi Industri, Universitas Islam Indonesia. Penelitian ini berfokus pada pembangunan sistem klasifikasi kekuatan kata sandi yang akurat, mengutamakan keamanan, sekaligus dapat ditafsirkan.

Dalam proses penyusunannya, penulis menyadari bahwa tugas akhir ini tidak akan selesai tanpa bantuan, bimbingan, dan dukungan dari berbagai pihak. Berbagai kesulitan yang penulis hadapi selama penelitian dapat dilalui berkat doa dan dukungan tersebut. Oleh karena itu, pada kesempatan ini penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua penulis, Bapak Kadar Atmo dan Ibu Rominah, atas doa yang tidak pernah putus, kasih sayang, serta selalu meyakinkan saya untuk menyelesaikan penelitian ini.
2. Seluruh dosen Program Studi Informatika – Program Sarjana Universitas Islam Indonesia, yang telah memberikan ilmu, bimbingan, dan pengalaman berharga selama masa perkuliahan.
3. Ibu Fayruz Rahma, S.T., M.Eng. dan Ibu Erika Ramadhani, S.T., M.Eng., selaku Dosen Pembimbing I dan II, atas bimbingan, arahan, koreksi, dan kesabaran beliau selama proses penyusunan tugas akhir ini.
4. Teman-teman seperjuangan yang telah menemani, membantu, dan memberikan semangat selama masa studi.
5. Semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah memberikan bantuan dalam penyelesaian tugas akhir ini.

Penulis menyadari bahwa laporan tugas akhir ini masih memiliki banyak kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan untuk perbaikan di masa mendatang. Akhir kata, penulis berharap semoga tugas akhir ini dapat memberikan manfaat bagi pembaca serta bagi perkembangan ilmu pengetahuan, khususnya di bidang keamanan informasi dan *machine learning*.

Yogyakarta, 5 Juli 2026



(Adi Prasetyo)

SARI

Autentikasi berbasis kata sandi masih menjadi metode autentikasi yang paling banyak digunakan sehingga keamanan banyak layanan digital pada akhirnya bergantung pada kekuatan kata sandi yang dipilih oleh penggunanya. Akan tetapi, pengguna cenderung memilih kata sandi yang lemah dan mudah ditebak. Sementara itu, sistem penilai kekuatan kata sandi yang ada masih memiliki keterbatasan, yaitu label pelatihan yang tidak konsisten, kesalahan klasifikasi yang tidak dibedakan berdasarkan risikonya, serta keputusan model yang tidak dapat dijelaskan. Penelitian ini bertujuan membangun sistem klasifikasi kekuatan kata sandi berbasis web yang akurat, mengutamakan keamanan, sekaligus dapat ditafsirkan.

Penelitian disusun mengikuti kerangka *Cross-Industry Standard Process for Data Mining* (CRISP-DM). *Dataset Password Strength Classifier* dari Kaggle yang berjumlah 669.639 entri diberi label ulang menggunakan *library zxcvbn* untuk mengganti label asli yang terbukti hanya berbasis panjang karakter. Kemudian, setiap kata sandi direpresentasikan ke dalam 25 fitur yang diekstraksi langsung dari komposisi karakter dan pencocokan terhadap basis data kata sandi yang bocor. Empat model *machine learning*, yaitu *Logistic Regression*, *Random Forest*, *Support Vector Machine*, dan *XGBoost*, dibandingkan dalam dua skenario, yaitu dengan dan tanpa penanganan ketidakseimbangan kelas. Batas keamanan asimetris diterapkan melalui optimasi *threshold,* yaitu dengan menekan proporsi kesalahan dalam menilai kata sandi lemah sebagai kuat (*weak→strong*) di bawah 5%, sedangkan keputusan model dijelaskan menggunakan *library* SHAP. Model terbaik kemudian diimplementasikan ke dalam *prototype* sistem berbasis web.

Random Forest tanpa penanganan ketidakseimbangan kelas terpilih sebagai model terbaik dengan *F1-Score* makro 0,8121, akurasi 0,8344, dan *recall* kelas *weak* 0,7601 pada data uji sehingga melampaui seluruh kriteria keberhasilan. Penanganan ketidakseimbangan kelas justru menurunkan performa pada seluruh algoritma. Melalui optimasi *threshold*, proporsi kesalahan *weak→strong* berhasil ditekan menjadi 4,73% disertai penurunan *F1-Score* makro yang kecil dan terkendali, yaitu sebesar 0,0104. Analisis SHAP menunjukkan bahwa panjang kata sandi dan keterpaparan pada basis data kata sandi yang bocor merupakan faktor yang paling berpengaruh terhadap penilaian.

Kata kunci: klasifikasi kata sandi, *machine learning*, SHAP, *threshold* asimetris, zxcvbn.

GLOSARIUM

Confusion Matrix	Tabel yang merangkum jumlah prediksi benar dan salah untuk setiap kelas, menjadi dasar penghitungan *precision, recall*, dan analisis jenis kesalahan.
CRISP-DM	*Cross-Industry Standard Process for Data Mining* adalah kerangka proses pengembangan *data science* terdiri dari 6 fase: pemahaman bisnis, pemahaman data, persiapan data, pemodelan, evaluasi, dan penerapan.
Cross-validation	Teknik evaluasi yang membagi data menjadi beberapa lipatan untuk menguji konsistensi performa model.
Data Leakage	Kondisi saat informasi uji memengaruhi pelatihan sehingga estimasi performa model menjadi terlalu optimis.
Dictionary Attack	Metode serangan yang memanfaatkan daftar kredensial umum yang pernah bocor untuk menebak kata sandi.
Entropi Shannon	Ukuran tingkat ketidakpastian atau keacakan kata sandi berdasarkan frekuensi kemunculan setiap karakter, nilai yang lebih tinggi menandakan kata sandi yang lebih sulit ditebak.
F1-Score Makro	Rata-rata *F1-Score* seluruh kelas dengan bobot setara, tanpa memperhitungkan jumlah sampel pada tiap kelas; lebih informatif pada data tidak seimbang.
Feature Engineering	Proses mengubah data mentah, seperti *string* kata sandi, menjadi himpunan nilai numerik yang informatif bagi model.
Feature Importance	Analisis yang menunjukkan seberapa besar kontribusi suatu fitur terhadap prediksi model secara keseluruhan.
Ground Truth	Label acuan yang dianggap benar dan menjadi dasar pelatihan serta evaluasi model.
Leet Substitution	Pola penggantian huruf dengan angka atau simbol yang bentuk visualnya menyerupai huruf aslinya, seperti angka 0 untuk huruf o, angka 3 untuk huruf e, atau simbol @ untuk huruf a.
LIME	*Local Interpretable Model-Agnostic Explanations* adalah metode interpretasi yang menjelaskan prediksi model pada satu instans dengan membangun model sederhana yang meniru perilaku model di sekitar instans tersebut; dapat diterapkan pada model apa pun.
LUDS	*Lower, Upper, Digits, Symbols* adalah penilaian kekuatan kata sandi berdasarkan keberadaan empat tipe karakter: huruf kecil, huruf besar, angka, dan simbol, tanpa mempertimbangkan apakah kata sandi tersebut mudah ditebak atau tidak.
MCC	*Matthews Correlation Coefficient* adalah ukuran kualitas klasifikasi

	yang memperhitungkan keempat sel *confusion matrix* sekaligus dan tetap informatif pada data tidak seimbang; nilainya berkisar dari -1 hingga 1.
PARS	*Password Analysis and Research System* adalah perangkat dari Georgia Institute of Technology yang menggabungkan beberapa *password meter* komersial untuk menilai kekuatan kata sandi; menjadi sumber label asli pada dataset yang digunakan.
SHAP	*SHapley Additive exPlanations* adalah metode interpretasi berbasis teori permainan yang memberikan nilai kontribusi setiap fitur terhadap prediksi model, baik secara global maupun pada prediksi tunggal.
Stratified splitting	Pembagian data yang mempertahankan proporsi kelas pada setiap subset, yaitu data latih, data validasi, dan data uji.
Supervised Learning	Pembelajaran terbimbing, yaitu pelatihan model dari data yang telah diberi label.
TF-IDF	*Term Frequency-Inverse Document Frequency* adalah ukuran yang menilai pentingnya suatu unsur berdasarkan frekuensi kemunculannya, dibobot dengan kelangkaannya pada keseluruhan data.
Threshold	Nilai batas keputusan pada keluaran probabilitas model yang menentukan kelas prediksi.
Undersampling	Penyeimbangan kelas dilakukan dengan mengurangi jumlah sampel pada kelas mayoritas.
XAI	*Explainable Artificial Intelligence* adalah bidang yang mengembangkan metode agar keputusan model *machine learning* dapat dipahami manusia, baik secara global maupun pada tiap prediksi.
Zxcvbn	*Library open-source* penilai kekuatan kata sandi berbasis pencocokan pola yang menghasilkan skor kekuatan dari 0 hingga 4. Dalam penelitian ini, skor tersebut digunakan sebagai acuan pelabelan ulang *dataset* menggantikan label asli yang hanya berbasis panjang karakter.

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI ix
GLOSARIUM x
DAFTAR ISI xii
DAFTAR TABEL xv
DAFTAR GAMBAR xvi
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan Penelitian 5
1.5 Manfaat Penelitian 5
1.6 Struktur Laporan 6
BAB II TINJAUAN PUSTAKA 8
2.1 Konsep Kekuatan Kata Sandi dan Model Ancaman 8
2.2 Pendekatan Penilaian Kekuatan Kata Sandi dan Perbandingannya 10
2.3 Pelabelan dan Penentuan *Ground Truth* Kekuatan Kata Sandi 13
2.4 Algoritma Machine Learning untuk Klasifikasi Kekuatan Kata Sandi 15
2.5 Rekayasa Fitur Kata Sandi 18
2.6 Explainability (XAI) dan Interpretasi Model 21
2.7 Metodologi Pengembangan Model dan CRISP-DM 22
2.8 Evaluasi dan Pengujian Model Klasifikasi 24
2.9 Posisi dan Kontribusi Penelitian 27
BAB III METODOLOGI PENELITIAN 30
3.1 *Business Understanding* 31
3.1.1 Identifikasi Masalah 31

3.1.2 Tujuan Teknis 32
3.1.3 Kriteria Keberhasilan 32
3.2 *Data Understanding* 32
3.2.1 Sumber Data 33
3.2.2 Eksplorasi Awal Dataset 33
3.2.3 Distribusi Label 33
3.2.4 Analisis Hubungan Label dan Panjang Kata Sandi 34
3.2.5 Keterbatasan Label Asli 35
3.3 *Data Preparation* 35
3.3.1 Penanganan *Missing Values* 35
3.3.2 *Relabeling* Dataset 36
3.3.3 Ekstraksi Fitur 36
3.3.4 Distribusi Label Setelah *Relabeling* 39
3.3.5 Pembagian Dataset 40
3.4 *Modeling* 41
3.4.1 Algoritma *Machine Learning* yang Digunakan 41
3.4.2 Skenario Eksperimen 42
3.4.3 Optimasi *Hyperparameter* 43
3.4.4 Optimasi *Threshold* 44
3.5 *Evaluation* 44
3.5.1 Metrik Evaluasi 44
3.5.2 Prosedur Pemilihan Model 45
3.5.3 Evaluasi Final 46
3.6 *Deployment* 46
3.6.1 Penyimpanan Model 47
3.6.2 Arsitektur Aplikasi Web 47
3.6.3 Implementasi Antarmuka Pengguna 50
3.6.4 Interpretabilitas SHAP 51
BAB IV HASIL DAN PEMBAHASAN 52
4.1 Hasil Perbandingan Model 52
4.2 Pemilihan Model Terbaik 53
4.3 Hasil Optimasi *Threshold* 55
4.3.1 Hasil Optimasi *Threshold* pada Model Final 56
4.3.2 Analisis Sensitivitas Batas Keamanan 56

4.4 Evaluasi Final pada Data Uji 57
4.4.1 Performa Keseluruhan dan Per Kelas 58
4.4.2 *Confusion Matrix* 58
4.4.3 Verifikasi Pemenuhan Target 60
4.5 Interpretasi Model 61
4.5.1 *Feature Importance* Global 61
4.5.2 Interpretasi SHAP per Kelas 62
4.6 Hasil Implementasi *Prototype* 66
4.7 Pembahasan 68
4.7.1 Strategi *Relabeling* dan Kualitas Data 68
4.7.2 Performa Model dan Pengaruh *Balancing* 69
4.7.3 Pengaruh Threshold Asimetris terhadap Performa dan Keamanan 70
4.7.4 Interpretasi Model dan Fitur Berpengaruh 72
4.7.5 Analisis Kesalahan Model dan Batas Kemampuannya 73
4.7.6 Keterbatasan Penelitian 74
BAB V KESIMPULAN DAN SARAN 76
5.1 Kesimpulan 76
5.2 Saran 77
DAFTAR PUSTAKA 79
LAMPIRAN 85

DAFTAR TABEL

Tabel 2.1 Perbandingan Pendekatan Penilaian Kekuatan Kata Sandi 12
Tabel 2.2 Perbandingan Kerangka Proses Pengembangan *Data Science* 23
Tabel 2.3 Ringkasan dan Posisi Penelitian Terdahulu 27
Tabel 3.1 Statistik panjang kata sandi per kelas 34
Tabel 3.2 Skoring zxcvbn 36
Tabel 3.3 Daftar seluruh fitur yang diekstraksi 37
Tabel 3.4 Pembagian dataset 41
Tabel 3.5 Grid *hyperparameter* per algoritma pada *RandomizedSearchCV* 43
Tabel 3.6 *Endpoint Backend* 49
Tabel 4.1 Perbandingan performa delapan model kandidat pada data validasi 52
Tabel 4.2 Konfigurasi *hyperparameter* model final *Random Forest* 54
Tabel 4.3 Hasil optimasi *threshold* model final pada data validasi 56
Tabel 4.4 *F1-Score* makro terbaik yang dapat dicapai pada berbagai batas keamanan 57
Tabel 4.5 Performa model final per kelas pada data uji 58
Tabel 4.6 Verifikasi pemenuhan target pada data uji 60
Tabel 4.7 Sepuluh fitur paling berpengaruh pada model final 61

DAFTAR GAMBAR

Gambar 3.1 Diagram alur CRISP-DM 31
Gambar 3.2 Grafik distribusi label asli 33
Gambar 3.3 Histogram distribusi panjang kata sandi per kelas 34
Gambar 3.4 Distribusi kelas setelah *relabeling zxcvbn* 40
Gambar 3.5 Diagram Arsitektur Sistem 48
Gambar 3.6 Tampilan antarmuka pengguna 50
Gambar 4.1 *Confusion matrix* sebelum optimasi *threshold* model final pada data uji 59
Gambar 4.2 *Confusion matrix* setelah optimasi *threshold* model final pada data uji 59
Gambar 4.3 Ringkasan SHAP global kata sandi *weak* 63
Gambar 4.4 Ringkasan SHAP global kata sandi *medium* 64
Gambar 4.5 Ringkasan SHAP global kata sandi *strong* 65
Gambar 4.6 Demonstrasi klasifikasi dan penjelasan untuk kata sandi *weak* pada antarmuka *prototype* 66
Gambar 4.7 Demonstrasi klasifikasi dan penjelasan untuk kata sandi *medium* pada antarmuka *prototype* 67
Gambar 4.8 Demonstrasi klasifikasi dan penjelasan untuk kata sandi *strong* pada antarmuka *prototype* 67
Gambar 4.9 Trade-off spesifikasi keamanan terhadap performa pada data validasi 71

BAB I
PENDAHULUAN

1.1 Latar Belakang

Autentikasi merupakan fondasi keamanan setiap sistem digital. Di antara beragam metode yang ada, autentikasi berbasis kata sandi masih menjadi yang paling banyak digunakan (Rehman et al., 2024). Meskipun alternatif yang dianggap lebih aman seperti biometrik, autentikasi multifaktor, dan *passkey* telah berkembang, kata sandi tetap bertahan karena kesederhanaan, biaya yang rendah, dan keakrabannya bagi pengguna (Rehman et al., 2024; Thai et al., 2024) sehingga keamanan sebagian besar layanan digital pada akhirnya bergantung pada kekuatan kata sandi yang dipilih penggunanya. Ketergantungan ini menjadi persoalan serius karena kredensial tetap menjadi salah satu sasaran yang paling sering diserang dalam serangan siber. Laporan Verizon Data Breach Investigations Report (DBIR) tahun 2026 mencatat bahwa jika dilihat hanya dari cara penyerang pertama kali masuk ke sistem, penggunaan kredensial curian memang menurun menjadi 13%, kalah oleh pemanfaatan celah keamanan perangkat lunak yang kini menjadi jalan masuk paling umum (31%). Akan tetapi, jika seluruh tahapan serangan diperhitungkan, penyalahgunaan kredensial tetap paling sering muncul, yaitu pada 39% pelanggaran, dan pencurian kredensial juga tetap menjadi tindakan utama dalam serangan terhadap aplikasi web (Verizon, 2026). Kredensial tersebut diperoleh penyerang melalui *phishing*, *infostealer*, maupun data dari pelanggaran sebelumnya yang dikemas dan diperjualbelikan di *dark web* sehingga menjadi bagian penting dari perangkat penyerang (Verizon, 2026).

Akar persoalan tersebut terletak pada perilaku pengguna yang cenderung memilih kata sandi yang lemah dan mudah ditebak. Studi tahunan NordPass edisi ketujuh mendapati bahwa "123456" kembali menjadi kata sandi yang paling umum digunakan di dunia, menempati posisi teratas selama tujuh tahun terakhir (NordPass, 2026). Bahkan, kualitas kata sandi buruk lintas generasi, dengan deret sederhana seperti "12345" dan "123456", muncul sebagai pilihan teratas di setiap kelompok usia (NordPass, 2026). Yang lebih mengkhawatirkan, setelah tujuh tahun pengamatan, NordPass mencatat bahwa masih sangat sedikit perbaikan dalam kebiasaan keamanan kata sandi pengguna, yang menunjukkan bahwa upaya yang ada belum mampu mendorong perubahan yang signifikan (NordPass, 2026). Temuan ini sejalan dengan analisis pada kumpulan data kata sandi nyata yang memperlihatkan bahwa mayoritas kata sandi

tergolong lemah, dengan panjang rata-rata yang pendek dan ragam karakter yang terbatas (Pande, 2026). Kecenderungan ini muncul karena ada *trade-off* mendasar antara kemudahan mengingat dan keamanan sehingga pengguna kerap memilih pola yang mudah ditebak (Thai & Tanaka, 2024a).

Untuk menjawab persoalan ini, standar keamanan telah bergeser. Pedoman NIST (*National Institute of Standards and Technology*) Special Publication 800-63B Revisi 4 yang diterbitkan pada Juli 2025 mensyaratkan panjang minimum kata sandi serta pemeriksaan terhadap *blocklist* yang berisi kata sandi umum, terduga, dan yang telah bocor, sekaligus melarang penerapan aturan komposisi karakter sebagai persyaratan wajib (Temoshok et al., 2025). NIST menjelaskan bahwa pengguna cenderung merespons aturan komposisi dengan modifikasi yang sederhana dan umum digunakan, misalnya dengan mengubah "*password"* menjadi "*Password1"* atau "*Password1!"* sehingga keberadaan huruf besar, angka, dan simbol tidak selalu mencerminkan kekuatan kata sandi yang sebenarnya (Temoshok et al., 2025). Sejalan dengan itu, Wheeler (2016) menunjukkan bahwa pendekatan berbasis LUDS (*lower, upper, digits, symbols*) hanya menghitung keberadaan huruf, angka, dan simbol sehingga gagal menangkap pola yang lazim digunakan pengguna, seperti kata-kata umum, tanggal, dan pola *keyboard*.

Penilaian kekuatan kata sandi telah berkembang dari *password strength meter* berbasis aturan sederhana menuju klasifikasi *machine learning* yang memanfaatkan pola dan karakteristik kata sandi secara otomatis (Khalid et al., 2024). Namun, pendekatan-pendekatan tersebut masih menyisakan beberapa keterbatasan. Pertama, banyak model dilatih menggunakan kumpulan data publik yang pelabelannya ditentukan sepenuhnya oleh alat ukur komersial secara otomatis, seperti yang dilakukan menggunakan PARS (*Password Analysis and Research System*) yang mengintegrasikan beberapa *password meter* komersial sekaligus, tanpa membandingkan hasilnya dengan standar keamanan yang lebih terukur (Divya et al., 2023). Kedua, sebagian besar penelitian hanya mengoptimalkan akurasi secara keseluruhan sehingga secara implisit memperlakukan semua jenis kesalahan secara setara, padahal kesalahan menilai kata sandi lemah sebagai kuat jauh lebih berbahaya karena dapat menjebak pengguna dalam rasa aman yang keliru (Thai et al., 2024). Ketiga, kemampuan model untuk menjelaskan dasar keputusannya merupakan aspek penting dalam membangun kepercayaan pengguna dan memberikan umpan balik yang bermakna; beberapa studi telah menerapkan kerangka *Explainable AI* untuk mengidentifikasi fitur atau karakter yang menentukan

klasifikasi kekuatan kata sandi (Shreya et al., 2024), tetapi penerapannya belum meluas di seluruh literatur.

Berdasarkan kesenjangan tersebut, penelitian ini membangun sistem klasifikasi kekuatan kata sandi berbasis *machine learning* yang berupaya menutup ketiga keterbatasan tersebut. Penelitian ini memberi label ulang pada data menggunakan *library zxcvbn* agar label yang dipakai lebih dapat dipercaya, mengandalkan fitur struktural atau ciri bentuk kata sandi agar penilaian tetap cepat ketika sistem digunakan. Selain itu, penelitian ini menerapkan batas keamanan asimetris, yaitu batas yang sengaja dibuat lebih ketat pada arah kesalahan yang paling berbahaya dibandingkan arah kesalahan lainnya, khususnya pada arah menilai kata sandi lemah sebagai kuat, serta memanfaatkan SHAP (*SHapley Additive exPlanations*) untuk menjelaskan dasar keputusan model. Dengan demikian, penelitian ini diharapkan menghasilkan klasifikasi kekuatan kata sandi yang tidak hanya akurat, tetapi juga aman dan dapat dijelaskan, serta diwujudkan ke dalam *prototype* sistem berbasis web yang dapat digunakan secara langsung. Untuk mengarahkan penelitian secara sistematis, rumusan masalah disusun sebagaimana diuraikan pada subbab berikut.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, persoalan inti yang dihadapi adalah bagaimana membangun penilaian kekuatan kata sandi yang akurat, aman, sekaligus dapat ditafsirkan, dengan label yang lebih andal dibandingkan label bawaan pada *dataset* publik. Dari persoalan tersebut, dirumuskan beberapa pertanyaan penelitian sebagai berikut:

1. Bagaimana membangun model klasifikasi kekuatan kata sandi yang memberikan performa terbaik sekaligus memenuhi batas keamanan terhadap kesalahan menilai kata sandi lemah sebagai kuat?
2. Bagaimana keputusan klasifikasi model dapat dijelaskan menggunakan SHAP agar hasilnya dapat ditafsirkan?
3. Bagaimana mengimplementasikan model terpilih beserta mekanisme penjelasannya ke dalam *prototype* sistem klasifikasi kekuatan kata sandi berbasis web?

1.3 Batasan Masalah

Agar penelitian terarah dan terukur, ditetapkan batasan masalah sebagai berikut.

1. Data

Penelitian menggunakan satu *dataset* utama untuk pelatihan dan klasifikasi, yaitu *Password Strength Classifier Dataset* bersumber dari Kaggle (Bhavik Bansal, 2019) sebanyak 669.640 entri. Selain itu, daftar kata RockYou (sekitar 13,8 juta kata unik) (William J. Burns, 2019) digunakan bukan sebagai *dataset* klasifikasi, melainkan sebagai kamus referensi untuk menghitung sebagian fitur pencocokan kamus. Penelitian ini tidak menggunakan *dataset* klasifikasi lain dan tidak melakukan pengumpulan data primer.

2. Kategori Kekuatan

   Klasifikasi dibatasi pada tiga kategori, yaitu *weak, medium,* dan *strong*, serta tidak menggunakan tingkatan yang lebih rinci.

3. Acuan Pelabelan

   Kekuatan kata sandi didefinisikan secara operasional melalui penilaian *zxcvbn* sebagai acuan (*ground truth*). Penelitian ini tidak menggunakan penilaian pakar manusia maupun metrik lain sebagai acuan utama, dan label bawaan pada *dataset* tidak digunakan.

4. Fitur

   Penelitian ini hanya menggunakan fitur struktural dan interaksi yang dihitung dari string kata sandi. Pembelajaran representasi otomatis melalui *deep learning* tidak digunakan.

5. Algoritma

   Hanya ada empat algoritma yang dibandingkan, yaitu *Logistic Regression*, *Random Forest, Support Vector Machine*, dan XGBoost. Pendekatan *deep learning* dan *large language model* berada di luar lingkup ini karena keduanya bekerja langsung dari representasi karakter mentah tanpa memerlukan ekstraksi fitur sehingga perbandingannya dengan model berbasis fitur struktural tidak akan setara secara metodologis.

6. Fokus Penelitian

   Penelitian ini berfokus pada penilaian dan klasifikasi kekuatan kata sandi, bukan pada pembangkitan kata sandi, rekomendasi kata sandi, maupun simulasi peretasan (*cracking*) secara langsung.

7. Representativitas Pengguna

   Penelitian ini menggunakan *dataset* dan kamus referensi yang berasal dari kumpulan kata sandi publik berskala internasional. Oleh karena itu, penelitian ini tidak

mengevaluasi secara khusus karakteristik maupun pola pembentukan kata sandi pengguna Indonesia.

8. Implementasi Sistem

   Sistem yang dibangun menggunakan FastAPI dan React merupakan *prototype* fungsional untuk mendemonstrasikan model beserta penjelasannya. Aspek keamanan infrastruktur, skalabilitas produksi, dan pengujian beban berada di luar lingkup ini.

Selain itu, penelitian ini berlandaskan pada beberapa asumsi. Pertama, *library zxcvbn* dianggap sebagai acuan pelabelan yang cukup andal untuk merepresentasikan kekuatan kata sandi karena telah digunakan secara nyata di industri dan dipakai sebagai acuan pelabelan dalam penelitian terdahulu. Kedua, *dataset* yang digunakan dianggap cukup representatif terhadap pola kata sandi yang lemah dan umum di dunia nyata. Ketiga, fitur struktural dianggap memadai untuk mendekati penilaian kekuatan yang dihasilkan oleh *zxcvbn*.

1.4 Tujuan Penelitian

Sejalan dengan rumusan masalah, penelitian ini bertujuan untuk:

1. Membangun model klasifikasi kekuatan kata sandi ke dalam kategori *weak*, *medium*, dan *strong* berbasis fitur struktural dengan *zxcvbn* sebagai acuan pelabelan melalui perbandingan performa *Logistic Regression*, *Random Forest*, *Support Vector Machine*, dan XGBoost pada dua skenario penanganan ketidakseimbangan kelas untuk memperoleh model terbaik berdasarkan *F1-Score* makro, serta penerapan optimasi *threshold* asimetris agar proporsi kesalahan klasifikasi kata sandi lemah sebagai kuat tidak melebihi batas maksimal 5%.
2. Menjelaskan keputusan klasifikasi model menggunakan SHAP, baik secara global maupun lokal, agar hasilnya dapat ditafsirkan.
3. Mengimplementasikan model terbaik beserta mekanisme penjelasan SHAP ke dalam *prototype* sistem klasifikasi kekuatan kata sandi berbasis web.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat, baik secara akademik maupun praktis.

1. Manfaat Akademik

   Penelitian ini melengkapi penelitian yang sudah ada tentang klasifikasi kekuatan kata sandi dengan menggabungkan empat hal dalam satu sistem yang utuh, yaitu pemberian

ulang label menggunakan *zxcvbn*, fitur struktural atau ciri bentuk kata sandi, batas keamanan asimetris, dan penjelasan keputusan model menggunakan SHAP. Dari sisi metode, sumbangan utamanya adalah penerapan batas keamanan asimetris, yaitu batas yang sengaja dibuat lebih ketat pada jenis kesalahan yang paling berbahaya sehingga dapat menjadi acuan bagi penelitian lain yang juga menghadapi situasi ketika satu jenis kesalahan jauh lebih merugikan daripada yang lain. Selain itu, penelitian ini membandingkan secara langsung kinerja empat algoritma *machine learning* untuk menilai kekuatan kata sandi ketika jumlah data pada tiap kelas tidak seimbang.

2. Manfaat Praktis

   Penelitian ini menghasilkan sistem klasifikasi kekuatan kata sandi berbasis *machine learning* yang mampu memberikan penilaian beserta penjelasan terhadap hasil klasifikasi melalui interpretabilitas SHAP. Penerapan optimasi *threshold* asimetris memungkinkan sistem mengurangi risiko kesalahan klasifikasi kata sandi *weak* sebagai *strong*, sehingga mendukung proses evaluasi yang lebih berorientasi pada keamanan. *Prototype* yang dikembangkan dapat memberikan umpan balik secara langsung mengenai faktor-faktor yang memengaruhi hasil klasifikasi sehingga membantu pengguna memahami alasan suatu kata sandi diklasifikasikan sebagai *weak*, *medium*, atau *strong*. Selain itu, hasil penelitian ini dapat dijadikan sebagai dasar bagi pengembang untuk mengintegrasikan mekanisme evaluasi kekuatan kata sandi yang bersifat *interpretable* ke dalam sistem pendaftaran maupun autentikasi.

1.6 Struktur Laporan

Penulisan laporan penelitian ini disusun dalam lima bab dengan sistematika sebagai berikut.

a. BAB 1 PENDAHULUAN

   Bab ini menguraikan latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.

b. BAB 2 TINJAUAN PUSTAKA

   Bab ini membahas landasan teori dan kajian pustaka yang menjadi dasar penelitian, meliputi konsep kekuatan kata sandi dan model ancaman, pendekatan penilaian kekuatan kata sandi, pelabelan dan penentuan *ground truth*, algoritma *machine learning* untuk klasifikasi, rekayasa fitur, *explainability* dan SHAP, metodologi pengembangan

CRISP-DM, evaluasi dan pengujian model, serta posisi dan kontribusi penelitian ini terhadap penelitian terdahulu.

c. BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan tahapan penelitian yang mengikuti kerangka CRISP-DM, mulai dari pemahaman masalah, pengumpulan dan pemahaman data, pelabelan ulang menggunakan *zxcvbn* beserta analisis distribusi label hasilnya, rekayasa fitur, pemodelan dengan empat algoritma *machine learning*, penerapan batas keamanan asimetris, hingga rancangan evaluasi dan implementasi sistem.

d. BAB 4 HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil penelitian beserta pembahasannya, meliputi perbandingan performa keempat algoritma berdasarkan *F1-Score* makro, pengaruh penanganan ketidakseimbangan kelas, hasil penerapan batas keamanan asimetris, interpretasi keputusan model menggunakan SHAP, hasil implementasi *prototype* sistem, serta pembahasan menyeluruh atas seluruh temuan dan keterbatasan penelitian.

e. BAB 5 KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang menjawab rumusan masalah serta saran untuk pengembangan dan penelitian selanjutnya.

BAB II
TINJAUAN PUSTAKA

Bab ini menguraikan landasan teori dan kajian pustaka yang menjadi dasar penelitian klasifikasi kekuatan kata sandi. Pembahasan dimulai dari konsep dasar kekuatan kata sandi dan model ancaman yang melatarbelakanginya, kemudian beralih ke ragam pendekatan penilaian kekuatan kata sandi beserta perbandingan keunggulan dan keterbatasannya. Tinjauan selanjutnya diarahkan pada komponen utama yang menyusun sistem klasifikasi berbasis *machine learning*, yaitu pelabelan dan penentuan *ground truth*, algoritma yang umum digunakan, serta rekayasa fitur kata sandi. Setelah itu dibahas aspek interpretabilitas model melalui *Explainable AI* dan SHAP, metodologi pengembangan CRISP-DM sebagai kerangka kerja penelitian, serta prinsip evaluasi dan pengujian model klasifikasi. Pada bagian akhir, seluruh tinjauan disintesis untuk menetapkan posisi dan kontribusi penelitian ini terhadap penelitian terdahulu.

2.1 Konsep Kekuatan Kata Sandi dan Model Ancaman

Kekuatan kata sandi pada dasarnya adalah ukuran sejauh mana sebuah kata sandi mampu bertahan terhadap upaya penebakan oleh penyerang, bukan sekadar tingkat kepatuhannya terhadap aturan komposisi karakter. Yu et al. (2024b) mengelompokkan penelitian penilaian kekuatan kata sandi ke dalam empat kategori utama: berbasis aturan (*rule-based*), deteksi pola (*pattern detection*), klasifikasi (*classification*), dan analisis serangan penebakan (*guessability attacks*). Pengelompokan ini menyiratkan bahwa "kekuatan" bukan properti tunggal yang terdefinisi secara mutlak, melainkan konsep yang operasionalisasinya bergantung pada cara seseorang memodelkan kemampuan penyerang.

Operasionalisasi yang paling klasik adalah pendekatan berbasis aturan. Penilaian NIST, misalnya, mengukur kekuatan dari dua faktor: panjang kata sandi dan jenis karakter yang terkandung di dalamnya. Pendekatan ini mudah diimplementasikan dan berlatensi rendah, tetapi kemampuannya untuk menilai kekuatan secara akurat masih terbatas (Yu et al., 2024b). Wheeler (2016) menamai paradigma ini LUDS (*lower, upper, digits, symbols*), yaitu penghitungan huruf kecil, huruf besar, digit, dan simbol yang selama lebih dari tiga puluh tahun mendominasi syarat dan umpan balik kata sandi, meskipun menurutnya terbukti memberatkan sekaligus tidak efektif sebagai praktik keamanan. Ukuran berbasis entropi seperti entropi NIST

(*National Institute of Standards and Technology*), PQI (*Password Quality Indicator*), dan PCM (*Password Complexity Metric*) juga lazim digunakan. Namun, Thai & Tanaka (2024a) menegaskan bahwa metode-metode tersebut bertumpu pada aturan heuristik dan tidak secara spesifik memodelkan proses penebakan yang sesungguhnya. Implikasinya, komposisi karakter dan entropi teoretis tidak selalu mencerminkan ketahanan kata sandi terhadap serangan nyata.

Sebagai respons atas keterbatasan tersebut, berkembang pandangan yang mendefinisikan kekuatan secara langsung sebagai jumlah tebakan (*guesses*) yang diperlukan penyerang untuk memecahkan kata sandi. *Zxcvbn*, estimator yang dikembangkan Wheeler dan digunakan oleh Dropbox, mengadopsi heuristik ini: untuk setiap pola yang dikenali, sistem mengajukan pertanyaan "jika penyerang mengetahui pola ini, berapa jumlah tebakan yang ia perlukan?", lalu mengestimasi ruang tebakan berdasarkan pola alih-alih mengasumsikan string acak (Wheeler, 2016). Thai & Tanaka (2024a) menggambarkan *zxcvbn* sebagai PSM (*Password Strength Meter*) *non-probabilistik* yang terinspirasi oleh cara kerja *password cracker*, yang mengenali puluhan ribu kata sandi umum melalui pencocokan pola. Dalam kerangka ini, kekuatan diukur relatif terhadap perilaku penyerang rasional yang terlebih dahulu mencoba pola yang lebih sederhana atau lebih mungkin.

Karena kekuatan didefinisikan relatif terhadap penyerang, nilainya bergantung pada model ancaman yang diasumsikan. Wheeler (2016) membedakan serangan daring (*online*) yang dibatasi laju percobaan dengan ambang praktis sekitar 10^6 tebakan dari serangan luring (*offline*) yang jauh lebih leluasa, serta mencatat bahwa teknik *key stretching* seperti Argon2 dapat membuat serangan *offline* tidak layak pada magnitudo tebakan yang tinggi. Artinya, sebuah kata sandi dapat dinilai memadai untuk menahan serangan *online,* tetapi tetap rapuh terhadap serangan *offline*. Lebih jauh, Thai & Tanaka (2024a) menunjukkan bahwa kata sandi yang dinilai kuat oleh sejumlah PSM tetap rentan apabila telah bocor atau dikompromikan. Dengan demikian, kekuatan kata sandi tidak hanya bergantung pada karakteristik string itu sendiri, tetapi juga pada pengetahuan dan sumber daya yang dimiliki penyerang.

Kompleksitas konsep ini diperberat oleh perilaku pengguna. Terdapat *trade-off* mendasar antara kemudahan mengingat dan keamanan: kata sandi sederhana seperti kombinasi nama dan tahun lahir (mis. *david1980*, *shane1983*) mudah diingat tetapi sangat mudah diprediksi, sedangkan kata sandi acak yang kompleks lebih aman tetapi kurang praktis (Thai & Tanaka, 2024a, 2024b). Ketegangan ini mendorong pengguna memilih strategi yang dapat diprediksi, termasuk penggunaan ulang dan pola umum sehingga kata sandi yang tampak memenuhi syarat komposisi pun bisa lemah secara nyata (Pande, 2026). Pola-pola semacam ini terdokumentasi

dalam data kata sandi dunia nyata (Yu et al., 2024b), dan bahkan kontribusi elemen tertentu, seperti karakter spesial, terhadap keamanan ternyata tidak sesederhana yang diasumsikan serta perlu dianalisis secara empiris (He et al., 2024, 2025). Konsekuensinya, penilaian kekuatan kata sandi yang andal menuntut pendekatan yang mampu menangkap keteraturan statistik dari kata sandi nyata, bukan sekadar menghitung komposisi karakter yang selama ini menjadi landasan bagi beragam pendekatan penilaian yang dibahas pada Subbab 2.2.

2.2 Pendekatan Penilaian Kekuatan Kata Sandi dan Perbandingannya

Alat utama untuk menilai kekuatan kata sandi adalah *password strength meter* (PSM), yang dikembangkan untuk membantu pengguna mengestimasi kekuatan kata sandi mereka (Thai & Tanaka, 2024b). Mengacu pada kategorisasi Yu et al. (2024b), pendekatan penilaian kekuatan kata sandi berkembang dari metode dari yang paling sederhana berbasis aturan, hingga metode yang belajar secara langsung dari data kata sandi. Bagian ini meninjau pendekatan-pendekatan tersebut beserta perbandingan keunggulan dan keterbatasannya, kemudian menjelaskan posisi pendekatan berbasis pembelajaran mesin yang digunakan dalam penelitian ini.

Pendekatan paling awal yang digunakan untuk menilai kekuatan kata sandi adalah pendekatan berbasis aturan dan entropi. Model LUDS untuk penghitungan huruf kecil, huruf besar, digit, dan simbol telah mendominasi syarat kata sandi selama puluhan tahun (Wheeler, 2016), sementara ukuran berbasis entropi seperti entropi NIST, PQI, dan PCM mengandalkan aturan heuristik tanpa secara spesifik memodelkan proses penebakan (Thai & Tanaka, 2024a). Kim & Lee (2024) menggolongkan LUDS dan entropi NIST sebagai metode berbasis algoritma. Upaya pada lini ini masih berlanjut, misalnya Lingamsetty et al. (2024) yang mengusulkan metrik kekuatan berbasis entropi dengan mekanisme penalti yang dirancang ringan untuk perangkat berdaya terbatas. Keunggulan pendekatan ini adalah kemudahan implementasinya dan berlatensi rendah (Yu et al., 2024b). Kelemahan utamanya, sebagaimana ditegaskan Wheeler (2016), adalah ketidakmampuannya menangkap pola yang justru paling umum pada kata sandi nyata, seperti kata, tanggal, dan pola *keyboard*, karena pola-pola tersebut tidak terwakili oleh sekadar hitungan kelas karakter.

Pendekatan berbasis pola dan kamus berusaha menutup celah tersebut. *Zxcvbn* mengestimasi kekuatan dengan mencocokkan kata sandi terhadap kamus, tanggal, pola *keyboard*, dan substitusi karakter, lalu menerjemahkannya menjadi perkiraan jumlah tebakan (Wheeler, 2016). Thai & Tanaka (2024a) mencatat bahwa *zxcvbn* bersifat *non-probabilistik*,

terinspirasi oleh cara kerja *password cracker*, dan mengenali lebih dari 40.000 kata sandi umum melalui pencocokan pola. Pendekatan ini lebih realistis daripada LUDS karena memodelkan pengetahuan penyerang secara eksplisit, tetapi bergantung pada kamus terkurasi serta estimasi heuristik yang konservatif.

Pendekatan probabilistik menawarkan akurasi dan reliabilitas yang tinggi dalam evaluasi kekuatan kata sandi karena memodelkan distribusi statistik kata sandi secara eksplisit, tetapi menuntut korpus pelatihan dan beban komputasi yang relatif lebih besar (Thai & Tanaka, 2024b). Model seperti *Simple Markov Model* dan *Layered Markov Model* mempertimbangkan posisi karakter dan panjang kata sandi, dan Thai & Tanaka (2024b) mengusulkan model hibrida untuk menutup kasus yang belum tercakup, sementara varian statistiknya memanfaatkan daftar kata sandi yang bocor sebagai basis penilaian. Sejalan dengan itu, *probabilistic context-free grammar* (PCFG) banyak digunakan untuk memodelkan struktur kata sandi. He et al. (2024, 2025) memanfaatkan PCFGv4 untuk mengukur dampak karakter spesial terhadap ketahanan kata sandi, dan Shinde et al. (2025) mencatat bahwa PCFG maupun model *deep learning* seperti *PassGAN* mampu mempelajari distribusi kata sandi pengguna untuk menghasilkan tebakan yang masuk akal. Pande (2026) memadukan entropi dengan analisis rantai *Markov* dalam kerangka klasifikasi.

Pendekatan berbasis *machine learning* memformulasikan penilaian kekuatan sebagai tugas klasifikasi. Model dilatih untuk menggolongkan kata sandi ke dalam kategori diskret seperti *weak*, *medium*, dan *strong* berdasarkan fitur-fiturnya (Khalid et al., 2024; Rehman et al., 2024). Keunggulan utamanya adalah kemampuan untuk belajar dari *dataset* kata sandi nyata berskala besar guna mengidentifikasi karakteristik kata sandi yang lemah (Asaduzzaman et al., 2024). Berbagai studi membandingkan algoritma klasik pada tugas ini. Rehman et al. (2024) melaporkan *Artificial Neural Network* sebagai yang terbaik di antara model yang mereka uji, sedangkan Mo et al. (2026) membandingkan enam algoritma, termasuk model *stacking*, dengan metrik *accuracy*, *recall*, dan *F1* melalui validasi dan penyetelan hiperparameter. Varian yang lebih canggih memanfaatkan *deep learning*, Alkurdi et al. (2023) menggunakan *Long Short-Term Memory* (LSTM), *Generative Adversarial Network* (GAN), dan model bahasa dua arah untuk menilai sekaligus memprediksi kerentanan kata sandi, sementara Yu et al. (2024a) menggabungkan *Bidirectional Long Short-Term Memory* (BiLSTM) untuk pemodelan sekuensial dan operasi *Convolutional Neural Network* (CNN) untuk pemodelan spasial dalam klasifikasi atribut kata sandi ultra-pendek.

Pendekatan mutakhir mulai memanfaatkan *large language model* (LLM). Kim & Lee (2024) mengusulkan evaluasi kekuatan menggunakan GPT-4 dengan metrik berbasis *prompt* yang mencakup kompleksitas, kemudahan diingat, dan inklusi informasi pribadi tanpa pelatihan tambahan, lalu membandingkannya dengan LUDS dan *zxcvbn* pada 2.000 kata sandi yang diambil sebagai sampel. Selain digunakan secara langsung sebagai evaluator, LLM juga mulai dimanfaatkan untuk mendukung pengembangan sistem penilaian. Pawade et al. (2024), misalnya, menggunakan GPT-3.5, Bard, dan Bing untuk menghasilkan *dataset* sintetis yang kemudian digunakan dalam eksperimen klasifikasi kekuatan kata sandi.

Perbandingan keenam pendekatan tersebut, beserta prinsip, kelebihan dan keterbatasannya, dirangkum pada Tabel 2.1. Dari perbandingan tersebut tampak pola *trade-off* yang konsisten. Pendekatan berbasis aturan dan entropi unggul dalam kesederhanaan dan kecepatan, tetapi terlalu kasar untuk menangkap pola kata sandi yang nyata; pendekatan berbasis pola seperti *zxcvbn* lebih realistis, tetapi bergantung pada kamus terkurasi dan heuristik; pendekatan probabilistik akurat dalam memodelkan distribusi, tetapi menuntut korpus dan komputasi yang lebih besar; sedangkan pendekatan LLM fleksibel, tetapi belum mapan. Penelitian ini memilih pendekatan klasifikasi *machine learning* karena beberapa pertimbangan: *machine learning* belajar dari distribusi kata sandi dunia nyata alih-alih aturan tetap, menghasilkan kategori diskret yang sesuai untuk umpan balik kepada pengguna, efisien pada tahap inferensi, dan mampu mengintegrasikan beragam fitur struktural maupun berbasis pola. Perlu ditekankan bahwa pendekatan-pendekatan tersebut tidak saling eksklusif. Keluaran dari estimator berbasis pola dapat digunakan sebagai acuan pelabelan, sementara hasil pencocokan kamus dapat dimanfaatkan sebagai fitur masukan. Oleh karena itu, metode pelabelan kekuatan kata sandi akan sangat menentukan performa model klasifikasi yang dihasilkan, sebagaimana dibahas pada Subbab 2.3.

Tabel 2.1 Perbandingan Pendekatan Penilaian Kekuatan Kata Sandi

Pendekatan	**Mekanisme**	**Kelebihan**	**Keterbatasan**	**Contoh Studi**
Berbasis aturan & entropi	Menghitung komposisi karakter (LUDS); entropi heuristik (NIST, PQI, PCM)	Sederhana; latensi rendah (Yu et al., 2024b).	Tidak menangkap kata, tanggal, dan pola *keyboard* (Wheeler, 2016); entropi tak memodelkan	(Kim & Lee, 2024; Lingamsetty et al., 2024; Wheeler, 2016)

Pendekatan	Mekanisme	Kelebihan	Keterbatasan	Contoh Studi
			penebakan (Thai & Tanaka, 2024a)	
Berbasis pola/kamus	Pencocokan kamus dan pola untuk mengestimasi jumlah tebakan (Wheeler, 2016)	Menangkap pola nyata; *cracker-inspired* (Thai & Tanaka, 2024a)	Bergantung kamus dan pola baku	(Thai & Tanaka, 2024a; Wheeler, 2016)
Probabilistik	Pemodelan distribusi statistik (*Markov*, PCFG)	Akurasi dan reliabilitas tinggi (Thai & Tanaka, 2024a)	Perlu korpus untuk pelatihan	(He et al., 2024, 2025; Pande, 2026; Thai & Tanaka, 2024b, 2024a)
Machine Learning (klasik)	Klasifikasi *supervised* dari fitur rancangan (*weak*/*medium*/*strong*)	Belajar dari data nyata (Asaduzzaman et al., 2024); keluaran kategori diskret	Bergantung pada fitur dan label; rentan *imbalanced* (Alajaji & Althaqeb, 2025)	(Khalid et al., 2024; Mo et al., 2026; Rehman et al., 2024; Sarkar & Nandan, 2022)
Deep Learning	Pembelajaran representasi otomatis (LSTM, GAN, BiLSTM + CNN)	Tanpa ekstraksi fitur	Representasi implisit, kurang dapat ditafsirkan	(Alkurdi et al., 2023; Yu et al., 2024a)
Berbasis LLM	LLM untuk evaluasi via *prompt,* atau generate *dataset* berlabel	Tanpa pelatihan tambahan (Kim & Lee, 2024)	Mutakhir; jumlah studi terbatas	(Kim & Lee, 2024; Pawade et al., 2024)

2.3 Pelabelan dan Penentuan *Ground Truth* Kekuatan Kata Sandi

Klasifikasi *supervised* menuntut data yang telah berlabel, dan kualitas label kekuatan (*ground truth*) secara langsung menentukan mutu model yang dihasilkan. Yu et al. (2024b) menjelaskan bahwa metode berbasis klasifikasi bekerja dengan belajar dari sekumpulan kata sandi yang telah dilabeli dengan baik untuk mengenali fitur pembeda antara kata sandi yang kuat dan kata sandi yang lemah. Persoalannya, daftar kata sandi yang bocor yang kerap menjadi sumber data umumnya hanya berupa daftar mentah tanpa label kekuatan apa pun (Darbutaitė

et al., 2023). Akibatnya, label harus diperoleh melalui suatu prosedur dan pilihan prosedur pelabelan itu sendiri menjadi penentu validitas seluruh proses pembelajaran.

Salah satu sumber data yang paling banyak digunakan dalam tugas ini adalah *Password Strength Classifier Dataset* yang tersedia di Kaggle (Bhavik Bansal, 2019) dan telah diadopsi dalam berbagai studi (Alajaji & Althaqeb, 2025; Divya et al., 2023; Khalid et al., 2024; Lingamsetty et al., 2024; Rehman et al., 2024). *Dataset* ini memuat tiga kelas, yaitu *weak*, *medium*, dan *strong* (Lingamsetty et al., 2024). Penelusuran asal-usul labelnya menjadi penting. Divya et al. (2023) menjelaskan bahwa kata sandi pada keluarga data ini, yang bersumber dari kebocoran 000webhost, dikategorikan menjadi lemah, sedang, dan kuat menggunakan PARS, yaitu perangkat dari Georgia Tech yang mengintegrasikan berbagai *password meter* komersial. Dengan demikian, label tersebut merupakan keluaran agregat dari sejumlah meter, bukan penilaian yang ditetapkan secara seragam. Kondisi ini menimbulkan persoalan konsistensi, sebab sebagaimana dikemukakan dalam tinjauan Divya et al. (2023), evaluasi terhadap berbagai *password meter* menunjukkan adanya kelemahan dan ketidakkonsistenan dalam label kekuatan yang diberikan, yang berpotensi membingungkan pengguna.

Untuk memperoleh label yang lebih andal, literatur menggunakan beberapa strategi. Strategi pertama memanfaatkan estimator yang kuat sebagai labeler. Salah satu estimator kekuatan kata sandi yang paling banyak digunakan adalah *zxcvbn*, yaitu *library open-source* yang dikembangkan oleh Daniel Wheeler dari Dropbox sebagai alternatif terhadap *password strength meter* berbasis aturan (LUDS). Berbeda dengan pendekatan yang hanya menghitung jumlah huruf besar, huruf kecil, angka, dan simbol, *zxcvbn* mengestimasi kekuatan kata sandi berdasarkan jumlah tebakan (*guess attempts*) yang diperkirakan diperlukan penyerang untuk berhasil menebaknya. Untuk memperoleh estimasi tersebut, *zxcvbn* melakukan pencocokan terhadap berbagai pola yang umum digunakan pengguna, seperti kata yang terdapat dalam kamus, nama umum, tanggal, urutan karakter, pola *keyboard*, pengulangan karakter, serta substitusi karakter (*leet substitution*). Selanjutnya, setiap pola diberi estimasi jumlah tebakan dan digabungkan untuk menghasilkan estimasi keseluruhan kekuatan kata sandi (Wheeler, 2016).

Pande (2026) menggunakan *library zxcvbn* sebagai sumber *ground truth* dengan alasan eksplisit untuk menghindari kebocoran label (*label leakage*), yaitu kondisi ketika label klasifikasi diturunkan dari fitur yang sama yang digunakan untuk prediksi. Strategi kedua adalah menyusun *dataset* khusus ketika data berlabel tidak tersedia, seperti Darbutaitė et al.

(2023) yang mengompilasi *dataset* kata sandi untuk konteks bahasa Lithuania dari sumber yang semula tidak memiliki label, kemudian menghasilkan label kekuatan kata sandi menggunakan versi modifikasi *password strength-meter zxcvbn* yang disesuaikan dengan karakteristik bahasa Lithuania. Setelah proses pelabelan, fitur-fitur kata sandi dan metrik kemiripan berbasis kamus diekstraksi untuk membangun model pembelajaran mesin.

Pemilihan pemberi label menjadi krusial karena konsekuensi kesalahan label bersifat asimetris. Thai et al. (2024) menegaskan bahwa keliru melabeli kata sandi kuat sebagai lemah tidak terlalu berdampak, tetapi melabeli kata sandi lemah sebagai kuat menimbulkan risiko yang signifikan. Oleh karena itu, *recall* yang tinggi untuk kata sandi lemah menjadi sangat penting agar kata sandi lemah tidak salah dikenali sebagai kata sandi yang kuat. Implikasinya, penelitian ini tidak memerlukan bukti bahwa *zxcvbn* merupakan estimator terbaik, melainkan membutuhkan estimator yang digunakan secara konsisten sebagai sumber pelabelan dan terpisah dari fitur-fitur prediktif yang digunakan oleh model. Dalam konteks tersebut, penggunaan *zxcvbn* sebagai pemberi label memenuhi kebutuhan metodologis untuk mengurangi risiko *label leakage* sekaligus menyediakan prosedur pelabelan yang dapat direplikasi.

Berdasarkan tinjauan tersebut, penggunaan *zxcvbn* sebagai pemberi label pakar (*expert labeler*) dapat dibenarkan karena telah digunakan dalam penelitian terdahulu untuk menghasilkan label kekuatan kata sandi yang konsisten, terutama ketika *dataset* sumber tidak memiliki label atau ketika reliabilitas label bawaan *dataset* dipertanyakan. Dengan pendekatan ini, tujuan pemodelan tidak lagi sekadar mereproduksi label asli *dataset*, melainkan mempelajari representasi fitur yang mampu mendekati penilaian kekuatan yang diberikan oleh estimator tersebut. Pendekatan ini mengubah rumusan tugas pemodelan dari sekadar memprediksi label asli *dataset* menjadi mempelajari cara mereproduksi penilaian kekuatan yang lebih andal berdasarkan fitur-fitur kata sandi. Strategi pelabelan ulang yang diterapkan dalam penelitian ini, yang dirinci pada Bab 3, mengikuti landasan tersebut. Setelah label yang andal ditetapkan, pertanyaan berikutnya adalah algoritma pembelajaran mana yang paling efektif untuk mereproduksinya, yang menjadi pokok bahasan pada Subbab 2.4.

2.4 Algoritma Machine Learning untuk Klasifikasi Kekuatan Kata Sandi

Sebagaimana dijelaskan pada Subbab 2.2, pendekatan berbasis *machine learning* memperlakukan penilaian kekuatan kata sandi sebagai tugas klasifikasi, yaitu memetakan sebuah kata sandi ke salah satu kategori diskret berdasarkan fitur-fiturnya. Pada tugas ini,

sejumlah algoritma *machine learning* kerap dibandingkan untuk menentukan model yang paling sesuai. Bagian ini meninjau algoritma-algoritma yang umum digunakan, kecenderungan performanya dalam literatur, peran metode *ensemble*, serta penanganan ketidakseimbangan kelas yang menyertainya.

Empat algoritma *machine learning* yang banyak digunakan dalam tugas klasifikasi adalah *Logistic Regression*, *Random Forest*, *Support Vector Machine* (SVM), dan XGBoost. Keempat algoritma tersebut mewakili pendekatan pembelajaran yang berbeda, yaitu model linier probabilistik, model berbasis *maximum margin*, *ensemble learning* berbasis *bagging*, dan *ensemble learning* berbasis *boosting*. Perbedaan mekanisme pembelajaran tersebut menyebabkan setiap algoritma memiliki karakteristik, kelebihan, dan keterbatasan yang berbeda dalam membangun model klasifikasi (Alshammari et al., 2026; Chen & Guestrin, 2016). Oleh karena itu, sebelum membahas kecenderungan performanya pada berbagai penelitian, mekanisme dasar masing-masing algoritma dijelaskan terlebih dahulu.

Logistic Regression merupakan algoritma klasifikasi yang memodelkan probabilitas suatu data untuk termasuk ke dalam kelas tertentu berdasarkan kombinasi linier dari fitur-fitur masukan. Berbeda dengan regresi linier yang menghasilkan nilai kontinu, *Logistic Regression* menggunakan fungsi logistik untuk mengubah hasil kombinasi linier tersebut menjadi nilai probabilitas antara 0 dan 1. Pada permasalahan klasifikasi multikelas, model *logistic regression* dapat diperluas menjadi *multinomial logistic regression* dengan menggunakan satu logit untuk setiap kelas dan fungsi *softmax* (Alshammari et al., 2026). Probabilitas keanggotaan suatu data pada kelas ke-k dihitung menggunakan Persamaan 2.1.

$$\mathrm{p(y = k|x)} = \frac{e^{w_k \cdot x + b_k}}{\sum_j e^{w_j \cdot x + b_j}} \qquad (2.1)$$

Random Forest merupakan algoritma *ensemble* yang membangun sejumlah *decision tree* menggunakan teknik *bagging* (*Bootstrap Aggregating*) (Alshammari et al., 2026). Setiap pohon dilatih menggunakan *bootstrap sample*, yaitu sampel data yang dipilih secara acak dengan pengembalian (*sampling with replacement*), sedangkan pada setiap proses pemisahan simpul hanya sebagian fitur yang dipilih secara acak untuk dipertimbangkan (Alshammari et al., 2026). Prediksi akhir ditentukan berdasarkan mekanisme *majority voting* dari seluruh pohon yang dibangun (Alshammari et al., 2026).

Support Vector Machine (SVM) merupakan algoritma klasifikasi yang bekerja dengan mencari *hyperplane* yang memaksimalkan margin sebagai batas pemisah antar kelas (Alshammari et al., 2026). Margin didefinisikan sebagai jarak *ortogonal* terpendek antara dua *class-bounding hyperplanes* yang menyentuh *support vectors* (Alshammari et al., 2026).

XGBoost (*Extreme Gradient Boosting*) merupakan algoritma *ensemble* berbasis *gradient boosting* yang membangun model secara bertahap (*additive training*) (Chen & Guestrin, 2016). Model dibangun secara aditif dengan menambahkan satu pohon pada setiap iterasi, di mana fungsi objektif terdiri atas *loss function* dan komponen regularisasi. (Chen & Guestrin, 2016). Selain itu, XGBoost menerapkan berbagai optimasi, seperti *column subsampling*, *sparsity-aware algorithm*, *weighted quantile sketch*, serta optimasi cache untuk meningkatkan efisiensi pada data berskala besar (Chen & Guestrin, 2016).

Kecenderungan performa dalam literatur menunjukkan dominasi metode berbasis pohon dan *boosting*, meskipun hasilnya bervariasi antarstudi. Divya et al. (2023) melaporkan *Random Forest* sebagai model terbaik dengan akurasi 98%, sedangkan *Logistic Regression* sebagai yang terendah dengan akurasi 82%. Kayalvili et al. (2024) melaporkan bahwa *Random Forest* menunjukkan performa terbaik dibandingkan dengan *Decision Tree* dan KNN yang diuji. Di sisi *boosting*, Sarkar & Nandan (2022) menemukan bahwa XGBoost mengungguli pengklasifikasi lain dengan akurasi 99%, dan Khalid et al. (2024) memilih XGBoost sebagai model terbaik dengan akurasi 0,9933 serta *Matthews Correlation Coefficient* (MCC) sebesar 0,99. Sementara itu, Mo et al. (2026) yang mengevaluasi enam algoritma melaporkan *Decision Tree* sebagai model unggul dengan akurasi 0,9998, *recall* 1,0, dan F1 0,9976, sedangkan Rehman et al. (2024) justru menempatkan *Artificial Neural Network* sebagai yang terbaik di antara model yang mereka uji. Variasi pemenang ini menunjukkan bahwa tidak ada satu algoritma pun yang unggul secara universal, tetapi metode linier seperti *Logistic Regression* cenderung tertinggal. Dalam tinjauan yang dirangkum oleh Alajaji & Althaqeb (2025), *Random Forest* mencapai akurasi 91%, sedangkan *Logistic Regression* mencapai 73% dan *Stochastic Gradient Descent* mencapai 67%. Perlu dicatat bahwa akurasi yang sangat tinggi, kerap di atas 99%, lazim ditemukan pada tugas ini, yang menyiratkan perlunya evaluasi yang lebih cermat daripada sekadar akurasi, sebagaimana dibahas pada Subbab 2.8.

Selain model tunggal, pendekatan *ensemble* diposisikan sebagai strategi untuk meningkatkan keandalan. Mo et al. (2026) memasukkan model *stacking* di antara enam algoritma yang diujinya, sedangkan Alajaji & Althaqeb (2025) mengembangkan ROBENS, yaitu kerangka berbasis *ensemble* yang memanfaatkan *Random Forest* dan SVM untuk

memperoleh penilaian yang andal meskipun ada keterbatasan data. Pande (2026) menempuh jalur serupa dengan memadukan entropi, analisis rantai Markov, dan pengklasifikasi XGBoost dalam satu kerangka. Penggabungan beberapa model maupun beberapa sumber informasi ini mencerminkan upaya untuk menutup kelemahan model individual dan meningkatkan *robustness* terhadap variasi data.

Persoalan yang menyertai klasifikasi kekuatan kata sandi adalah ketidakseimbangan kelas, sebab distribusi kata sandi nyata cenderung timpang dengan proporsi kategori tertentu yang jauh lebih kecil. Alajaji & Althaqeb (2025) menyoroti bahwa *dataset* yang terbatas, tidak seimbang, dan mengandung *noise* dapat memengaruhi performa model. Dalam salah satu studi yang mereka tinjau, *Random Forest* mencapai akurasi 91%, tetapi masih ditemukan keterbatasan pada nilai *recall*, *F1-Score*, dan MCC. Studi tersebut juga menyimpulkan bahwa ketidakseimbangan kelas dan kualitas data tetap menjadi tantangan yang signifikan. Untuk mengatasinya, literatur menawarkan beberapa cara. Pande (2026) menerapkan teknik *oversampling* SMOTE pada data RockYou, sedangkan Alajaji & Althaqeb (2025) menjadikan penyeimbangan data sebagai komponen dalam kerangka mereka. Namun, penyeimbangan juga memiliki konsekuensi, sebab Alajaji & Althaqeb (2025) mengingatkan bahwa pengurangan sampel (*undersampling*) berisiko menghilangkan informasi yang berguna. Pertimbangan ini menjelaskan alasan eksperimen dengan dan tanpa penyeimbangan kelas yang relevan dilakukan, sekaligus alasan metrik seperti *recall* dan F1 lebih informatif daripada akurasi pada konteks yang timpang.

Berdasarkan tinjauan tersebut, pemilihan *Random Forest* dan XGBoost sebagai kandidat berbasis pohon yang berperforma kuat, berdampingan dengan *Logistic Regression* dan SVM sebagai pembanding linier, serta penyertaan penanganan ketidakseimbangan kelas, sejalan dengan temuan-temuan di atas. Namun, performa algoritma sangat bergantung pada fitur yang diberikan kepadanya. Oleh karena itu, rekayasa fitur kata sandi menjadi penentu yang tidak kalah penting dan menjadi pokok bahasan pada Subbab 2.5.

2.5 Rekayasa Fitur Kata Sandi

Performa algoritma klasifikasi sangat ditentukan oleh kualitas fitur yang disediakan kepadanya. Rekayasa fitur adalah proses mengubah sebuah *string* kata sandi mentah menjadi himpunan nilai numerik yang informatif bagi model. Pada penilaian kekuatan kata sandi, fitur-fitur yang digunakan dalam literatur dapat dikelompokkan ke dalam beberapa keluarga, mulai dari fitur struktural sederhana, ukuran entropi, fitur pola, hingga fitur yang membandingkan

kata sandi dengan sumber daya eksternal. Bagian ini meninjau keluarga-keluarga fitur tersebut beserta rasional perilaku yang melatarbelakanginya.

Keluarga fitur yang paling mendasar adalah fitur struktural yang dihitung langsung dari komposisi karakter. Mo et al. (2026) mengekstraksi sejumlah karakteristik kata sandi, meliputi panjang, jumlah digit, jumlah huruf besar dan huruf kecil, serta jumlah karakter spesial. Fitur serupa juga digunakan oleh Khalid et al. (2024) yang menghitung, antara lain, jumlah huruf kecil dan jumlah karakter spesial pada setiap kata sandi, sedangkan Darbutaitė et al. (2023) menyertakan fitur kata sandi sederhana, seperti panjang kata sandi, sebagai salah satu fitur. Selain hitungan komposisi, ukuran entropi digunakan untuk merepresentasikan kompleksitas atau tingkat keacakan kata sandi. Pande (2026) menempatkan entropi Shannon sebagai salah satu metode utama. Fitur-fitur ini menangkap intuisi bahwa kata sandi yang lebih panjang dan lebih beragam jenis karakternya cenderung lebih sulit ditebak.

Keluarga berikutnya adalah fitur pola dan fitur berbasis referensi yang berupaya menangkap keteraturan yang tidak terwakili oleh sekadar hitungan karakter. *Zxcvbn* Wheeler (2016) mengenali pola-pola umum seperti tanggal, urutan (*sequences*), pengulangan (*repeats*), dan pola *keyboard*, lalu mengestimasi jumlah tebakan untuk setiap pola yang dikenali. Pola-pola inilah yang mendasari fitur penanda seperti deteksi urutan dan pengulangan. Di samping itu, fitur berbasis referensi membandingkan kata sandi dengan korpus eksternal. Salah satu korpus yang umum digunakan adalah RockYou, yaitu kumpulan kata sandi yang berasal dari kebocoran data perusahaan RockYou pada tahun 2009. Pada insiden tersebut, kata sandi pengguna disimpan dalam bentuk *plain text* sehingga seluruh daftar kata sandi berhasil diperoleh penyerang dan kemudian dipublikasikan. *Dataset* yang digunakan dalam penelitian ini merupakan distribusi *rockyou.txt* yang tersedia pada Kali Linux dan dipublikasikan melalui Kaggle sebagai kamus kata sandi (*password dictionary*) untuk berbagai kebutuhan penelitian keamanan informasi (William J. Burns, 2019).

Darbutaitė et al. (2023) menyertakan metrik kemiripan yang mengukur kedekatan kata sandi terhadap sejumlah kamus, dengan total 36 metrik kemiripan yang berasal dari empat fungsi kemiripan pada sembilan kamus berbeda. Pande (2026) memanfaatkan data RockYou beserta ukuran keragaman n-gram, sementara Asaduzzaman et al. (2024) mengusulkan model berbasis skor *Term Frequency-Inverse Document Frequency* (TF-IDF) yang menganalisis seberapa sering karakter tertentu muncul dalam kata sandi. Secara khusus, He et al. (2024) menyoroti segmen karakter spesial yang kerap terabaikan dan menyelidiki karakteristik serta pola semantiknya sebagai faktor yang memengaruhi ketahanan kata sandi.

Keluarga fitur yang lebih kompleks adalah fitur interaksi yang menggabungkan beberapa atribut sekaligus. Pande (2026) melakukan seleksi fitur berbasis suku interaksi antara panjang kata sandi dan jumlah kelas karakter, serta menemukan bahwa fitur nonlinier yang melibatkan kedua atribut tersebut merupakan faktor yang paling berpengaruh dalam klasifikasi kekuatan kata sandi. Temuan ini menunjukkan bahwa hubungan antaratribut dapat menjadi sumber informasi yang relevan dalam analisis kekuatan kata sandi. Contoh keteraturan semacam itu terlihat pada pola kata sandi yang menggabungkan nama dengan tahun kelahiran, seperti *david1980* atau *shane1983*, yang dilaporkan sebagai pola yang mudah diingat tetapi juga mudah diprediksi oleh penyerang (Thai & Tanaka, 2024a, 2024b). Selain itu, analisis terhadap data kata sandi nyata menunjukkan bahwa pengguna cenderung mengikuti pola tertentu saat membuat kata sandi (Yu et al., 2024b). Oleh karena itu, fitur yang mampu merepresentasikan keterkaitan antara beberapa komponen kata sandi dapat membantu mengidentifikasi pola-pola yang sering muncul pada kata sandi lemah meskipun secara permukaan tampak memenuhi aturan komposisi.

Sebagai alternatif terhadap fitur yang dirancang secara manual, sebagian pendekatan memanfaatkan pembelajaran representasi. Yu et al. (2024a) menggunakan BiLSTM untuk pemodelan sekuensial teks kata sandi dan operasi konvolusi untuk pemodelan spasial lokal, yang kemudian digabungkan menjadi representasi terpadu untuk klasifikasi atribut kata sandi ultra-pendek. Pendekatan ini mempelajari representasi langsung dari data kata sandi tanpa bergantung pada seperangkat fitur yang ditentukan sebelumnya. Dari tinjauan ini tampak adanya pembedaan penting antara fitur struktural yang intrinsik terhadap *string*, seperti panjang, komposisi, dan entropi, dengan fitur berbasis referensi yang memanfaatkan korpus eksternal, seperti kamus atau daftar kata sandi yang pernah bocor. Fitur berbasis referensi mampu menangkap pola kelemahan yang telah muncul dalam penggunaan nyata, tetapi efektivitasnya bergantung pada cakupan dan ketersediaan korpus referensi yang digunakan.

Pada akhirnya, himpunan fitur yang lebih kaya dan relevan berpotensi meningkatkan kemampuan model dalam membedakan kelas, tetapi sekaligus membuat keputusan model menjadi lebih sulit ditafsirkan. Memahami fitur mana yang paling memengaruhi suatu klasifikasi menjadi penting, baik untuk validasi maupun untuk memberikan umpan balik yang bermakna kepada pengguna. Kebutuhan akan interpretabilitas inilah yang mendorong pemanfaatan metode penjelasan model yang menjadi pokok bahasan Subbab 2.6.

2.6 Explainability (XAI) dan Interpretasi Model

Model klasifikasi yang berperforma tinggi pada penilaian kekuatan kata sandi umumnya berupa model kompleks, seperti ansambel pohon, yang sulit ditafsirkan secara langsung. Lundberg dan Lee (2017) menegaskan bahwa memahami alasan di balik suatu prediksi sama pentingnya dengan memperoleh prediksi yang akurat. Mereka juga menunjukkan bahwa model dengan akurasi tertinggi pada banyak *dataset* modern umumnya merupakan model yang sulit dipahami, bahkan oleh para pakar. Kondisi tersebut mencerminkan adanya *trade-off* antara akurasi dan interpretabilitas. Dalam konteks analisis kekuatan kata sandi, interpretabilitas tidak hanya membantu memahami faktor-faktor yang memengaruhi keputusan model, tetapi juga memberikan penjelasan mengenai alasan suatu kata sandi diklasifikasikan sebagai *weak*, *medium*, atau *strong*. Sebagai contoh, Shreya et al. (2024) memanfaatkan teknik *Explainable AI* berupa *Local Interpretable Model-Agnostic Explanations* (LIME) untuk menjelaskan hasil klasifikasi kata sandi.

Untuk memahami bagaimana model menghasilkan prediksi, berbagai pendekatan interpretabilitas telah dikembangkan. Salah satu pendekatan yang umum digunakan pada model berbasis pohon adalah analisis kepentingan fitur (*feature importance*) yang memberikan gambaran global mengenai kontribusi relatif setiap fitur terhadap prediksi model. Sebaliknya, pendekatan lokal berfokus pada penjelasan alasan di balik prediksi untuk suatu kata sandi tertentu. Sebagai contoh, Shreya et al. (2024) menerapkan teknik LIME untuk menjelaskan alasan model menggolongkan sebuah kata sandi ke kelas tertentu. Sebagai pendekatan yang lebih komprehensif, Lundberg dan Lee (2017) memperkenalkan SHAP (*SHapley Additive exPlanations*), yang memberikan nilai kontribusi setiap fitur terhadap suatu prediksi. Berbeda dengan pendekatan interpretabilitas yang dikembangkan secara terpisah, SHAP menyatukan berbagai metode sebelumnya ke dalam satu kerangka teoretis yang konsisten. Selain mampu menjelaskan prediksi pada tingkat lokal, SHAP juga dapat digunakan untuk menganalisis pengaruh fitur secara global sehingga memberikan pemahaman yang lebih menyeluruh mengenai perilaku model.

Karena model berbasis pohon banyak digunakan, tetapi relatif sedikit memperoleh perhatian dalam aspek interpretabilitas, Lundberg et al. (2020) mengembangkan *TreeExplainer*, yaitu formulasi SHAP yang dioptimalkan untuk model pohon. *TreeExplainer* memungkinkan perhitungan nilai SHAP secara eksak dengan kompleksitas yang jauh lebih rendah dibandingkan pendekatan SHAP umum. Selain itu, metode ini juga mendukung analisis

interaksi antarfitur pada tingkat lokal serta analisis kepentingan fitur pada tingkat global. Dengan kemampuan tersebut, *TreeExplainer* tidak hanya meningkatkan efisiensi komputasi, tetapi juga memberikan penjelasan yang lebih komprehensif terhadap perilaku model berbasis pohon. Formulasi ini relevan secara langsung dengan penelitian ini karena model yang digunakan merupakan *Random Forest*, sehingga nilai SHAP dapat dihitung secara langsung dan efisien menggunakan *TreeExplainer*.

Penerapan XAI dalam penilaian kekuatan kata sandi masih relatif terbatas dalam literatur yang ditinjau. Salah satu contoh yang ditemukan adalah penelitian Shreya et al. (2024) yang memanfaatkan LIME untuk menjelaskan hasil klasifikasi. Penelitian ini menempatkan diri pada arah yang sama, tetapi mengadopsi SHAP, khususnya formulasi TreeSHAP, untuk menjelaskan klasifikasi yang dihasilkan dari fitur-fitur struktural kata sandi. Penjelasan tersebut dapat dimanfaatkan pada tingkat global untuk mengidentifikasi fitur yang paling berpengaruh terhadap penilaian kekuatan kata sandi, serta pada tingkat lokal untuk memahami alasan sebuah kata sandi tertentu memperoleh label tertentu. Dengan demikian, interpretabilitas tidak hanya membantu memahami perilaku model, tetapi juga menyediakan dasar yang lebih kuat untuk menjelaskan hasil klasifikasi yang dihasilkan. Pembangunan model beserta mekanisme penjelasannya menuntut metodologi pengembangan yang terstruktur, yang menjadi pokok bahasan pada Subbab 2.7.

2.7 Metodologi Pengembangan Model dan CRISP-DM

Pengembangan model *machine learning* bukan sekadar pemilihan algoritma, melainkan rangkaian aktivitas yang perlu diorganisasikan secara sistematis dari pemahaman masalah hingga penerapan. Dalam ranah *data science*, beberapa model proses telah berkembang. Shimaoka et al. (2024) mengidentifikasi empat yang lazim digunakan, yaitu KDD (*Knowledge Discovery in Databases*), SEMMA (*Sample, Explore, Modify, Model, Assess*), CRISP-DM, dan TDSP (*Team Data Science Process*) dari Microsoft, yang perbandingannya dirangkum dalam Tabel 2.2. Di antara keempatnya, CRISP-DM (*Cross-Industry Standard Process for Data Mining*) menempati posisi sentral. Model yang dikembangkan pada era 1990-an oleh Teradata, SPSS, Daimler-Chrysler, dan OHRA ini menyerap prinsip dari KDD dan SEMMA dan, karena ketangguhannya, menjadi basis bagi proposal yang lebih baru seperti TDSP.

Wirth dan Hipp (2000) mendeskripsikan CRISP-DM sebagai model proses yang komprehensif dan independen terhadap sektor industri maupun teknologi. Model ini memecah proyek ke dalam enam fase, yaitu Pemahaman Bisnis, Pemahaman Data, Persiapan Data,

Pemodelan, Evaluasi, dan Penerapan, dengan urutan yang tidak kaku karena prosesnya bersifat iteratif sehingga peneliti dapat kembali ke fase sebelumnya bila diperlukan.

CRISP-DM dipilih dalam penelitian ini berdasarkan tiga pertimbangan berikut. Dari sisi kematangan, ia merupakan salah satu kerangka yang paling dikenal dan paling luas digunakan dalam proyek *data mining* dan *data science*; dari sisi karakteristik, ia tangguh, fleksibel, adaptif, dan dapat diterapkan pada berbagai domain (Shimaoka et al., 2024). Kemudian, dari sisi cakupan, keenam fasenya merentang dari pemahaman masalah hingga penerapan, selaras dengan alur penelitian ini yang dimulai dari pemahaman masalah, pelabelan ulang data, rekayasa fitur, pemodelan, evaluasi, hingga implementasi sistem. Sifat iteratifnya pun relevan karena perbaikan fitur kerap menuntut peninjauan ulang pada tahap persiapan data dan tahap pemodelan. Validitas model yang dihasilkan pada akhirnya bergantung pada cara evaluasinya, yang menjadi pokok bahasan pada Subbab 2.8.

Tabel 2.2 Perbandingan Kerangka Proses Pengembangan *Data Science*

Kerangka	**Sifat**	**Fase**	**Karakteristik Kunci**
KDD	Klasik dan umum, untuk penemuan pengetahuan	Selection → Preprocessing → Transformation → Data Mining → Interpretation/Evaluation (5 fase)	Tidak mendefinisikan fase Pemahaman Bisnis dan Penerapan secara eksplisit; evaluasi dan penerapan digabungkan dalam Interpretation/Evaluation. CRISP-DM merupakan evolusi praktisnya.
SEMMA	Proprietary, terikat pada satu vendor yaitu SAS (*Statistical Analysis System*)	Sample → Explore → Modify → Model → Assess (5 fase)	Berfokus pada tahap inti pemrosesan dan pemodelan data.
CRISP-DM	Terbuka, non-proprietary, lintas vendor	Business Understanding → Data Understanding → Data Preparation → Modeling → Evaluation → Deployment (6 fase)	Mencakup keseluruhan siklus dari pemahaman masalah hingga penerapan; bersifat iteratif, tangguh, fleksibel, dan adaptif; menjadi basis bagi TDSP.
TDSP	Berorientasi tim, lebih iteratif, selaras dengan konteks *agile*	Business Understanding → Data Acquisition and Understanding → Modeling → Deployment	Memiliki fase Customer Acceptance khusus dan peran yang terdefinisi memberi arah tetapi dapat membatasi fleksibilitas.

Kerangka	Sifat	Fase	Karakteristik Kunci
		→ Customer Acceptance (5 fase)	

2.8 Evaluasi dan Pengujian Model Klasifikasi

Setelah model klasifikasi dibangun, langkah berikutnya adalah mengevaluasi performanya menggunakan metrik yang sesuai. Evaluasi empiris merupakan pendekatan yang paling umum digunakan untuk membandingkan performa algoritma klasifikasi maupun menilai kesesuaiannya pada suatu permasalahan. Oleh karena itu, pemilihan metrik evaluasi yang tepat menjadi penting agar hasil pengujian dapat menggambarkan performa model secara objektif sesuai dengan karakteristik tugas klasifikasi yang dihadapi.

Dasar dari berbagai metrik evaluasi klasifikasi adalah *confusion matrix*, yaitu tabel yang merangkum hasil prediksi model terhadap label sebenarnya. Pada klasifikasi biner, *confusion matrix* terdiri atas empat komponen, yaitu *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN) (Sokolova & Lapalme, 2009). Keempat komponen tersebut menunjukkan jumlah prediksi yang benar maupun yang salah untuk setiap kelas. Pada klasifikasi multikelas, nilai TP, TN, FP, dan FN dihitung untuk masing-masing kelas, kemudian digunakan sebagai dasar dalam perhitungan berbagai metrik evaluasi (Sokolova & Lapalme, 2009).

Salah satu metrik evaluasi yang paling umum digunakan adalah *accuracy*. Metrik ini mengukur proporsi seluruh prediksi yang diklasifikasikan dengan benar dibandingkan dengan jumlah seluruh data yang dievaluasi. Nilai *accuracy* menunjukkan efektivitas keseluruhan *classifier* dalam melakukan klasifikasi (Sokolova & Lapalme, 2009). Pada klasifikasi biner, nilai *accuracy* dihitung berdasarkan jumlah *true positive* (TP) dan *true negative* (TN) terhadap seluruh data sebagaimana ditunjukkan pada Persamaan 2.2.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad (2.2)$$

Selain *accuracy*, evaluasi model klasifikasi juga banyak menggunakan metrik *precision* dan *recall*. *Precision* mengukur proporsi data yang diprediksi sebagai suatu kelas dan benar-benar termasuk dalam kelas tersebut, sedangkan *recall* mengukur proporsi data pada suatu kelas yang berhasil dikenali dengan benar (Sokolova & Lapalme, 2009). Nilai *precision* dan *recall* masing-masing dihitung menggunakan Persamaan 2.3 dan Persamaan 2.4.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.3)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.4)$$

Untuk memperoleh satu ukuran yang mempertimbangkan *precision* dan *recall* secara bersamaan, digunakan *F1-Score*. Metrik ini merupakan *harmonic mean* dari *precision* dan *recall*, sehingga nilai *F1-Score* dipengaruhi oleh kedua metrik tersebut (Powers, 2011). Nilai *F1-score* dihitung menggunakan Persamaan 2.5.

$$\text{F1} = 2 \text{ x } \frac{Precision \ x \ Recall}{Precision + Recall} \quad (2.5)$$

Pada klasifikasi multikelas, nilai *F1-Score* dihitung untuk setiap kelas secara terpisah, kemudian dirata-ratakan untuk memperoleh *F1-Score* makro. Pendekatan *macro averaging* memberikan bobot yang sama kepada setiap kelas sehingga setiap kelas memiliki kontribusi yang setara terhadap nilai evaluasi akhir (Sokolova & Lapalme, 2009). Nilai *F1-score* makro dihitung menggunakan Persamaan 2.6, dengan C menyatakan jumlah kelas dan F1 menyatakan nilai *F1-Score* pada kelas ke-i.

$$\text{F1macro} = \frac{1}{C}\sum_{i=1}^{C} F1_i \quad (2.6)$$

Pemilihan formulasi *averaged F1* pada penelitian ini didasarkan pada dua pertimbangan. Pertama, formulasi ini memberikan bobot yang sama kepada setiap kelas dengan menghitung nilai *F1-Score* secara terpisah sebelum dirata-ratakan, sehingga evaluasi tidak dipengaruhi oleh distribusi jumlah sampel antar kelas (Sokolova & Lapalme, 2009). Karakteristik ini sejalan dengan tujuan penelitian, yang menempatkan performa pada kelas *weak* sebagai perhatian utama dan tidak ingin performa kelas tersebut tenggelam oleh kelas mayoritas dalam proses evaluasi. Kedua, formulasi ini merupakan implementasi standar pada pustaka *scikit-learn* melalui parameter *average='macro'*, sehingga digunakan secara konsisten pada seluruh proses

optimasi *hyperparameter*, pemilihan model, dan optimasi *threshold*. Konsistensi tersebut memastikan bahwa metrik yang dioptimalkan selama pengembangan model sama dengan metrik yang digunakan pada tahap evaluasi. Penetapan formulasi ini sejak awal penelitian menjadi penting karena Opitz dan Burst (2021) menunjukkan bahwa *averaged F1* dan *F1 of averages* dapat menghasilkan nilai maupun peringkat model yang berbeda. Oleh karena itu, kejelasan formulasi yang digunakan diperlukan agar proses evaluasi berlangsung secara konsisten dan hasil penelitian dapat direproduksi.

Dalam penelitian klasifikasi kekuatan kata sandi, metrik-metrik tersebut beserta *confusion matrix* merupakan ukuran yang paling umum dilaporkan (Alajaji & Althaqeb, 2025; Divya et al., 2023; Shreya et al., 2024), dengan *confusion matrix* menjadi dasar untuk memeriksa jenis kesalahan yang dilakukan model (Khalid et al., 2024). Meskipun demikian, *accuracy* tidak memadai sebagai metrik tunggal pada data yang tidak seimbang karena ketidakseimbangan kelas dapat meningkatkan nilai akurasi tanpa mencerminkan performa yang sesungguhnya (Alajaji & Althaqeb, 2025). Ilustrasi konkret diberikan oleh Pande (2026), yang dengan klasifikasi tiga kelas memperoleh akurasi 76% tetapi *F1-Score* makro hanya 0,50 pada model XGBoost. Selisih tersebut menunjukkan bahwa *F1-Score* makro lebih informatif dalam konteks tidak seimbang dibandingkan akurasi maupun rata-rata berbobot (*weighted*) yang cenderung didominasi oleh kelas mayoritas.

Berbagai metrik evaluasi memiliki karakteristik dan tujuan penggunaan yang berbeda sehingga pemilihannya perlu disesuaikan dengan karakteristik permasalahan klasifikasi (Sokolova & Lapalme, 2009). Pada penelitian ini, *F1-Score* makro dipilih sebagai metrik utama karena memberikan bobot yang sama kepada setiap kelas, sehingga evaluasi tidak didominasi oleh kelas dengan jumlah sampel yang lebih banyak. Dengan demikian, performa model pada setiap kelas dapat dievaluasi secara lebih seimbang.

Selain metrik agregat, analisis per kelas juga memiliki arti penting tersendiri. Thai et al. (2024) menekankan pentingnya *recall* pada kelas lemah untuk mencegah model gagal mendeteksi kata sandi yang lemah. Pertimbangan ini penting karena kesalahan mengklasifikasikan kata sandi lemah sebagai lebih kuat dapat menimbulkan risiko keamanan yang lebih besar dibandingkan kesalahan sebaliknya. *Confusion matrix* memungkinkan analisis kesalahan semacam ini secara terperinci, seperti yang dilakukan Pande (2026), dengan memanfaatkannya untuk mengungkap banyaknya kata sandi lemah yang salah diklasifikasikan sebagai sedang akibat ambiguitas batas antara kedua kelas tersebut. Dengan demikian, evaluasi

yang baik tidak berhenti pada angka tunggal, melainkan menelaah bagaimana kesalahan terdistribusi antarkelas.

Validitas evaluasi juga bergantung pada disiplin pengujian, khususnya pemisahan data yang tepat. Studi terdahulu menggunakan pembagian data latih dan uji, seperti Sarkar dan Nandan (2022) yang memakai proporsi 80% berbanding 20%, atau *cross-validation*, seperti Darbutaitė et al. (2023) yang menerapkan *10-fold stratified cross-validation*. Penyetelan *hyperparameter* dilakukan pada tahap pelatihan, sedangkan kinerja akhir diukur pada data uji yang terpisah (Mo et al., 2026). Prinsip-prinsip ini, yaitu metrik yang sesuai, perhatian pada kelas minoritas, dan pemisahan data yang ketat, menjadi landasan untuk menilai model secara adil sekaligus memposisikan penelitian ini terhadap studi terdahulu, yang dirangkum pada Subbab 2.9.

2.9 Posisi dan Kontribusi Penelitian

Bagian-bagian sebelumnya telah meninjau konsep kekuatan kata sandi, ragam pendekatan penilaiannya, persoalan pelabelan, algoritma dan fitur, interpretabilitas, metodologi pengembangan, serta evaluasi model. Bagian ini mensintesis temuan-temuan tersebut untuk menetapkan posisi penelitian ini terhadap penelitian terdahulu. Tabel 2.3 merangkum sintesis tersebut yang disusun berdasarkan beberapa tema utama, yaitu kualitas label, orientasi evaluasi, optimasi *threshold*, interpretasi model dan penerapan sistem. Untuk setiap tema, tabel menampilkan penelitian terdahulu yang relevan, keterbatasannya, serta posisi penelitian ini.

Tabel 2.3 Ringkasan dan Posisi Penelitian Terdahulu

No	Tema	Penelitian Terdahulu	Keterbatasan	Posisi Penelitian Ini
1	Kualitas Label	Divya et al. (2023) memakai PARS; Pande (2026) memakai *zxcvbn*	Label agregat seperti PARS konsistensinya diragukan	Pelabelan ulang berbasis *zxcvbn* mengikuti praktik Pande, diterapkan pada *dataset* berbeda yang lebih besar dan beragam
2	Orientasi Evaluasi	(Khalid et al., 2024; Rehman et al., 2024; Sarkar & Nandan, 2022)	Dominan berfokus pada akurasi	Evaluasi berfokus pada *F1-Score* makro, dan *recall* per kelas

No	Tema	Penelitian Terdahulu	Keterbatasan	Posisi Penelitian Ini
3	Optimasi *Threshold*	Thai et al. (2024) paling dekat dengan menekankan *recall* kelas lemah	*Threshold* umumnya dibiarkan *default*; tidak ada kontrol yang dapat disetel terhadap kesalahan menilai kata sandi lemah sebagai kuat	Optimasi *threshold* berbatas keamanan asimetris
4	Interpretabilitas Model	(Shreya et al., 2024)	Interpretasi menggunakan LIME, yaitu metode interpretasi lokal yang bersifat *model agnostic*	Interpretasi menggunakan TreeSHAP yang dioptimalkan untuk model berbasis pohon serta mendukung analisis pada tingkat lokal dan global.
5	Penerapan Sistem	(Kayalvili et al., 2024)	Implementasi belum memadukan orientasi keamanan maupun interpretabilitas	Sistem web terintegrasi dengan klasifikasi dan *threshold* keamanan.

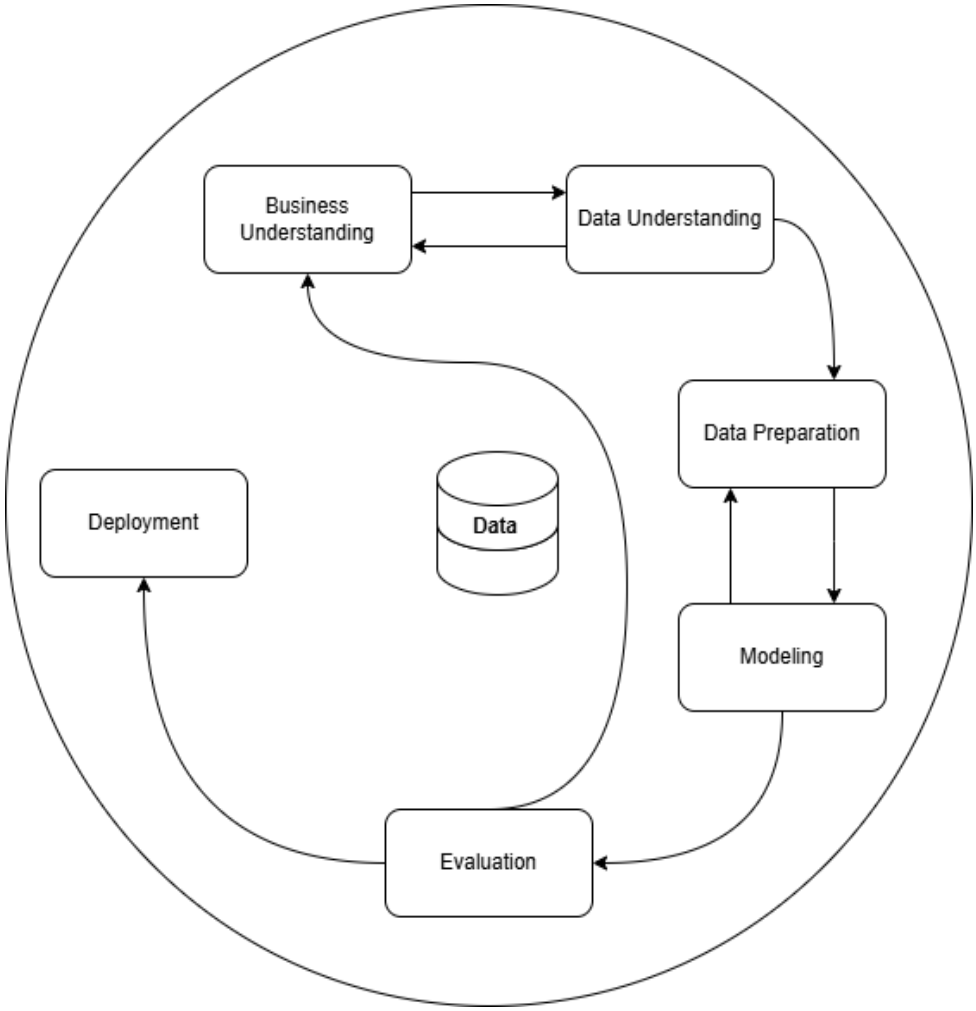
Sintesis tersebut memperlihatkan bahwa penelitian terdahulu unggul pada aspek tertentu, tetapi menyisakan celah yang saling berkaitan. Penggunaan *zxcvbn* sebagai acuan label telah dilakukan oleh Pande (2026) sehingga penelitian ini mengikuti praktik tersebut alih-alih mengeklaimnya sebagai kebaruan, meskipun diterapkan pada *dataset* yang berbeda dan jauh lebih besar. Penelitian terdahulu dominan menilai performa melalui akurasi; penelitian ini mengevaluasi menggunakan *F1-Score* makro dan *recall* per kelas agar performa pada kelas *weak*, yang jumlahnya paling sedikit, tidak tenggelam oleh kelas mayoritas.

Pada sisi mekanisme keputusan, Thai et al. (2024) telah menekankan pentingnya *recall* kelas lemah tetapi membiarkan *threshold* pada nilai *default* tanpa kontrol yang dapat disetel terhadap kesalahan menilai kata sandi lemah sebagai kuat; penelitian ini menutup celah tersebut melalui optimasi *threshold* berbatas keamanan asimetris. Pada sisi interpretabilitas, Shreya et al. (2024) menggunakan LIME yang bersifat aproksimasi lokal dan *model-agnostic*; penelitian ini memakai SHAP (*TreeSHAP*) yang konsisten dengan model berbasis pohon yang digunakan. Kebaruan-kebaruan tersebut disatukan dalam satu sistem yang utuh, implementasi oleh Kayalvili et al. (2024) belum mengabungkan orientasi keamanan maupun interpretabilitas,

sedangkan penelitian ini mewujudkan keduanya sekaligus ke dalam *prototype* sistem web yang terintegrasi.

BAB III
METODOLOGI PENELITIAN

Bab ini menguraikan metodologi penelitian yang digunakan untuk membangun sistem klasifikasi kekuatan kata sandi. Penelitian disusun mengikuti kerangka kerja CRISP-DM yang terdiri dari enam tahapan, yaitu *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment*. Tahap *Business Understanding* mendefinisikan permasalahan, tujuan teknis, serta kriteria keberhasilan penelitian. Tahap *Data Understanding* mengeksplorasi karakteristik dan kualitas *dataset* kata sandi serta mengidentifikasi keterbatasan pada pelabelan aslinya. Tahap *Data Preparation* mencakup pembersihan data, pelabelan ulang menggunakan *zxcvbn*, ekstraksi fitur, dan pembagian *dataset*. Tahap *Modeling* melatih dan membandingkan empat algoritma klasifikasi pada dua skenario penanganan ketidakseimbangan kelas, lengkap dengan optimasi hiperparameter dan *threshold* keputusan yang mempertimbangkan batasan keamanan. Tahap *Evaluation* mengukur performa seluruh model dan memilih model terbaik berdasarkan prosedur yang objektif. Tahap *Deployment* mengintegrasikan model terpilih ke dalam aplikasi web fungsional yang dilengkapi dengan interpretabilitas berbasis SHAP. Keseluruhan tahapan ini dirancang agar setiap keputusan metodologis dapat ditelusuri dan direproduksi.



Gambar 3.1 Diagram alur CRISP-DM

3.1 *Business Understanding*

Tahap *business understanding* bertujuan untuk mendefinisikan permasalahan yang akan diselesaikan, menentukan tujuan penelitian dari perspektif teknis, serta menetapkan kriteria keberhasilan yang dapat diukur.

3.1.1 Identifikasi Masalah

Penggunaan kata sandi yang lemah merupakan salah satu celah keamanan siber yang paling umum terjadi. Banyak pengguna membuat kata sandi berdasarkan kata yang mudah diingat, pola sederhana, atau informasi pribadi yang mudah ditebak. Sistem pemeriksa kekuatan kata sandi yang ada saat ini sebagian besar menggunakan aturan berbasis heuristik statis, seperti panjang minimum atau keharusan menggunakan karakter khusus, tanpa mempertimbangkan kompleksitas kata sandi secara menyeluruh.

Penelitian ini mengidentifikasi kebutuhan akan sistem yang mampu mengklasifikasikan kekuatan kata sandi secara lebih komprehensif menggunakan pendekatan *machine learning*, serta menyediakan penjelasan interpretabilitas berbasis SHAP yang memungkinkan pengguna memahami faktor-faktor yang memengaruhi klasifikasi kekuatan kata sandi mereka.

3.1.2 Tujuan Teknis

Berdasarkan identifikasi masalah di atas, tujuan teknis yang ditetapkan dalam penelitian ini adalah sebagai berikut:

a. Membangun model klasifikasi kekuatan kata sandi ke dalam tiga kelas, yaitu *weak, medium,* dan *strong,* menggunakan algoritma *machine learning*.
b. Mengidentifikasi fitur-fitur yang paling berpengaruh terhadap kekuatan kata sandi melalui analisis *feature importance*.
c. Mengintegrasikan model beserta kemampuan interpretabilitas berbasis SHAP ke dalam aplikasi web yang dapat digunakan langsung oleh pengguna.

3.1.3 Kriteria Keberhasilan

Model dinyatakan berhasil apabila memenuhi kriteria berikut:

a. Nilai akurasi pada data uji mencapai minimal 80%.
b. Nilai *F1-Score* makro pada data uji mencapai minimal 0,75.
c. Recall untuk kelas *weak* mencapai minimal 0,70, mengingat deteksi kata sandi lemah merupakan prioritas utama dari perspektif keamanan.

Selain ketiga kriteria keberhasilan di atas, ditetapkan pula satu spesifikasi keamanan yang bersifat batasan wajib, yaitu proporsi kesalahan klasifikasi kata sandi *weak* sebagai *strong* pada data uji tidak melebihi 5%. Berbeda dengan kriteria keberhasilan yang berperan sebagai target performa, spesifikasi ini merupakan syarat yang harus dipenuhi terlebih dahulu oleh konfigurasi model sebelum performanya diperbandingkan. Mekanisme penegakan spesifikasi ini melalui optimasi *threshold* diuraikan pada Subbab 3.4.4.

3.2 *Data Understanding*

Tahap *Data Understanding* bertujuan untuk memperoleh pemahaman menyeluruh terhadap *dataset* yang akan digunakan, meliputi karakteristik umum data, kualitas data, serta distribusi label. Pada tahap ini dilakukan serangkaian aktivitas eksplorasi data, yaitu pemeriksaan struktur dan kualitas *dataset*, analisis distribusi label, serta analisis hubungan

antara panjang kata sandi dan label untuk mengevaluasi validitas skema pelabelan aslinya. Hasil dari tahap ini menjadi dasar pengambilan keputusan pada tahap *Data Preparation*, khususnya terkait kebutuhan relabeling *dataset*.

3.2.1 Sumber Data

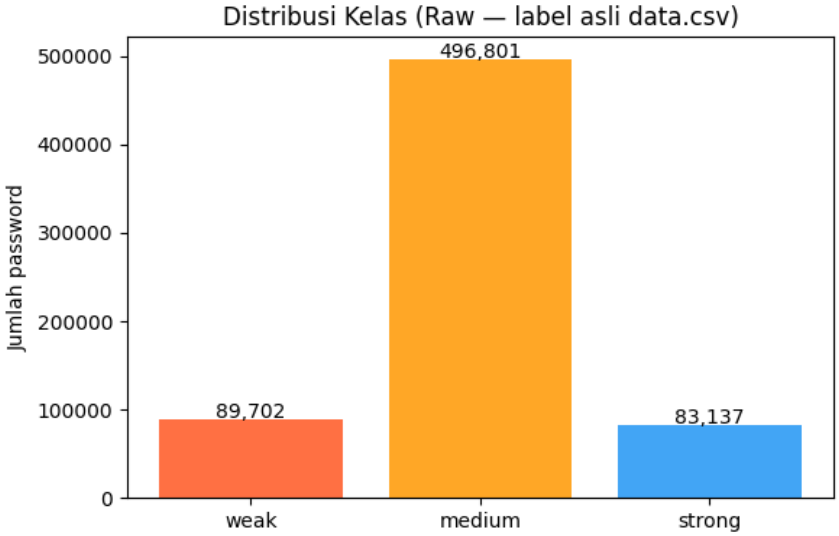
Dataset yang digunakan dalam penelitian ini adalah *Password Strength Classifier Dataset* yang diperoleh dari platform Kaggle (Bhavik Bansal, 2019). *Dataset* ini berisi 669.640 entri kata sandi beserta label kekuatannya. Label pada *dataset* asli menggunakan tiga kelas: 0 (*weak*), 1 (*medium*), dan 2 (*strong*), yang dihasilkan oleh *Password Analysis and Research System* (PARS) dari Georgia Institute of Technology.

3.2.2 Eksplorasi Awal Dataset

Sebelum dilakukan *preprocessing*, dilakukan eksplorasi awal untuk memahami karakteristik dan kualitas data. Hasil eksplorasi menunjukkan bahwa *dataset* memiliki 669.640 baris dan dua kolom, yaitu kolom *password* dan kolom *strength*. Ditemukan satu *missing value* pada kolom *password* dan tidak terdapat duplikat secara langsung pada data mentah. Kolom *strength* berisi label numerik dengan tiga nilai: 0 (*weak*), 1 (*medium*), dan 2 (*strong*).

3.2.3 Distribusi Label

Distribusi label asli pada *dataset* ditampilkan pada Gambar 3.2.

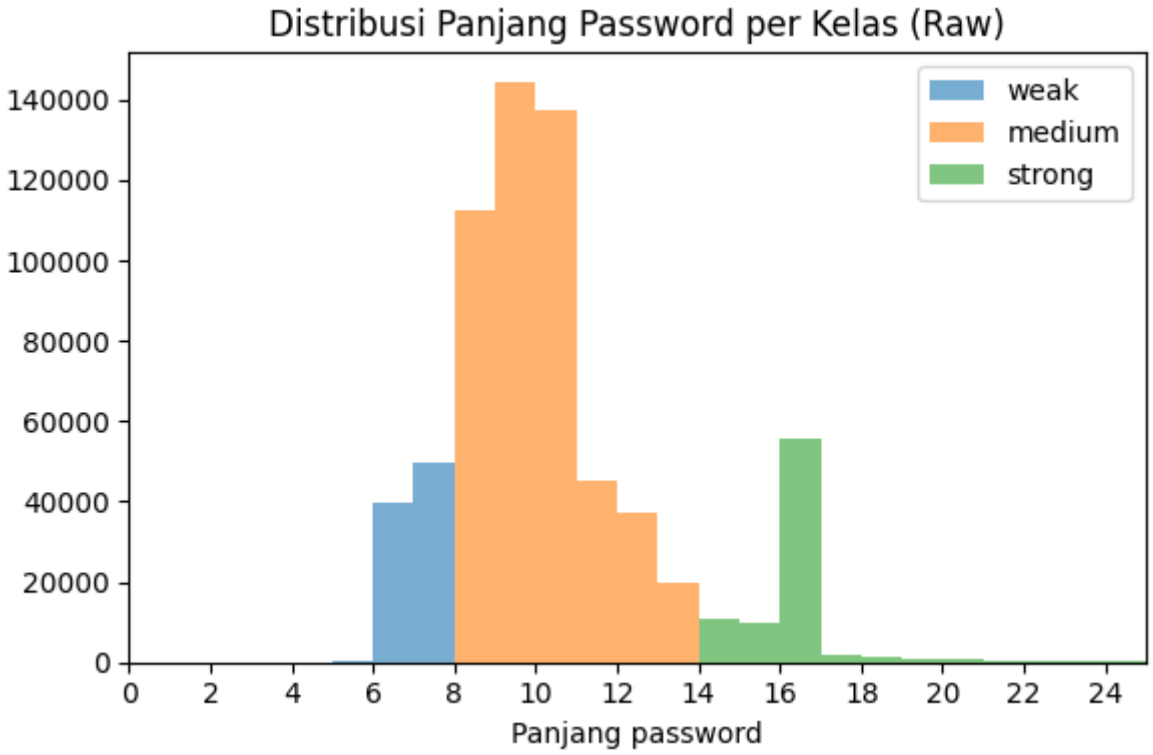


Gambar 3.2 Grafik distribusi label asli

Berdasarkan Gambar 3.2, distribusi label asli menunjukkan ketidakseimbangan yang sangat ekstrem. Kelas *medium* mendominasi dengan 496.801 data (74,2%), sementara kelas *weak* berjumlah 89.702 data (13,4%) dan kelas *strong* berjumlah 83.137 data (12,4%).

3.2.4 Analisis Hubungan Label dan Panjang Kata Sandi

Untuk memahami penyebab distribusi yang tidak seimbang tersebut, dilakukan analisis hubungan antara panjang karakter kata sandi dan label yang diberikan. Distribusi panjang kata sandi per kelas ditampilkan pada Gambar 3.3.



Gambar 3.3 Histogram distribusi panjang kata sandi per kelas

Berdasarkan Gambar 3.3, terlihat bahwa distribusi panjang kata sandi pada setiap kelas tidak saling tumpang tindih. Kelas *weak* hanya mencakup kata sandi dengan panjang pendek, kelas *medium* mencakup kata sandi dengan panjang menengah, dan kelas *strong* mencakup kata sandi dengan panjang lebih panjang. Hal ini diperkuat oleh statistik panjang kata sandi per kelas yang disajikan pada Tabel 3.1.

Tabel 3.1 Statistik panjang kata sandi per kelas

Kelas	Panjang Minimum	Panjang Maksimum	Rata-rata Panjang
Weak	1	7	6,55
Medium	8	13	9,62
Strong	14	220	15,93

Berdasarkan Tabel 3.1, kelas *weak* mencakup seluruh kata sandi dengan panjang 1–7 karakter, kelas *medium* mencakup kata sandi dengan panjang 8–13 karakter, dan kelas *strong* mencakup kata sandi dengan panjang 14 karakter atau lebih. Tidak ada tumpang tindih rentang panjang antarkelas. Temuan ini mengonfirmasi bahwa pelabelan asli pada *dataset* dilakukan semata-mata berdasarkan panjang karakter, tanpa mempertimbangkan kompleksitas atau karakteristik keamanan kata sandi yang sesungguhnya.

3.2.5 Keterbatasan Label Asli

Skema pelabelan berbasis panjang karakter memiliki keterbatasan yang mendasar dari perspektif keamanan. Kata sandi yang panjang belum tentu kuat apabila menggunakan pola yang mudah ditebak, dan sebaliknya, kata sandi pendek yang kompleks dapat memiliki tingkat keamanan yang lebih tinggi. Sebagai contoh, kata sandi "wertyuiop@1234" (14 karakter) diklasifikasikan sebagai *strong* pada label asli meskipun mengandung pola *keyboard* berurutan yang mudah dikenali oleh serangan *dictionary attack*. Sebaliknya, kata sandi "K#9mR" (5 karakter) diklasifikasikan sebagai *weak* meskipun kombinasi karakter yang digunakan relatif kompleks.

Keterbatasan ini menjadi justifikasi utama dilakukannya proses *relabeling* pada tahap *Data Preparation* menggunakan *zxcvbn* sebagai standar penilaian yang lebih komprehensif.

3.3 *Data Preparation*

Tahap *Data Preparation* bertujuan untuk mengubah data mentah menjadi *dataset* yang siap digunakan dalam proses pemodelan. Pada tahap ini dilakukan serangkaian aktivitas, yaitu penanganan *missing value* yang ditemukan pada tahap *Data Understanding*, lalu *relabeling dataset* menggunakan *zxcvbn* untuk menggantikan label asli yang terbukti tidak memadai, ekstraksi fitur dari setiap kata sandi menjadi representasi vektor numerik, analisis distribusi label hasil *relabeling*, serta pembagian *dataset* menjadi *training set*, *validation set,* dan *test set.* Output dari tahap ini adalah *dataset* berfitur lengkap dengan label yang valid dan siap digunakan pada tahap *Modeling*.

3.3.1 Penanganan *Missing Values*

Berdasarkan hasil eksplorasi pada tahap *Data Understanding*, dilakukan *preprocessing* untuk membersihkan data sebelum digunakan. Langkah pertama adalah menghapus baris dengan nilai *null* atau tipe data tidak valid pada kolom *password*, yang menghasilkan

penghapusan satu baris. Tidak ditemukan kata sandi dengan panjang nol karakter, tidak ditemukan duplikat pada data mentah, dan tidak ditemukan label yang tidak valid. *Dataset* bersih yang dihasilkan berjumlah 669.639 baris.

Seluruh *pipeline* ekstraksi fitur, pelabelan *zxcvbn*, dan proses inferensi mengikuti prinsip konsistensi representasi: satu-satunya normalisasi yang dilakukan adalah konversi ke tipe *string* dan pemotongan pada 100 karakter pertama, tanpa melakukan penghapusan spasi di awal atau akhir (*strip*). Keputusan ini didasarkan pada temuan bahwa karakter spasi merupakan bagian yang sah dari beberapa kata sandi dalam *dataset*.

3.3.2 *Relabeling* Dataset

Label asli *dataset* yang terbukti hanya berbasis panjang karakter diganti menggunakan *library zxcvbn*. Proses *relabeling* dilakukan dengan cara memasukkan setiap kata sandi ke dalam fungsi *zxcvbn*, yang kemudian menghasilkan skor kekuatan kata sandi dari 0 hingga 4. Skor tersebut selanjutnya dipetakan ke dalam tiga kelas label sebagaimana ditunjukkan pada Tabel 3.2.

Tabel 3.2 Skoring zxcvbn

Skor zxcvbn	Interpretasi	Label	Kelas
0 – 1	Sangat lemah hingga lemah	0	*Weak*
2	Cukup	1	*Medium*
3-4	Kuat hingga sangat kuat	2	*Strong*

Pemilihan skema pemetaan ini didasarkan pada pertimbangan keseimbangan jumlah kelas dan kesesuaiannya dengan konteks keamanan kata sandi. Skor 0 dan 1 digabungkan ke dalam kelas *weak* karena keduanya merepresentasikan kata sandi yang tidak layak digunakan dari perspektif keamanan. Skor 3 dan 4 digabungkan ke dalam kelas *strong* karena keduanya merepresentasikan kata sandi dengan tingkat keamanan yang baik. Proses *relabeling* berhasil diselesaikan untuk seluruh 669.639 kata sandi.

3.3.3 Ekstraksi Fitur

Setiap kata sandi direpresentasikan sebagai vektor fitur numerik agar dapat diproses oleh algoritma *machine learning*. Penelitian ini mengekstraksi total 25 fitur yang dikelompokkan ke dalam tiga kategori: 18 fitur struktural, 3 fitur RockYou, dan 4 fitur interaksi. Seluruh fitur bersifat struktural dalam arti dapat dihitung langsung dari komposisi karakter kata sandi atau melalui pencocokan dengan basis data RockYou, tanpa memanggil pustaka *zxcvbn* pada tahap

inferensi. Dengan demikian, model yang telah dilatih dapat melakukan prediksi secara mandiri tanpa bergantung pada *zxcvbn*, yang hanya digunakan pada tahap pelabelan selama pelatihan. Rincian seluruh fitur disajikan pada Tabel 3.3.

Tabel 3.3 Daftar seluruh fitur yang diekstraksi

No	Nama Fitur	Tipe	Deskripsi	Kelompok
1	*length*	int	Panjang kata sandi	Struktural
2	*upper*	int	Jumlah huruf kapital	Struktural
3	*lower*	int	Jumlah huruf kecil	Struktural
4	*digit*	int	Jumlah angka	Struktural
5	*symbol*	int	Jumlah simbol	Struktural
6	*space_count*	int	Jumlah karakter spasi	Struktural
7	*entropy*	float	Entropi Shannon dari distribusi karakter	Struktural
8	*common_pattern*	int	Keberadaan pola umum	Struktural
9	*unique_chars*	int	Jumlah karakter unik	Struktural
10	*unique_ratio*	float	Rasio karakter unik terhadap panjang	Struktural
11	*digit_ratio*	float	Rasio digit terhadap panjang	Struktural
12	*symbol_ratio*	float	Rasio simbol terhadap panjang	Struktural
13	*upper_ratio*	float	Rasio huruf kapital terhadap panjang	Struktural
14	*has_sequential*	int (0/1)	Keberadaan pola berurutan	Struktural
15	*has_repeated*	int (0/1)	Keberadaan karakter berulang 3 kali atau lebih	Struktural
16	*char_variety*	int (0-4)	Jumlah tipe karakter yang digunakan (0-4)	Struktural
17	*has_year*	int (0/1)	Keberadaan pola tahun 19xx atau 20[0-2]x	Struktural
18	*max_digit_run*	int	Panjang urutan digit berturut-turut terpanjang yang muncul dalam kata sandi.	Struktural
19	*in_rockyou*	int (0/1)	Kecocokan langsung kata sandi (*lowercase*) dengan entri rockyou.txt	RockYou
20	*rockyou_base_match*	int (0/1)	Keberadaan *base word* kata sandi dari kata	RockYou

No	Nama Fitur	Tipe	Deskripsi	Kelompok
			sandi dalam rockyou.txt (setelah menghapus angka/simbol di akhir)	
21	*rockyou_core_match*	int (0/1)	Kecocokan *core word* kata sandi dalam rockyou.txt (setelah menghapus angka/simbol/*underscore* di awal dan akhir)	RockYou
22	*ix_length_entropy*	float	Interaksi: length x entropy	Interaksi
23	*ix_length_unique_ratio*	float	Interaksi: length x unique_ratio	Interaksi
24	*ix_entropy_unique*	float	Interaksi entropy x unique_chars	Interaksi
25	*ix_core_year*	float (0/1)	Interaksi: rockyou_core_match x has_year	Interaksi

A. Kelompok Fitur Struktural (18 Fitur)

Fitur dalam kelompok ini mengekstraksi karakteristik kata sandi secara langsung dari komposisi karakter penyusunnya dan bersifat deterministik. Kelompok ini mencakup fitur dasar seperti panjang kata sandi (*length*), jumlah karakter per tipe (*upper*, *lower*, *digit*, *symbol*, *space_count*), entropi Shannon (*entropy*), jumlah dan rasio karakter unik, serta indikator pola seperti pola umum (*common_pattern*), pola sekuensial (*has_sequential*), karakter berulang (*has_repeated*), dan variasi tipe karakter (*char_variety*). Tiga fitur ditambahkan pada versi akhir berdasarkan analisis diagnostik, yaitu *space_count* (jumlah spasi), *has_year* (deteksi pola tahun 19xx atau 20[0-2]x), dan *max_digit_run* (panjang rentetan digit terpanjang).

Salah satu fitur yang memerlukan penjelasan lebih lanjut adalah entropi Shannon. Entropi Shannon mengukur tingkat ketidakpastian atau keacakan suatu kata sandi berdasarkan frekuensi kemunculan setiap karakter. Nilai entropi yang lebih tinggi menandakan kata sandi yang lebih sulit ditebak. Entropi Shannon dihitung menggunakan Persamaan 3.1 berikut:

$$H = -\sum p(x) \log_2 p(x) \quad (3.1)$$

dengan H adalah nilai entropi dan p(x) adalah probabilitas kemunculan karakter x dalam kata sandi.

B. Kelompok Fitur RockYou (3 Fitur)

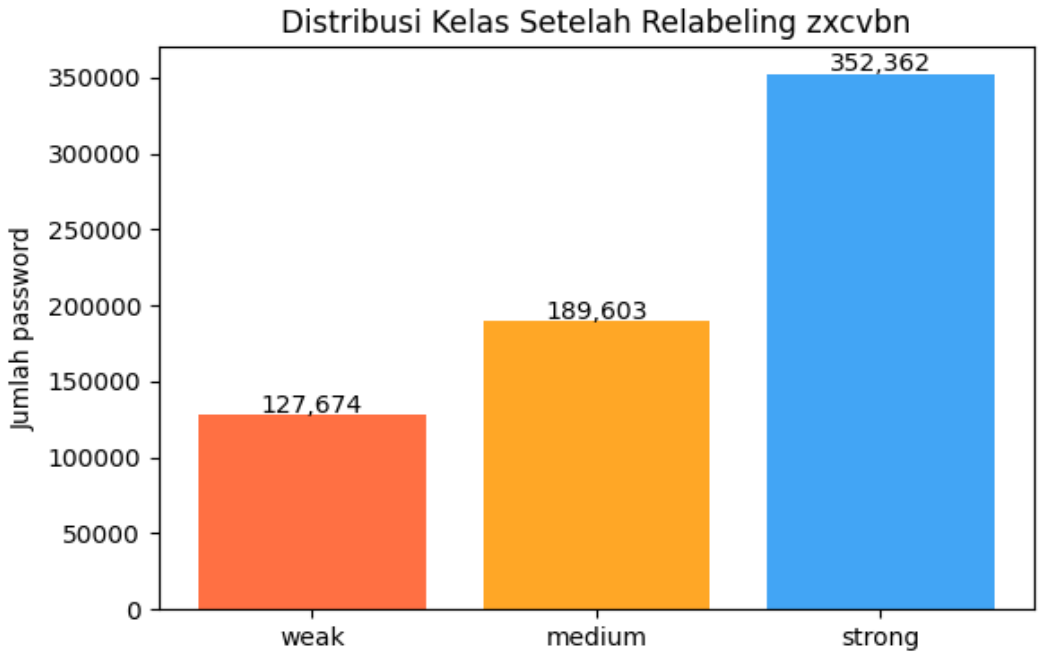
Tiga fitur ini diekstraksi melalui pencocokan kata sandi dengan basis data *rockyou.txt*, yaitu daftar 13,8 juta kata sandi unik yang bocor akibat insiden keamanan RockYou. Fitur *in_rockyou* mendeteksi kecocokan secara langsung antara kata sandi (huruf kecil) dan entri dalam file *rockyou.txt*. Fitur *rockyou_base_match* mendeteksi kecocokan bentuk dasar kata sandi setelah menghapus digit dan simbol pada bagian akhir. Fitur *rockyou_core_match*, yang ditambahkan pada versi akhir, mendeteksi kecocokan setelah menghapus digit, simbol, dan *underscore* di bagian awal maupun akhir sehingga mampu menangkap pola seperti "1988ponce" yang tidak tertangkap oleh *rockyou_base_match.*

C. Kelompok Fitur Interaksi (4 Fitur)

Fitur interaksi dibentuk dari perkalian dua fitur yang secara semantik berkaitan, untuk menangkap hubungan nonlinier antarfitur. Tiga fitur pertama menggabungkan fitur struktural yang saling memperkuat, yaitu *ix_length_entropy* (panjang x entropi), *ix_length_unique_ratio* (panjang x rasio karakter unik), dan *ix_entropy_unique* (entropi x jumlah karakter unik). Fitur keempat, *ix_core_year* (*rockyou_core_match* x *has_year*), ditambahkan untuk menangkap pola kata bocor yang dikombinasikan dengan tahun, yang teridentifikasi sebagai pola dominan dalam kesalahan klasifikasi *weak→strong* pada versi sebelumnya.

3.3.4 Distribusi Label Setelah *Relabeling*

Setelah proses *relabeling* menggunakan *zxcvbn*, distribusi label berubah secara signifikan dibandingkan dengan label asli. Distribusi label hasil *relabeling* disajikan pada Gambar 3.4.



Gambar 3.4 Distribusi kelas setelah *relabeling zxcvbn*

Perbandingan antara distribusi asli dan distribusi label hasil *relabeling* menunjukkan perubahan yang signifikan. Kelas *medium* mengalami penurunan drastis dari 496.801 data (74,2%) menjadi 189.603 data (28,3%), sementara kelas *strong* meningkat secara signifikan dari 83.137 data (12,4%) menjadi 352.362 data (52,6%). Kelas *weak* juga meningkat dari 89.701 data (13,4%) menjadi 127.674 data (19,1%). Perubahan ini mencerminkan bahwa *zxcvbn* menilai bahwa sebagian besar kata sandi yang sebelumnya diklasifikasikan sebagai *medium* berdasarkan panjang karakter ternyata memiliki karakteristik keamanan yang lebih kuat secara aktual. Distribusi hasil *relabeling* tetap menunjukkan ketidakseimbangan kelas dengan kelas *strong* mendominasi sebesar 52,6%. Kondisi ini menjadi salah satu pertimbangan pada tahap pemodelan.

3.3.5 Pembagian Dataset

Dataset dibagi menjadi tiga bagian menggunakan *stratified splitting* untuk menjaga proporsi kelas pada setiap subset. *Stratified splitting* dipilih untuk memastikan bahwa distribusi kelas pada *training set, validation set,* dan *test set* tetap representatif terhadap distribusi keseluruhan *dataset*. Skema pembagian *dataset* ditunjukkan pada Tabel 3.4.

Tabel 3.4 Pembagian dataset

Subset	Jumlah Sample	Proporsi	Fungsi
Train Main	455.354	68%	Pelatihan model + *cross-validation*
Validation	80.357	12%	Pemilihan model dan optimasi *threshold*
Test	133.928	20%	Evaluasi final
Total	669.639	100%	-

Pembagian ini menghasilkan tiga subset dengan fungsi yang berbeda. *Training set* (455.354 sampel, 68%) digunakan untuk melatih model dengan *cross-validation*. *Validation set* (80.357 sampel, 12%) digunakan untuk pemilihan model terbaik dan optimasi *threshold* tanpa melibatkan *test set*. *Test set* (133.928 sampel, 20%) hanya digunakan satu kali pada evaluasi akhir untuk menghasilkan angka performa yang dapat dilaporkan secara objektif. Pemisahan *test set* sejak awal dan tidak menggunakannya selama proses pemilihan model dilakukan untuk menghindari kebocoran informasi (*data leakage*) yang dapat menyebabkan estimasi performa model menjadi terlalu optimis.

3.4 *Modeling*

Tahap *Modeling* bertujuan untuk membangun dan melatih model klasifikasi kekuatan kata sandi menggunakan *dataset* yang telah disiapkan pada tahap *Data Preparation*. Pada tahap ini dilakukan serangkaian aktivitas, yaitu pemilihan dan penerapan empat algoritma klasifikasi *supervised learning*, eksperimen pada dua skenario penanganan ketidakseimbangan kelas, optimasi hiperparameter untuk setiap model, serta optimasi *threshold* keputusan dengan mempertimbangkan batasan keamanan. Output dari tahap ini adalah model terbaik beserta konfigurasi *threshold* optimal yang siap dievaluasi pada tahap *Evaluation*.

3.4.1 Algoritma *Machine Learning* yang Digunakan

Empat algoritma *supervised learning* yang digunakan dalam penelitian ini adalah *Logistic Regression*, *Random Forest*, *Support Vector Machine* (SVM) Linear, dan XGBoost, yang mekanisme dasarnya telah dijelaskan pada Subbab 2.4. Keempat algoritma tersebut dipilih karena keragaman pendekatan yang diwakili oleh masing-masing algoritma sehingga perbandingan yang dihasilkan dapat memberikan gambaran yang komprehensif tentang

karakteristik data. Subbab ini menguraikan konfigurasi penerapan masing-masing algoritma pada penelitian ini.

Untuk *Logistic Regression*, klasifikasi multikelas diterapkan menggunakan pendekatan *multinomial* dengan fungsi softmax Persamaan 2.1 karena penelitian ini melibatkan tiga kelas kekuatan kata sandi. Untuk SVM, digunakan *kernel* linier karena dataset memiliki jumlah fitur yang cukup dan distribusi data yang memungkinkan pemisahan secara linier, dengan strategi *one-versus-rest* (OvR) untuk menangani klasifikasi multikelas, yaitu satu model SVM dibangun untuk setiap kelas dengan memisahkan kelas tersebut dari seluruh kelas lainnya. Adapun XGBoost diterapkan dengan regularisasi L1 dan L2 serta teknik *subsampling* baris dan kolom sebagaimana dijelaskan pada Subbab 2.4, dengan nilai setiap parameter ditentukan melalui proses optimasi *hyperparameter* yang diuraikan pada Subbab 3.4.3.

3.4.2 Skenario Eksperimen

Untuk mengetahui pengaruh ketidakseimbangan kelas terhadap performa model, setiap algoritma dilatih dalam dua skenario eksperimen sebagai berikut:

A. Skenario 1 Tanpa Penanganan Ketidakseimbangan (*Without Balancing*)
   Model dilatih langsung menggunakan *training set* dengan distribusi kelas asli hasil *relabeling*, yaitu *weak* 19,1%, *medium* 28,3%, dan *strong* 52,6%. Skenario ini bertujuan untuk mengetahui performa model pada kondisi distribusi data yang sebenarnya.
B. Skenario 2 dengan Penanganan Ketidakseimbangan (*With Balancing*)
   Model dilatih menggunakan *training set* yang telah diseimbangkan dengan teknik *random undersampling*. Teknik ini bekerja dengan mengurangi jumlah sampel pada kelas mayoritas secara acak hingga seluruh kelas memiliki jumlah sampel yang seimbang. *Random undersampling* dipilih karena *dataset* memiliki jumlah data yang besar sehingga pengurangan sampel pada kelas mayoritas tidak mengurangi representativitas data secara signifikan.

Perbandingan antara kedua skenario ini dilakukan untuk menentukan apakah penanganan ketidakseimbangan kelas memberikan peningkatan performa yang signifikan pada *dataset* penelitian ini.

3.4.3 Optimasi *Hyperparameter*

Pencarian *hyperparameter* optimal dilakukan menggunakan *RandomizedSearchCV* dengan *3-fold cross-validation. RandomizedSearchCV* bekerja dengan mengambil sampel kombinasi hiperparameter secara acak dari ruang pencarian yang telah ditentukan, kemudian mengevaluasi setiap kombinasi menggunakan *cross-validation*. Pendekatan ini dipilih karena lebih efisien secara komputasi dibandingkan dengan *GridSearchCV* yang mengevaluasi seluruh kombinasi secara menyeluruh, terutama untuk *dataset* berskala besar seperti yang digunakan dalam penelitian ini. Metrik yang digunakan sebagai acuan untuk memilih *hyperparameter* terbaik adalah *F1-Score* makro karena metrik ini memberikan bobot yang sama pada setiap kelas tanpa dipengaruhi oleh ketidakseimbangan distribusi kelas. Ruang pencarian *hyperparamter* untuk setiap algoritma yang digunakan dalam proses optimasi disajikan pada Tabel 3.5.

Tabel 3.5 Grid *hyperparameter* per algoritma pada *RandomizedSearchCV*

Algoritma	Grid Hyperparameter	n_iter	Subsample
Logistic Regression	C: [0.01, 0.1, 1, 10]	4	Tidak
Random Forest	n_estimators: [200, 300] max_depth: [15, 20, None] min_samples_leaf: [3, 5, 10] max_features: ["sqrt", 0.5]	15	Tidak
SVM *Linear* (CalibratedClassifierCV)	C: [0.01, 0.1, 1]	3	Tidak
XGBoost	n_estimators: [200, 300, 400] learning_rate: [0.05, 0.08, 0.1, 0.15] max_depth: [3, 4, 5, 6] subsample: [0.7, 0.8, 0.9, 1.0] colsample_bytree: [0.7, 0.8, 0.9, 1.0] reg_alpha: [0, 0.1, 0.5, 1.0] reg_lambda: [1.0, 1.5, 2.0] min_child_weight: [1, 3, 5] gamma: [0, 0.1, 0.3, 0.5]	50	Ya (~20%)

Untuk XGBoost, proses *tuning* dilakukan pada subset data sebesar 20% dari *X_train_main* (sekitar 91.070 sampel) menggunakan *StratifiedShuffleSplit* untuk efisiensi komputasi. Setelah *hyperparameter* terbaik ditemukan pada subset ini, model di-refit pada seluruh *X_train_main* (455.354 sampel). Perlakuan ini diterapkan secara simetris pada kedua kondisi eksperimen.

3.4.4 Optimasi *Threshold*

Secara *default*, model klasifikasi probabilistik memprediksi kelas menggunakan mekanisme *argmax* dari probabilitas yang dihasilkan. Pada penelitian ini, mekanisme prediksi dimodifikasi menjadi *argmax(proba/threshold)*, di mana *threshold* yang lebih kecil membuat suatu kelas lebih mudah dipilih sebagai prediksi. Optimasi *threshold* dilakukan pada *validation set* untuk memenuhi batasan keamanan yang telah ditetapkan.

Batasan keamanan utama yang diterapkan adalah MAX_WEAK_TO_STRONG = 0,05, artinya maksimal 5% kata sandi berlabel *weak* (*ground truth zxcvbn*) boleh diprediksi sebagai *strong* oleh model. Batasan ini telah ditetapkan pada Subbab 3.1.3 sebelum eksperimen dimulai, berdasarkan pertimbangan bahwa misklasifikasi dari kelas lemah ke kelas kuat (*weak→strong*) berpotensi memberikan rasa aman yang palsu kepada pengguna, sementara misklasifikasi satu tingkat (*weak→medium*) masih memberikan peringatan dan dianggap dapat diterima.

Optimasi dilakukan melalui *grid search* yang mencoba seluruh kombinasi *threshold* untuk ketiga kelas secara sistematis. *Threshold* kelas *weak* dicoba dari 0,10 hingga 0,55, *threshold* kelas *medium* dan kelas *strong* masing-masing dicoba dari 0,15 hingga 0,65, dengan interval 0,05 di antara setiap nilai yang diuji. Dari seluruh kombinasi yang memenuhi batasan weak→strong ≤ 5%, dipilih kombinasi dengan nilai *F1-Score* makro tertinggi. Penggunaan *validation set*, bukan *test set*, dalam proses ini dilakukan untuk menghindari kebocoran informasi yang dapat menyebabkan estimasi performa model menjadi terlalu optimis.

3.5 *Evaluation*

Tahap *Evaluation* bertujuan untuk mengukur dan membandingkan performa seluruh model yang telah dibangun pada tahap *Modeling*, serta memilih model terbaik yang akan digunakan pada tahap *Deployment*. Pada tahap ini dilakukan serangkaian aktivitas, yaitu pengukuran performa setiap model menggunakan metrik evaluasi yang telah ditetapkan, pemilihan model terbaik berdasarkan prosedur yang objektif, serta evaluasi final model terpilih pada *test set* yang belum pernah digunakan sebelumnya. Output dari tahap ini adalah model terpilih beserta angka performa final yang dapat dilaporkan secara ilmiah.

3.5.1 Metrik Evaluasi

Evaluasi performa model pada penelitian ini dilakukan menggunakan *accuracy*, *precision*, *recall*, *F1-Score*, dan *F1-Score* makro. Definisi, persamaan, serta interpretasi masing-masing

metrik telah dijelaskan pada Subbab 2.8. Oleh karena itu, subbab ini berfokus pada penerapan metrik tersebut dalam proses evaluasi model pada penelitian ini. *Accuracy* digunakan untuk memberikan gambaran umum mengenai proporsi prediksi yang benar terhadap seluruh data uji. Sementara itu, *precision*, *recall*, dan *F1-Score* dihitung untuk setiap kelas (*weak*, *medium*, dan *strong*) guna mengevaluasi performa model secara lebih rinci pada masing-masing kelas.

F1-Score makro digunakan sebagai metrik utama dalam pemilihan model, optimasi *hyperparameter*, maupun optimasi *threshold*. Metrik ini dipilih karena memberikan bobot yang sama kepada setiap kelas sehingga evaluasi tidak didominasi oleh kelas dengan jumlah sampel yang lebih banyak. Selain itu, penelitian ini memiliki fokus khusus pada kemampuan model dalam mendeteksi kata sandi *weak* sehingga *recall* kelas *weak* ditetapkan sebagai salah satu kriteria keberhasilan sebagaimana diuraikan pada Subbab 3.1.3. Adapun pengendalian kesalahan klasifikasi kata sandi *weak* menjadi *strong*, yang merupakan jenis kesalahan dengan risiko keamanan paling tinggi, dilakukan melalui batasan pada proses optimasi *threshold* yang dijelaskan pada Subbab 3.4.4.

3.5.2 Prosedur Pemilihan Model

Pemilihan model terbaik dilakukan melalui prosedur berikut. Pertama, seluruh model dari keempat algoritma pada kedua skenario eksperimen (*with balancing* dan *without balancing*) dievaluasi menggunakan *validation set*. Kedua, model dengan nilai *F1-Score* makro tertinggi pada *validation set* dipilih sebagai kandidat model terbaik. Penggunaan *validation set*, bukan *test set*, sebagai dasar pemilihan model bertujuan untuk menjaga objektivitas evaluasi final. *Test set* tidak digunakan dalam proses pemilihan model karena harus tetap independen hingga tahap evaluasi akhir.

Apabila dua model terbaik menunjukkan performa yang sulit dibedakan berdasarkan nilai *F1-Score* makro pada *validation set*, maka dilakukan uji signifikansi statistik menggunakan *paired bootstrap* dengan 2.000 replikasi terhadap perbedaan (*difference*) *F1-Score* makro kedua model. Pengujian dilakukan dengan mengambil sampel ulang secara berpasangan pada *indeks* data *validation set* yang sama untuk kedua model sehingga variasi yang diukur berasal dari perbedaan model, bukan dari perbedaan sampel yang dievaluasi. Hasil uji ini digunakan untuk menentukan apakah selisih *F1-Score* makro yang diamati merupakan perbedaan yang signifikan secara statistik atau hanya disebabkan oleh variasi pengambilan sampel.

Selain uji signifikansi, dilakukan pula analisis sensitivitas terhadap formulasi *F1-Score* makro yang digunakan. Analisis ini dilakukan dengan menghitung kembali performa kedua model menggunakan formulasi *F1 of averages* sebagai pembanding terhadap formulasi *averaged F1* yang menjadi metrik utama penelitian. Tujuan analisis ini adalah untuk memastikan bahwa keputusan pemilihan model tetap konsisten terhadap perubahan formulasi *F1-Score* makro.

Selain nilai *F1-Score* makro pada *validation set*, nilai *cross-validation score* juga digunakan untuk memverifikasi konsistensi performa model selama proses pelatihan. Model yang menunjukkan selisih besar antara performa pada *validation set* dan *cross-validation score* mengindikasikan kecenderungan *overfitting* yang perlu diwaspadai. Apabila hasil verifikasi menunjukkan bahwa dua model terbaik tidak dapat dibedakan secara meyakinkan berdasarkan *F1-Score* makro, maka konsistensi performa antara *validation set* dan *cross-validation score* digunakan sebagai kriteria sekunder dalam penetapan model final.

3.5.3 Evaluasi Final

Setelah model terbaik terpilih dan *threshold* dioptimalkan menggunakan *validation set,* dilakukan evaluasi final menggunakan *test set* yang belum pernah digunakan selama proses pelatihan maupun pemilihan model. Evaluasi final menghasilkan angka performa yang bersifat definitif dan dapat dilaporkan sebagai representasi kemampuan model pada data yang belum pernah dilihat sebelumnya (*unseen data*).

Hasil evaluasi final disajikan dalam bentuk *classification report* yang memuat nilai *precision*, *recall*, dan *F1-Score* untuk setiap kelas secara terpisah, serta nilai *accuracy* dan *F1-Score* makro secara keseluruhan. Selain itu, juga diperiksa pemenuhan batasan keamanan *weak→strong* ≤ 5% pada *test set.*

3.6 *Deployment*

Tahap *Deployment* bertujuan untuk mengintegrasikan model terpilih ke dalam aplikasi yang dapat digunakan secara langsung oleh pengguna. Pada tahap ini dilakukan serangkaian aktivitas, yaitu penyimpanan model terpilih beserta seluruh konfigurasinya, pengembangan *backend* sebagai layanan prediksi berbasis API (*Application Programming Interface*), pengembangan *frontend* sebagai antarmuka pengguna, serta implementasi interpretabilitas model menggunakan SHAP. Output dari tahap ini adalah aplikasi web yang fungsional dan

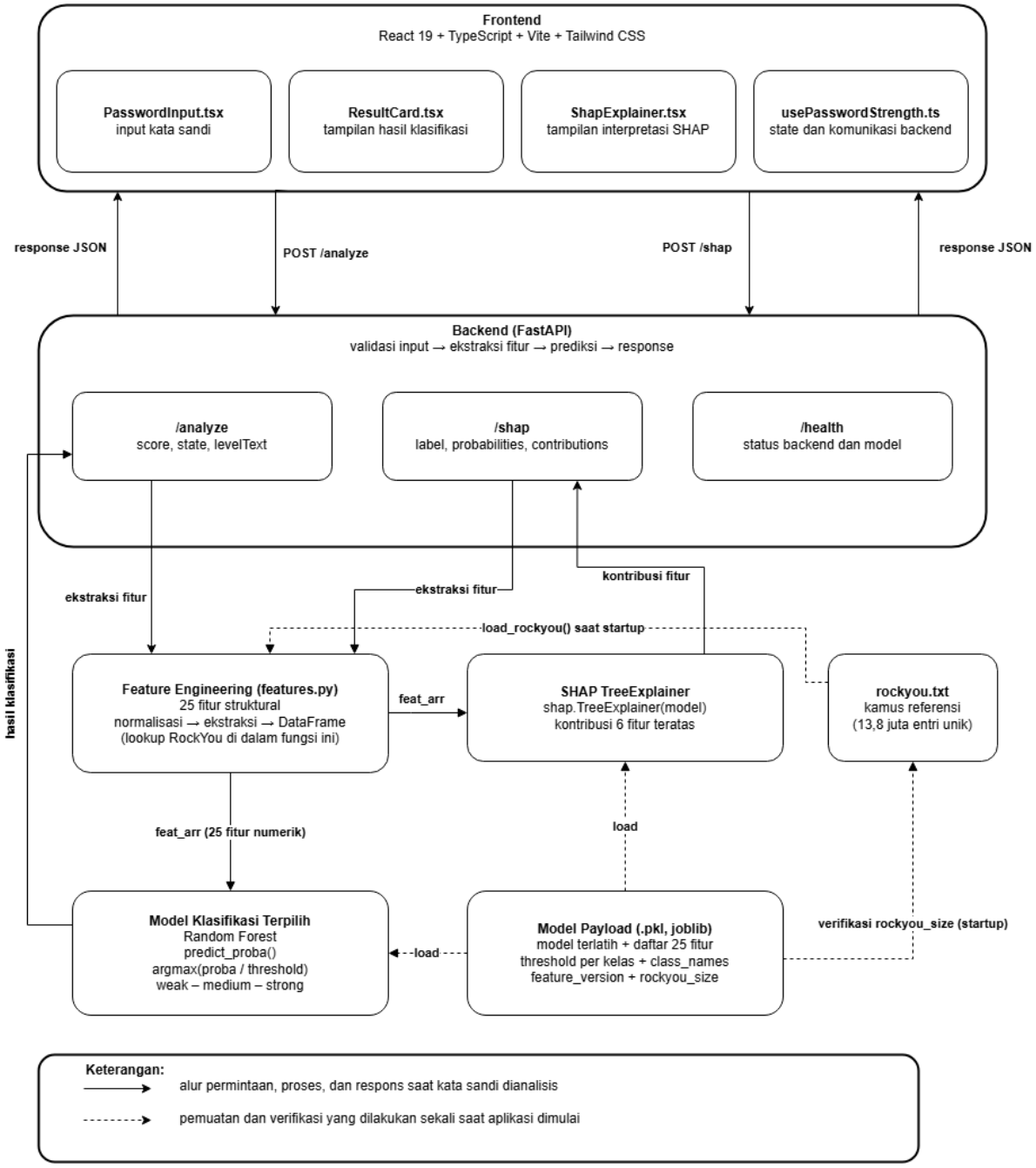
dapat diakses oleh pengguna untuk menganalisis kekuatan kata sandi secara *real-time* beserta penjelasan interpretabilitas berbasis SHAP yang dapat diakses secara interaktif.

3.6.1 Penyimpanan Model

Model terpilih beserta seluruh konfigurasinya disimpan menggunakan *library* joblib dalam format *pickle*. Komponen yang disimpan dalam satu paket meliputi objek model terlatih, daftar 25 nama fitur beserta urutannya, nilai *threshold* optimal untuk setiap kelas, nama algoritma model terpilih, kondisi eksperimen, serta metrik *F1-Score* makro untuk evaluasi final. Penyimpanan seluruh komponen dalam satu paket bertujuan untuk memastikan konsistensi antara proses ekstraksi fitur pada aplikasi web dan proses yang dilakukan selama pelatihan sehingga tidak terjadi ketidaksesuaian yang dapat memengaruhi hasil prediksi. Saat aplikasi dijalankan, sistem melakukan verifikasi otomatis untuk memastikan jumlah fitur yang diekstraksi identik dengan jumlah fitur yang diharapkan oleh model, termasuk verifikasi *fingerprint dataset* RockYou untuk memastikan konsistensi fitur berbasis *database* kata sandi yang bocor.

3.6.2 Arsitektur Aplikasi Web

Aplikasi web dikembangkan menggunakan arsitektur *full-stack* yang memisahkan komponen *backend* dan *frontend* secara eksplisit. Gambar 3.5 menunjukkan arsitektur keseluruhan sistem.



Gambar 3.5 Diagram Arsitektur Sistem

Gambar 3.5 menunjukkan arsitektur keseluruhan sistem yang terdiri dari empat lapisan utama. *Frontend* menerima input kata sandi dari pengguna dan mengirimkannya ke *backend* melalui *endpoint* /analyze untuk klasifikasi atau /shap untuk interpretabilitas. *Backend* meneruskan kata sandi ke modul *feature engineering* untuk diekstraksi menjadi 25 fitur struktural yang kemudian diproses oleh model klasifikasi terpilih untuk menghasilkan prediksi kelas. Untuk permintaan interpretabilitas, *SHAP TreeExplainer* menggunakan model yang

sama untuk menghitung kontribusi tiap fitur terhadap hasil prediksi. Seluruh komponen model dimuat dari *payload model* saat aplikasi pertama kali dijalankan.

A. *Backend*

Komponen *backend* dikembangkan menggunakan *framework* FastAPI berbasis Python. FastAPI dipilih karena mendukung pembuatan API yang cepat, memiliki validasi input otomatis berbasis Pydantic, dan kompatibel dengan seluruh *library* Python yang digunakan selama pengembangan model. *Backend* menyediakan tiga *endpoint* utama sebagaimana ditunjukkan pada Tabel 3.6.

Tabel 3.6 *Endpoint Backend*

No	Endpoint	Method	Deskripsi
1	/analyze	POST	Menerima input kata sandi dan mengembalikan hasil klasifikasi beserta probabilitas untuk setiap kelas.
2	/shap	POST	Menerima input kata sandi dan mengembalikan nilai SHAP per fitur untuk interpretabilitas.
3	/health	GET	Memverifikasi status *backend* dan model yang dimuat.

Proses yang terjadi di *backend* saat menerima *request* analisis kata sandi pada *endpoint* /analyze adalah sebagai berikut:

a. Validasi input menggunakan Pydantic untuk memastikan kata sandi tidak kosong.
b. Ekstraksi 25 fitur dari kata sandi menggunakan *pipeline* yang identik dengan proses pelatihan.
c. Prediksi kelas menggunakan model terpilih dengan *threshold* yang telah dioptimalkan melalui mekanisme *argmax*(*proba/threshold*).

B. *Frontend*

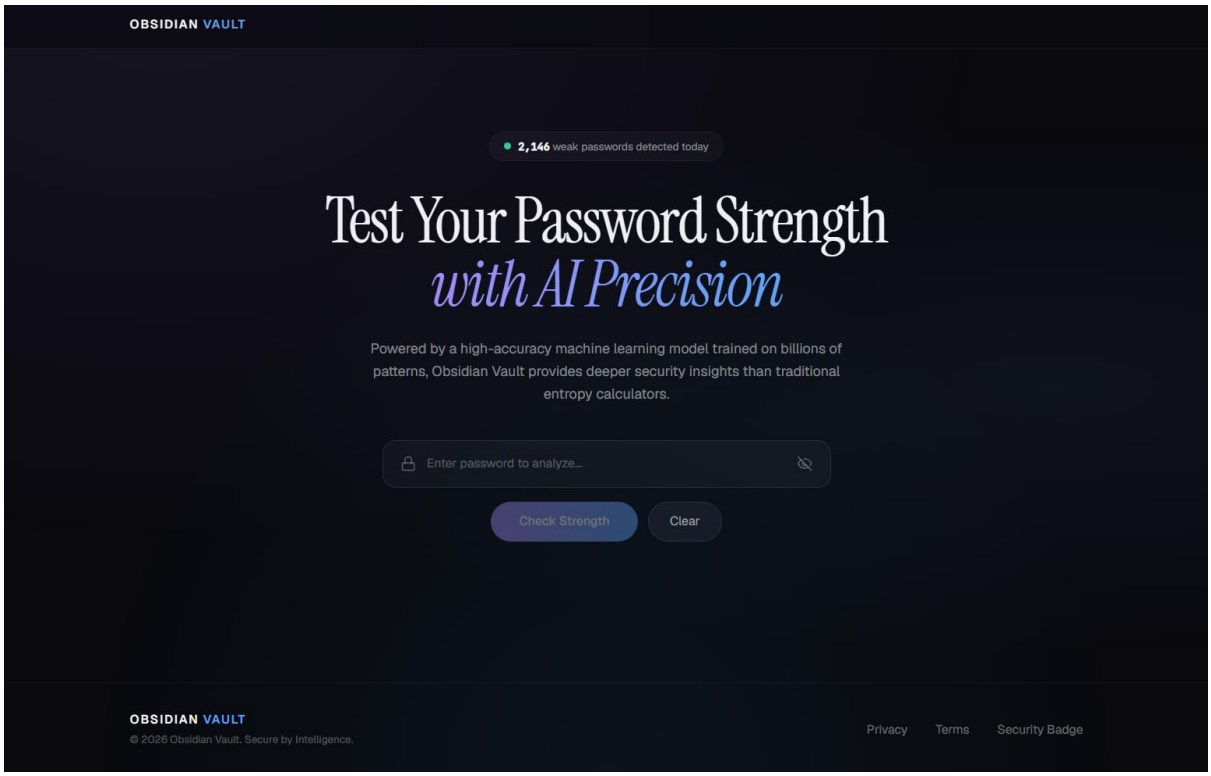
Komponen *frontend* dikembangkan menggunakan React 19 dengan TypeScript sebagai bahasa pemrograman, Vite sebagai *build tool*, Tailwind CSS untuk *styling*, TanStack Query untuk manajemen *state* data asinkron, serta Axios untuk komunikasi dengan *backend* melalui *Hypertext Transfer Protocol* (HTTP). Antarmuka tersusun atas empat komponen utama sebagaimana ditunjukkan pada Gambar 3.5, yaitu PasswordInput.tsx untuk masukan kata sandi, ResultCard.tsx untuk menampilkan hasil klasifikasi, ShapExplainer.tsx untuk menampilkan interpretasi SHAP, serta usePasswordStrength.ts yang mengelola *state* dan komunikasi dengan *backend*. *Frontend* berkomunikasi dengan *backend* melalui

HTTP menggunakan Axios dengan batas waktu (*timeout*) 5 detik. Apabila *backend* tidak dapat dijangkau dalam batas waktu tersebut, sistem menampilkan pesan *error* secara eksplisit kepada pengguna tanpa menggunakan mekanisme *fallback*.

3.6.3 Implementasi Antarmuka Pengguna

Antarmuka pengguna diimplementasikan sebagai aplikasi *single-page* yang berfokus pada satu halaman fungsional utama. Halaman utama ini menjadi titik masuk pengguna untuk mengakses sistem. Halaman ini terdiri dari dua bagian utama yang ditampilkan secara berurutan.

Bagian pertama adalah formulir *input* kata sandi sebagaimana ditunjukkan pada Gambar 3.6. Pengguna memasukkan kata sandi pada kolom *input* yang dilengkapi dengan tombol *show/hide* untuk menampilkan atau menyembunyikan karakter kata sandi. Setelah menekan tombol *Check Strength*, sistem mengirimkan kata sandi ke *backend* untuk dianalisis.



Gambar 3.6 Tampilan antarmuka pengguna

Bagian kedua menampilkan hasil analisis yang disajikan melalui komponen *ResultCard*. Komponen ini menampilkan *arc ring* yang menunjukkan kelas kekuatan kata sandi dengan tiga posisi tetap (*weak* pada sepertiga pertama, *medium* pada dua pertiga, dan *strong* pada posisi penuh) beserta badge label kelas, daftar kelemahan spesifik yang terdeteksi pada kata sandi,

serta komponen interpretabilitas SHAP yang dapat diakses pengguna. Tampilan lengkap antarmuka dan contoh hasil analisis akan disajikan pada Bab 4.

3.6.4 Interpretabilitas SHAP

Interpretabilitas model diimplementasikan menggunakan SHAP (*SHapley Additive exPlanations*), yaitu metode berbasis teori permainan yang memberikan penjelasan per prediksi secara individual. SHAP mengkuantifikasi kontribusi setiap fitur terhadap prediksi model untuk setiap kata sandi yang dianalisis sehingga pengguna dapat memahami alasan sebuah kata sandi diklasifikasikan sebagai *weak*, *medium*, atau *strong* berdasarkan fitur strukturalnya.

Implementasi menggunakan *TreeExplainer* dari *library* SHAP yang dioptimalkan untuk model berbasis pohon seperti *Random Forest*. *TreeExplainer* dipilih karena kompatibel langsung dengan Random Forest tanpa memerlukan data *background*, serta efisien secara komputasi sehingga dapat dijalankan per permintaan dalam aplikasi web. Objek *TreeExplainer* dibangun sekali saat aplikasi dimulai dari model yang tersimpan, lalu digunakan ulang untuk setiap permintaan interpretabilitas tanpa perlu membangunnya dari awal.

Untuk setiap kata sandi yang dianalisis, nilai SHAP dihitung untuk seluruh 25 fitur dan mencerminkan besarnya kontribusi masing-masing fitur terhadap prediksi model. Nilai SHAP positif menunjukkan bahwa fitur tersebut mendorong prediksi ke arah kelas yang diklasifikasikan, sedangkan nilai SHAP negatif menunjukkan bahwa fitur tersebut menentang prediksi tersebut. Enam fitur dengan kontribusi absolut terbesar dipilih dan ditampilkan sebagai representasi penjelasan yang paling relevan bagi pengguna.

Pada antarmuka pengguna, hasil interpretabilitas SHAP ditampilkan melalui komponen *ShapExplainer* yang dapat diakses dengan menekan tombol "Lihat Penjelasan Fitur". Komponen ini menampilkan dua bagian: probabilitas klasifikasi untuk ketiga kelas dalam bentuk *progress bar*, serta kontribusi enam fitur teratas yang divisualisasikan sebagai *diverging bar chart* horizontal. Bar berwarna biru menunjukkan fitur yang mendukung prediksi, sedangkan bar berwarna merah menunjukkan fitur yang menentang prediksi. Tampilan komponen interpretabilitas SHAP beserta contoh analisis akan disajikan pada Bab 4.

BAB IV
HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil penelitian beserta pembahasannya berdasarkan tahapan yang telah dirancang pada Bab III. Seluruh eksperimen dilakukan menggunakan *dataset* yang telah melalui proses pelabelan ulang menggunakan *zxcvbn*. Penyajian hasil disusun mengikuti urutan pelaksanaan penelitian, dimulai dari perbandingan performa model pada data validasi, pemilihan model terbaik, optimasi *threshold*, evaluasi pada data uji, interpretasi model menggunakan SHAP, implementasi *prototype*, hingga pembahasan hasil penelitian.

4.1 Hasil Perbandingan Model

Sesuai tahapan pemodelan yang telah dijelaskan pada Bab III, langkah pertama dalam penelitian ini adalah membandingkan performa empat algoritma klasifikasi, yaitu *Logistic Regression*, *Random Forest*, *Support Vector Machine* (SVM) Linear, dan XGBoost. Setiap algoritma dievaluasi pada dua skenario penanganan ketidakseimbangan kelas, yaitu *without balancing* dan *with balancing* menggunakan *random undersampling*. Dengan demikian, diperoleh delapan model kandidat yang selanjutnya dibandingkan pada tahap pemilihan model. Setiap model kandidat merupakan hasil pelatihan menggunakan kombinasi *hyperparameter* terbaik yang diperoleh melalui proses optimasi *RandomizedSearchCV* sebagaimana dijelaskan pada Subbab 3.4.3.

Seluruh proses perbandingan dilakukan menggunakan data validasi. Data uji tidak digunakan pada tahap ini karena disimpan untuk evaluasi akhir, sesuai dengan prosedur yang telah dijelaskan pada Bab III. Langkah ini dilakukan agar hasil evaluasi akhir tetap objektif dan tidak dipengaruhi oleh proses pemilihan model. Perbandingan performa menggunakan *F1-Score makro* sebagai metrik utama karena distribusi kelas pada *dataset* tidak seimbang. Hasil pengujian kedelapan model kandidat tersebut pada data validasi disajikan pada Tabel 4.1.

Tabel 4.1 Perbandingan performa delapan model kandidat pada data validasi

Algoritma	Skenario	F1-Score Makro	Akurasi	Skor Cross-validation	Waktu latih (detik)
Random Forest	*Without Balancing*	**0,8215**	**0,8515**	**0,8214**	706,05
XGBoost	*Without Balancing*	0,8205	0,8505	0,8184	176,55

Algoritma	Skenario	F1-Score Makro	Akurasi	Skor Cross-validation	Waktu latih (detik)
Random Forest	*With Balancing*	0,8091	0,8344	0,8082	208,89
XGBoost	*With Balancing*	0,8082	0,8338	0,8035	122,60
Logistic Regression	*Without Balancing*	0,7779	0,8145	0,7816	73,28
Logistic Regression	*With Balancing*	0,7707	0,7989	0,7714	41,64
SVM *Linear*	*Without Balancing*	0,7671	0,8083	0,7736	12.008,69
SVM *Linear*	*With Balancing*	0,7649	0,7894	0,7688	4.495,58

Berdasarkan Tabel 4.1, terdapat tiga hal yang dapat diamati. Pertama, pada keempat algoritma, skenario *without balancing* secara konsisten menghasilkan *F1-Score makro* yang sedikit lebih tinggi dibandingkan skenario *with balancing*. Kedua, *Random Forest* dan XGBoost memberikan performa terbaik dengan selisih yang sangat kecil, yaitu 0,8215 dan 0,8205. Sementara itu, *Logistic Regression* dan SVM *Linear* memperoleh *F1-Score makro* pada kisaran 0,76 hingga 0,78. Ketiga, waktu pelatihan setiap algoritma berbeda cukup jauh. SVM *Linear* membutuhkan waktu pelatihan paling lama, yaitu sekitar 12.009 detik pada skenario *without balancing*. Sebagai perbandingan, *Random Forest* membutuhkan sekitar 706 detik, sedangkan *XGBoost* hanya sekitar 177 detik.

Selain itu, nilai *cross-validation* setiap model mendekati nilai *F1-Score* makro pada data validasi. Hal ini menunjukkan bahwa performa model relatif konsisten dan tidak terdapat indikasi *overfitting*. Berdasarkan hasil tersebut, *Random Forest* pada skenario *without balancing* memperoleh *F1-Score makro* tertinggi sehingga dipilih sebagai model terbaik untuk tahap berikutnya. Pembahasan mengenai penyebab skenario *without balancing* memberikan hasil yang lebih baik dibandingkan *with balancing*, serta alasan pemilihan *Random Forest* dibandingkan *XGBoost*, dibahas lebih lanjut pada Subbab 4.7.

4.2 Pemilihan Model Terbaik

Model terbaik dipilih berdasarkan hasil perbandingan pada Subbab 4.1 menggunakan dua kriteria yang telah ditetapkan pada Bab III, yaitu nilai *F1-Score* makro tertinggi pada data validasi dan konsistensi hasil *cross-validation*. *Random Forest* pada skenario *without balancing* memenuhi kedua kriteria tersebut. Model ini memperoleh *F1-Score* makro sebesar

0,8215, yang merupakan nilai tertinggi di antara seluruh model yang diuji. Selain itu, nilai *cross-validation* sebesar 0,8214 sangat mendekati nilai *F1-Score* makro pada data validasi, sehingga menunjukkan bahwa performa model cukup konsisten.

Berdasarkan hasil tersebut, *Random Forest without balancing* dipilih sebagai model final dan digunakan pada seluruh tahapan penelitian berikutnya. Konfigurasi *hyperparameter* terbaik yang diperoleh melalui proses *tuning* menggunakan *RandomizedSearchCV* disajikan pada Tabel 4.2.

Tabel 4.2 Konfigurasi *hyperparameter* model final *Random Forest*

Hyperparameter	Nilai	Keterangan
n_estimators	300	Jumlah pohon dalam *ensemble*
max_depth	20	Kedalaman maksimum setiap pohon
min_samples_leaf	5	Jumlah minimum sampel pada setiap daun
max_features	0,5	Proporsi fitur yang dipertimbangkan pada setiap *split*
criterion	Gini	Fungsi pengukuran kualitas *split*
bootstrap	True	Pengambilan sampel dengan pengembalian
random_state	42	Nilai *seed* untuk reprodusibilitas

Meskipun *Random Forest* memperoleh nilai *F1-Score* makro tertinggi, selisih terhadap XGBoost hanya sebesar 0,0010. Mengingat perbedaan tersebut relatif kecil, dilakukan uji signifikansi statistik menggunakan *paired bootstrap* dengan 2.000 replikasi terhadap selisih (*difference*) *F1-Score* makro kedua model. Pengujian dilakukan melalui pengambilan sampel ulang secara berpasangan pada *indeks* data *validation set* yang sama sehingga variasi yang diukur berasal dari perbedaan model, bukan dari perbedaan sampel yang dievaluasi.

Hasil pengujian menunjukkan selang kepercayaan 95% terhadap selisih *F1-Score* makro berada pada rentang −0,0001 hingga +0,0022 dengan nilai *p* sebesar 0,086. Pada taraf signifikansi 5%, hasil tersebut menunjukkan bahwa tidak terdapat bukti yang cukup untuk menyimpulkan bahwa perbedaan *F1-Score* makro antara *Random Forest* dan XGBoost signifikan secara statistik. Dengan demikian, keunggulan *Random Forest* pada *validation set* belum dapat dinyatakan sebagai keunggulan yang meyakinkan secara statistik.

Selain uji signifikansi tersebut, dilakukan pula analisis sensitivitas terhadap formulasi *F1-Score* makro yang digunakan dengan menghitung kembali performa kedua model

menggunakan formulasi *F1 of averages* sebagai pembanding terhadap formulasi *averaged F1*. *Random Forest* memperoleh *Macro-Precision* sebesar 0,8494 dan *Macro-Recall* sebesar 0,8029 sehingga menghasilkan nilai *F1 of averages* sebesar 0,8255. Sementara itu, XGBoost memperoleh *Macro-Precision* sebesar 0,8477 dan *Macro-Recall* sebesar 0,8022 sehingga menghasilkan nilai *F1 of averages* sebesar 0,8243. Selisih sebesar 0,0012 tersebut sedikit lebih besar dibandingkan selisih pada formulasi *averaged F1* sebesar 0,0010. Meskipun demikian, kedua formulasi menghasilkan peringkat model yang sama, yaitu *Random Forest* tetap berada pada posisi terbaik. Hasil ini menunjukkan bahwa keputusan pemilihan model tetap konsisten meskipun formulasi *F1-Score* makro yang digunakan diubah.

Berdasarkan hasil uji signifikansi tersebut, tidak terdapat bukti yang cukup untuk membedakan kedua model berdasarkan *F1-Score* makro. Oleh karena itu, sesuai prosedur pemilihan model yang telah ditetapkan pada Subbab 3.5.2, penetapan model final dilakukan menggunakan kriteria sekunder berupa konsistensi performa antara *validation set* dan *cross-validation*. *Random Forest* memperoleh nilai *cross-validation* sebesar 0,8214, sehingga selisih terhadap *F1-Score* makro pada *validation set* hanya sebesar 0,0001 (0,8215 dibandingkan 0,8214). Sebaliknya, XGBoost memperoleh nilai *cross-validation* sebesar 0,8184 dengan selisih sebesar 0,0021 terhadap *F1-Score* makro pada *validation set* (0,8205 dibandingkan 0,8184). Hasil tersebut menunjukkan bahwa *Random Forest* memiliki performa yang lebih konsisten antara estimasi performa selama proses pelatihan dan evaluasi pada data validasi. Atas dasar tersebut, *Random Forest* pada skenario *without balancing* ditetapkan sebagai model final dalam penelitian ini. Waktu pelatihan tidak dijadikan sebagai pertimbangan pemilihan model karena sistem yang dikembangkan hanya melakukan klasifikasi terhadap satu kata sandi pada setiap permintaan. Pembahasan lebih lanjut mengenai pertimbangan pemilihan model disajikan pada Subbab 4.7.

4.3 Hasil Optimasi *Threshold*

Setelah model final ditetapkan, tahap berikutnya adalah melakukan optimasi *threshold* untuk memenuhi spesifikasi keamanan yang telah ditetapkan pada Bab III, yaitu proporsi kata sandi *weak* yang salah diklasifikasikan sebagai *strong* tidak lebih dari 5 persen. Optimasi ini dilakukan karena prediksi menggunakan metode *argmax* standar masih menghasilkan proporsi kesalahan *weak→strong* sebesar 15,04 persen pada data uji. Nilai tersebut sekitar tiga kali lebih besar dibandingkan batas yang ditetapkan. Sesuai dengan penjelasan pada Subbab 3.4, prediksi kelas ditentukan berdasarkan hasil pembagian probabilitas setiap kelas dengan nilai *threshold*-

nya. Semakin kecil nilai *threshold* suatu kelas, semakin besar peluang kelas tersebut dipilih sebagai hasil prediksi. Pencarian kombinasi *threshold* dilakukan menggunakan data validasi. Kombinasi terbaik dipilih berdasarkan *F1-Score makro* tertinggi dari seluruh kombinasi yang tetap memenuhi batas keamanan *weak→strong* sebesar 5 persen.

4.3.1 Hasil Optimasi *Threshold* pada Model Final

Hasil optimasi *threshold* untuk model final *Random Forest* pada skenario *without balancing* disajikan pada Tabel 4.3.

Tabel 4.3 Hasil optimasi *threshold* model final pada data validasi

Model	F1-Score Makro Sebelum	F1-Score Makro Sesudah	Selisih	weak→strong	Recall weak	Threshold
Random Forest	0,8215	0,8111	-0,0104	4,88%	0,7530	[0,25; 0,25; 0,55]

Berdasarkan Tabel 4.3, penerapan *threshold* [0,25; 0,25; 0,55] berhasil menurunkan proporsi kesalahan klasifikasi *weak→strong* menjadi 4,88 persen pada data validasi. Nilai tersebut telah memenuhi batas keamanan yang ditetapkan, yaitu maksimal 5 persen. Selain itu, *recall* untuk kelas *weak* mencapai 0,7530, sehingga juga memenuhi target minimum sebesar 0,70 yang telah ditentukan pada Bab III. Penerapan *threshold* menyebabkan *F1-Score* makro pada data validasi turun dari 0,8215 menjadi 0,8111, atau berkurang sebesar 0,0104. Penurunan ini terjadi karena model menjadi lebih berhati-hati dalam memberikan prediksi *strong*. Dengan kata lain, sebagian performa dikorbankan agar kesalahan mengklasifikasikan kata sandi *weak* sebagai *strong* dapat dikurangi.

4.3.2 Analisis Sensitivitas Batas Keamanan

Untuk melihat pengaruh perubahan batas keamanan terhadap performa model, proses optimasi *threshold* dilakukan kembali menggunakan beberapa batas *weak→strong* yang berbeda pada data validasi. Hasil terbaik yang diperoleh pada setiap batas keamanan disajikan pada Tabel 4.4.

Tabel 4.4 *F1-Score* makro terbaik yang dapat dicapai pada berbagai batas keamanan

Batas *weak→strong*	**F1-Score Makro Terbaik**
3%	0,8012
5%	0,8111
8%	0,8171
10%	0,8205
12%	0,8213
15%	0,8222
20%	0,8222

BerdasarkanTabel 4.4, terlihat bahwa semakin ketat batas keamanan yang digunakan, semakin rendah *F1-Score makro* terbaik yang dapat diperoleh. Sebagai contoh, ketika batas keamanan diperketat dari 5 persen menjadi 3 persen, *F1-Score* makro turun dari 0,8111 menjadi 0,8012. Sebaliknya, ketika batas keamanan dilonggarkan menjadi 8 persen, *F1-Score* makro meningkat menjadi 0,8171. Pada batas 15 persen atau lebih, peningkatan *F1-Score* makro mulai sangat kecil dan cenderung stabil di sekitar 0,8222. Hal ini menunjukkan bahwa pelonggaran batas keamanan setelah titik tersebut tidak lagi memberikan peningkatan performa yang berarti.

Hasil ini menunjukkan adanya hubungan antara keamanan dan performa model. Semakin ketat batas keamanan yang diterapkan, semakin besar penurunan performa yang harus diterima. Oleh karena itu, penelitian ini menggunakan batas 5 persen sebagai titik kompromi antara keamanan dan performa. Pembahasan mengenai alasan pemilihan batas tersebut dijelaskan lebih lanjut pada Subbab 4.7.

4.4 Evaluasi Final pada Data Uji

Model final *Random Forest* dengan *threshold* [0,25; 0,25; 0,55] selanjutnya dievaluasi menggunakan data uji yang terdiri atas 133.928 sampel. Sesuai dengan prosedur yang telah dijelaskan pada Bab III, data uji tidak digunakan pada tahap pelatihan, pemilihan model, maupun optimasi *threshold*. Oleh karena itu, hasil pada subbab ini memberikan gambaran performa model terhadap data yang belum pernah dilihat sebelumnya. Secara keseluruhan, model memperoleh akurasi sebesar 0,8344 dan *F1-Score makro* sebesar 0,8121 pada data uji.

4.4.1 Performa Keseluruhan dan Per Kelas

Selain mengevaluasi performa model secara keseluruhan, penelitian ini juga mengevaluasi performa pada setiap kelas untuk mengetahui kemampuan model dalam mengklasifikasikan kategori *weak*, *medium*, dan *strong*. Evaluasi per kelas memberikan informasi yang lebih rinci mengenai kekuatan dan kelemahan model pada masing-masing kategori. Rincian performa model per kelas pada data uji disajikan pada

Tabel 4.5.

Tabel 4.5 Performa model final per kelas pada data uji

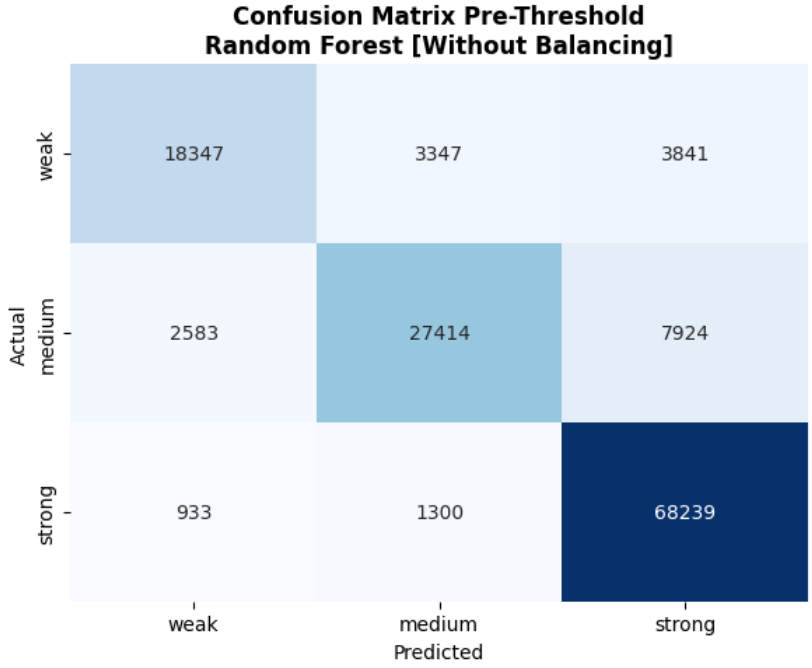
Kelas	**Precision**	**Recall**	**F1-Score**	**Jumlah Sampel**
weak	0,7827	0,7601	0,7712	25.535
medium	0,7252	0,8175	0,7686	37.921
strong	0,9240	0,8704	0,8964	70.472
Rata-rata makro	0,8106	0,8160	0,8121	133.928

Berdasarkan

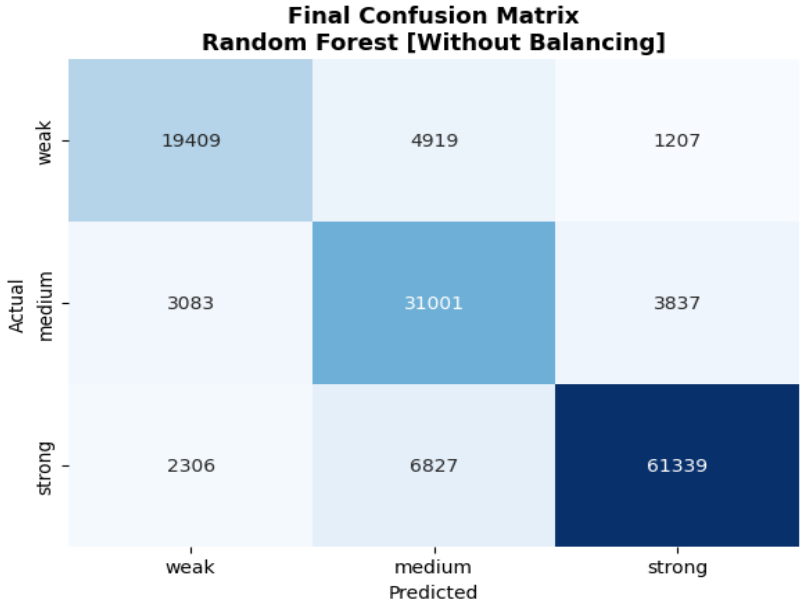
Tabel 4.5, kelas *strong* memiliki performa terbaik dengan *F1-Score* sebesar 0,8964. Sementara itu, kelas *weak* dan *medium* memperoleh *F1-Score* masing-masing sebesar 0,7712 dan 0,7686. Perbedaan ini menunjukkan bahwa model lebih baik dalam mengidentifikasi kata sandi yang termasuk kategori *strong*. Sebaliknya, pemisahan antara kelas *weak* dan *medium* lebih sulit karena karakteristik kedua kelas tersebut lebih mirip. Selain itu, *recall* kelas *weak* sebesar 0,7601 menunjukkan bahwa sekitar 76 persen kata sandi lemah berhasil dikenali dengan benar. Hasil ini penting karena kemampuan mendeteksi kata sandi lemah menjadi salah satu fokus utama dalam penelitian ini.

4.4.2 *Confusion Matrix*

Selain menggunakan metrik evaluasi per kelas, pola kesalahan klasifikasi juga dianalisis melalui *confusion matrix*. Visualisasi ini memberikan gambaran mengenai distribusi prediksi benar maupun salah pada setiap kelas sehingga jenis kesalahan yang paling dominan dapat diidentifikasi. *Confusion matrix* hasil prediksi model sebelum dan sesudah dilakukan optimasi *threshold* pada data uji disajikan pada Gambar 4.1 dan Gambar 4.2.



Gambar 4.1 *Confusion matrix* sebelum optimasi *threshold* model final pada data uji



Gambar 4.2 *Confusion matrix* setelah optimasi *threshold* model final pada data uji

Berdasarkan Gambar 4.2, perhatian utama diberikan pada baris kelas *weak*. Dari 25.535 kata sandi yang sebenarnya termasuk kelas *weak*, sebanyak 19.409 (76,01 persen) berhasil diklasifikasikan dengan benar. Sebanyak 4.919 (19,26 persen) diklasifikasikan sebagai *medium*, sedangkan 1.207 (4,73 persen) diklasifikasikan sebagai *strong*. Hasil tersebut menunjukkan bahwa kesalahan yang paling berisiko, yaitu mengklasifikasikan kata sandi *weak* sebagai *strong*, berhasil ditekan hingga berada di bawah batas 5 persen. Sebagian besar

kesalahan justru bergeser menjadi klasifikasi *weak* ke *medium*. Kesalahan seperti ini masih lebih aman karena kata sandi tetap tidak dinilai sebagai *strong*, sehingga pengguna masih memperoleh peringatan bahwa kata sandinya belum cukup kuat. Pembahasan lebih lanjut mengenai pengaruh penerapan *threshold* terhadap pola kesalahan ini dijelaskan pada Subbab 4.7.

4.4.3 Verifikasi Pemenuhan Target

Hasil evaluasi pada data uji kemudian dibandingkan dengan kriteria keberhasilan penelitian dan spesifikasi keamanan yang telah ditetapkan pada Bab III. Hasil verifikasi tersebut disajikan pada Tabel 4.6.

Tabel 4.6 Verifikasi pemenuhan target pada data uji

Aspek	**Jenis**	**Target**	**Hasil**	**Status**
Akurasi	Kriteria keberhasilan	≥ 80%	83,44%	Lulus
F1-Score makro	Kriteria keberhasilan	≥ 0,75	0,8121	Lulus
Recall weak	Kriteria keberhasilan	≥ 0,70	0,7601	Lulus
Proporsi misklasifikasi *weak→strong*	Spesifikasi keamanan	≤ 5%	4,73%	Lulus

Berdasarkan Tabel 4.6, seluruh target penelitian berhasil dipenuhi. Akurasi, *F1-Score* makro, dan *recall* kelas *weak* telah melampaui nilai minimum yang ditetapkan. Selain itu, proporsi kesalahan klasifikasi *weak→strong* sebesar 4,73 persen juga berada di bawah batas maksimal 5 persen.

Untuk melihat pengaruh penerapan *threshold*, model juga dievaluasi menggunakan prediksi *argmax* standar tanpa optimasi *threshold*. Pada kondisi tersebut, proporsi kesalahan *weak→strong* mencapai 15,04 persen atau 3.841 dari 25.535 kata sandi *weak*. Nilai ini jauh melebihi batas keamanan yang telah ditetapkan, meskipun akurasi dan *F1-Score makro* terlihat lebih tinggi, yaitu masing-masing sebesar 0,8512 dan 0,8215.

Setelah *threshold* [0,25; 0,25; 0,55] diterapkan, proporsi kesalahan *weak→strong* turun menjadi 4,73 persen atau 1.207 kasus. Hasil ini menunjukkan bahwa optimasi *threshold* berperan penting dalam memenuhi spesifikasi keamanan penelitian. Meskipun terjadi sedikit penurunan pada performa keseluruhan, jumlah kata sandi lemah yang salah diklasifikasikan sebagai *strong* dapat dikurangi secara signifikan.

4.5 Interpretasi Model

Agar keputusan model lebih mudah dipahami, dilakukan interpretasi dari dua sudut pandang. Pertama, digunakan *feature importance* bawaan *Random Forest* untuk mengetahui fitur yang paling berpengaruh secara umum terhadap proses klasifikasi. Kedua, digunakan SHAP untuk menjelaskan kontribusi setiap fitur terhadap hasil prediksi pada masing-masing kata sandi. Sebagaimana telah dijelaskan pada Bab II, perhitungan SHAP pada penelitian ini menggunakan *TreeExplainer* yang sesuai untuk model berbasis pohon seperti *Random Forest*.

4.5.1 *Feature Importance* Global

Untuk memperoleh gambaran awal mengenai kontribusi masing-masing fitur terhadap proses klasifikasi, dilakukan analisis menggunakan *feature importance* pada model *Random Forest*. Analisis ini digunakan untuk mengidentifikasi fitur-fitur yang memberikan kontribusi terbesar terhadap model. Sepuluh fitur dengan nilai *feature importance* tertinggi pada model final disajikan pada Tabel 4.7.

Tabel 4.7 Sepuluh fitur paling berpengaruh pada model final

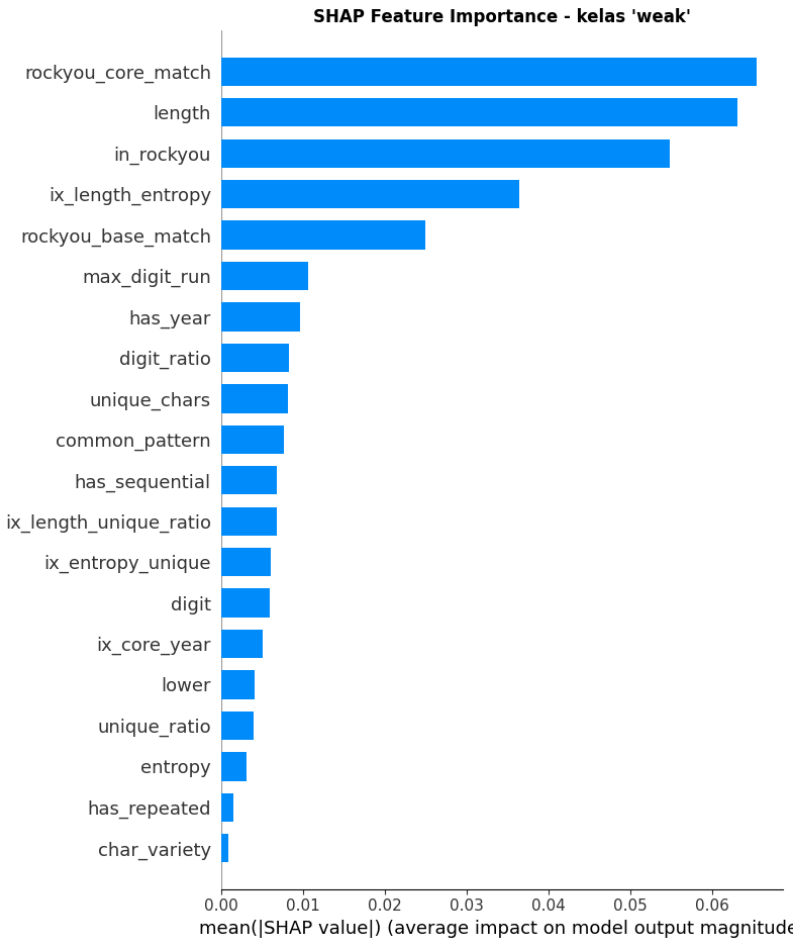
Peringkat	Fitur	Importance
1	length	0,3581
2	ix_length_entropy	0,1918
3	in_rockyou	0,0978
4	rockyou_core_match	0,0822
5	rockyou_base_match	0,0435
6	unique_chars	0,0344
7	ix_entropy_unique	0,0296
8	ix_length_unique_ratio	0,0283
9	digit_ratio	0,0229
10	max_digit_run	0,0197

Berdasarkan Tabel 4.7, fitur *length* merupakan fitur yang paling berpengaruh dengan nilai *importance* sebesar 0,3581. Fitur ini diikuti oleh *ix_length_entropy* dengan nilai 0,1918. Selain kedua fitur tersebut, beberapa fitur yang berkaitan dengan kemunculan kata sandi pada basis data RockYou juga memiliki pengaruh yang cukup besar. Fitur *in_rockyou* berada pada peringkat ketiga dengan nilai 0,0978, sedangkan *rockyou_core_match* berada pada peringkat keempat dengan nilai 0,0822. Dapat diperhatikan juga bahwa *rockyou_core_match* merupakan

salah satu fitur yang ditambahkan pada tahap rekayasa fitur. Nilai *importance* yang cukup tinggi menunjukkan bahwa fitur tersebut memberikan kontribusi dalam proses klasifikasi. Nilai *feature importance* ini diperoleh dari mekanisme internal *Random Forest*, yaitu berdasarkan seberapa besar suatu fitur membantu mengurangi ketidakmurnian (*impurity*) pada proses pembentukan pohon keputusan. Untuk memperoleh gambaran yang lebih rinci mengenai pengaruh setiap fitur pada masing-masing kelas, analisis selanjutnya dilakukan menggunakan SHAP pada Subbab 4.5.2.

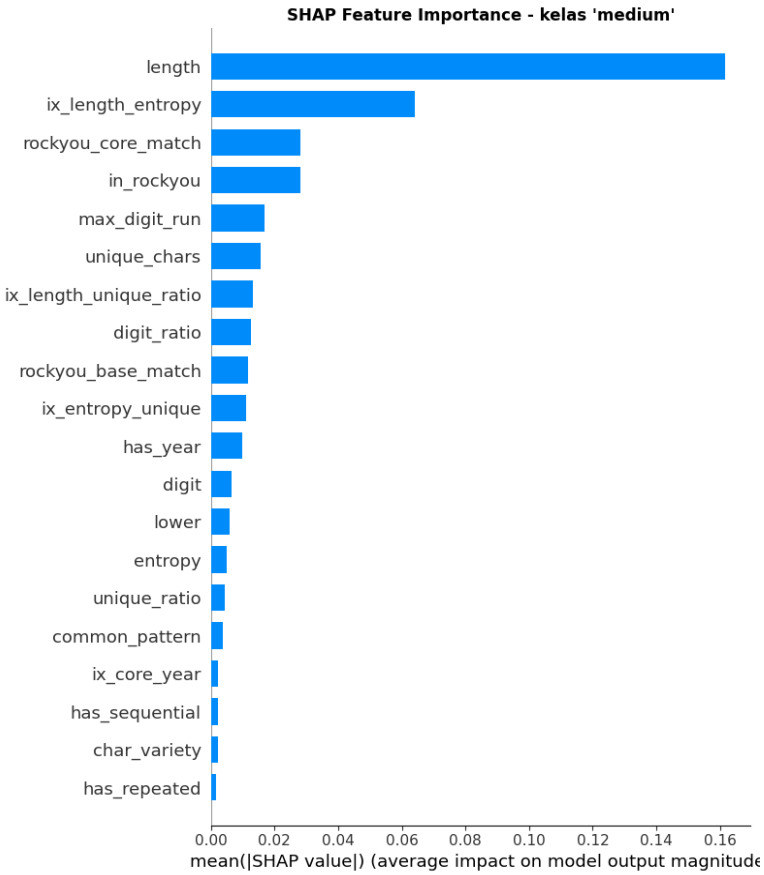
4.5.2 Interpretasi SHAP per Kelas

Selain menggunakan *feature importance*, penelitian ini juga menggunakan SHAP untuk melihat pengaruh setiap fitur terhadap masing-masing kelas. Ringkasan SHAP dihitung menggunakan rata-rata nilai absolut SHAP (*mean absolute SHAP value*) pada seluruh data uji. Nilai ini menunjukkan besarnya pengaruh suatu fitur terhadap hasil prediksi, tetapi tidak menunjukkan apakah pengaruh tersebut meningkatkan atau menurunkan peluang suatu kelas. Ringkasan SHAP untuk kelas *weak*, *medium*, dan *strong* ditunjukkan pada Gambar 4.3 sampai Gambar 4.5.



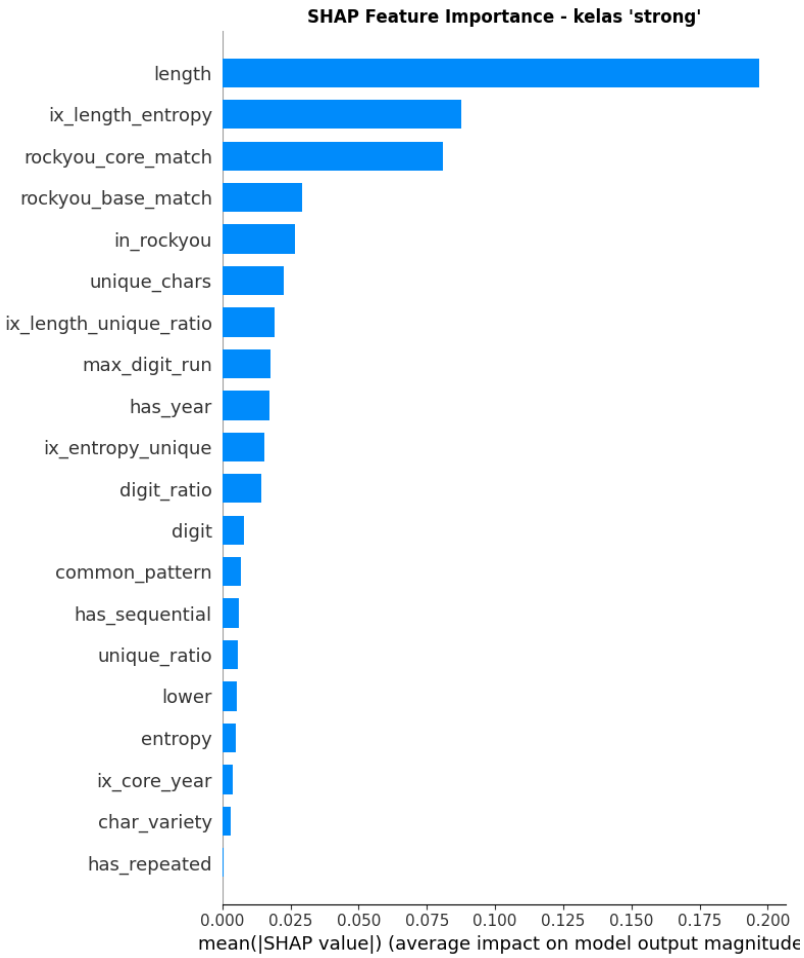
Gambar 4.3 Ringkasan SHAP global kata sandi *weak*

Berdasarkan Gambar 4.3, fitur yang berkaitan dengan kemunculan kata sandi pada basis data kata sandi yang bocor, yaitu *rockyou_core_match* dan *in_rockyou*, termasuk fitur dengan pengaruh terbesar terhadap prediksi kelas *weak*. Selain itu, fitur *length* juga memiliki pengaruh yang besar. Hasil ini menunjukkan bahwa keberadaan kata sandi dalam basis data kata sandi yang bocor serta panjang kata sandi menjadi faktor penting ketika model mengklasifikasikan suatu kata sandi sebagai *weak*.



Gambar 4.4 Ringkasan SHAP global kata sandi *medium*

Pada Gambar 4.4, fitur *length* menjadi fitur yang paling berpengaruh terhadap prediksi kelas *medium*. Sementara itu, fitur yang berkaitan dengan basis data RockYou, seperti *rockyou_core_match* dan *in_rockyou*, juga masih memberikan pengaruh yang cukup besar. Hal ini menunjukkan bahwa klasifikasi pada kelas *medium* dipengaruhi oleh kombinasi beberapa fitur. Kata sandi pada kelas ini belum cukup lemah untuk masuk ke kelas *weak*, tetapi juga belum memenuhi karakteristik kata sandi yang termasuk kategori *strong*.



Gambar 4.5 Ringkasan SHAP global kata sandi *strong*

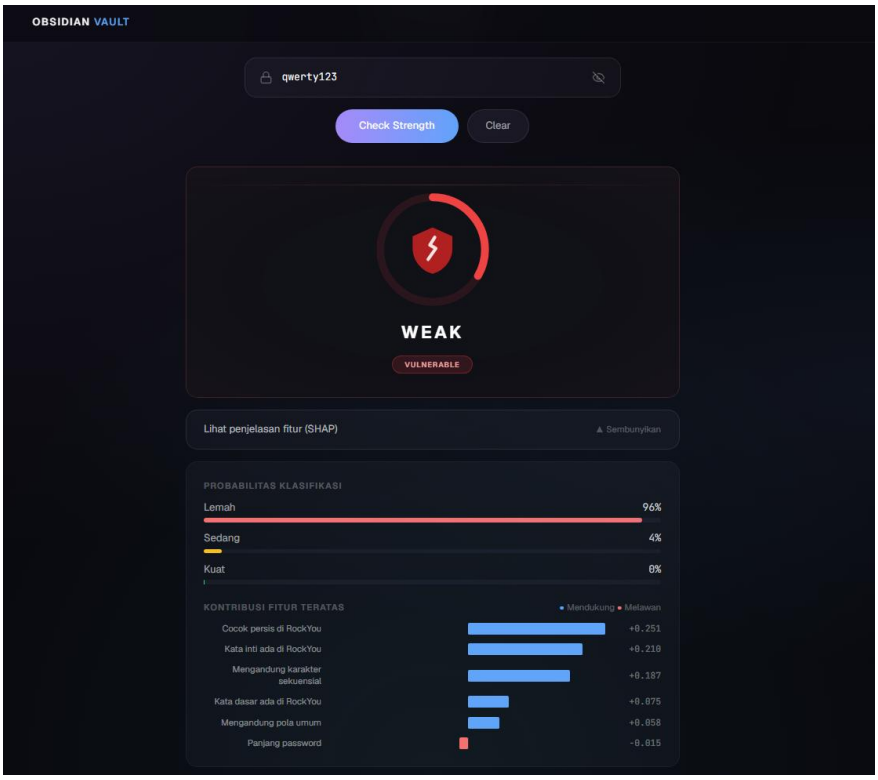
Berdasarkan Gambar 4.5, fitur *length* dan *ix_length_entropy* merupakan dua fitur yang paling berpengaruh terhadap prediksi kelas *strong*. Selain itu, fitur yang berkaitan dengan basis data *RockYou* dan keragaman karakter juga masih memberikan kontribusi terhadap hasil klasifikasi. Jika dibandingkan dengan kelas *weak*, terlihat bahwa urutan fitur yang paling berpengaruh berbeda. Pada kelas *weak*, fitur yang berkaitan dengan basis data kata sandi yang bocor memiliki pengaruh yang lebih besar, sedangkan pada kelas *strong*, pengaruh terbesar berasal dari panjang kata sandi beserta fitur turunannya.

Secara keseluruhan, hasil SHAP mendukung hasil *feature importance* pada Subbab 4.5.1. Perbedaannya, SHAP memberikan informasi yang lebih rinci mengenai pengaruh fitur pada setiap kelas, sedangkan *feature importance* hanya menunjukkan pengaruh fitur terhadap model secara umum. Arah pengaruh setiap fitur terhadap prediksi suatu kata sandi dijelaskan lebih lanjut melalui contoh pada antarmuka *prototype* di Subbab 4.6.

4.6 Hasil Implementasi *Prototype*

Model final beserta mekanisme interpretasinya diimplementasikan ke dalam sebuah *prototype* sistem berbasis web. Implementasi ini dilakukan sebagai bentuk penerapan model yang telah dikembangkan sekaligus untuk memenuhi tujuan penelitian. Rancangan arsitektur sistem, daftar *endpoint*, dan desain antarmuka telah dijelaskan pada Bab III. Oleh karena itu, subbab ini berfokus pada hasil implementasi sistem, yaitu proses klasifikasi kata sandi mulai dari *input* pengguna hingga tampilan hasil klasifikasi beserta penjelasannya.

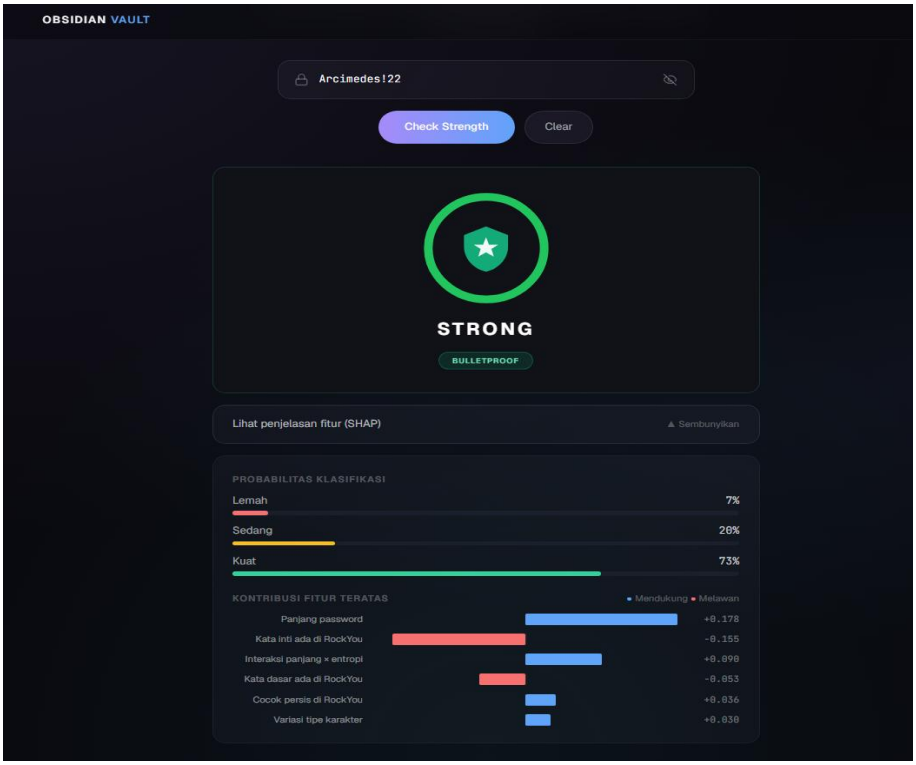
Untuk menunjukkan bahwa sistem dapat digunakan pada berbagai kondisi, dilakukan pengujian menggunakan tiga contoh kata sandi yang mewakili kelas *weak*, *medium*, dan *strong*. Hasil pengujian tersebut ditampilkan pada Gambar 4.6 sampai Gambar 4.8. Pada setiap contoh, sistem menampilkan hasil klasifikasi serta penjelasan SHAP yang menunjukkan enam fitur dengan pengaruh terbesar terhadap hasil prediksi. Penjelasan tersebut juga memperlihatkan apakah suatu fitur mendukung atau mengurangi peluang suatu kelas dipilih oleh model.



Gambar 4.6 Demonstrasi klasifikasi dan penjelasan untuk kata sandi *weak* pada antarmuka *prototype*



Gambar 4.7 Demonstrasi klasifikasi dan penjelasan untuk kata sandi *medium* pada antarmuka *prototype*



Gambar 4.8 Demonstrasi klasifikasi dan penjelasan untuk kata sandi *strong* pada antarmuka *prototype*

Secara keseluruhan, hasil implementasi menunjukkan bahwa *prototype* berhasil mengintegrasikan model klasifikasi dan mekanisme interpretasi SHAP ke dalam satu sistem berbasis web. Pengguna tidak hanya memperoleh hasil klasifikasi kekuatan kata sandi, tetapi juga penjelasan mengenai faktor-faktor yang memengaruhi hasil prediksi tersebut. Dengan demikian, model yang dikembangkan pada penelitian ini telah berhasil diterapkan ke dalam sebuah sistem yang dapat digunakan untuk melakukan klasifikasi kekuatan kata sandi sekaligus menampilkan penjelasan hasil prediksinya.

4.7 Pembahasan

Subbab ini membahas hasil penelitian yang telah diperoleh pada subbab sebelumnya dan menghubungkannya dengan penelitian terdahulu yang dibahas pada Bab II. Selain itu, pada bagian ini juga dijelaskan beberapa keterbatasan penelitian yang perlu diperhatikan. Pembahasan disusun berdasarkan beberapa tema, mulai dari kualitas data, performa model, penerapan *threshold*, interpretasi model, hingga keterbatasan penelitian.

4.7.1 Strategi *Relabeling* dan Kualitas Data

Pelabelan ulang menggunakan *zxcvbn* mengubah distribusi kelas pada *dataset* secara cukup besar. Sebelum dilakukan *relabeling*, kelas *medium* mendominasi *dataset* dengan proporsi sekitar 74,2 persen. Setelah pelabelan ulang, distribusi kelas menjadi lebih seimbang dengan kelas *strong* sebagai kelas yang memiliki jumlah sampel terbanyak.

Perubahan tersebut terjadi karena label asli pada *dataset* lebih banyak ditentukan berdasarkan panjang kata sandi. Akibatnya, sebagian besar kata sandi dengan panjang antara 8 hingga 13 karakter langsung masuk ke dalam kelas *medium*, meskipun tingkat keamanannya belum tentu sama. Dengan menggunakan *zxcvbn*, penentuan label tidak hanya mempertimbangkan panjang kata sandi, tetapi juga pola, kemunculan pada basis data kata sandi yang bocor, serta karakteristik lain yang memengaruhi tingkat kekuatan kata sandi. Pendekatan ini mereplikasi praktik Pande (2026) yang menggunakan *zxcvbn* sebagai acuan pelabelan untuk mengurangi kemungkinan *label leakage*. Dengan cara ini, model tidak hanya mempelajari hubungan antara panjang kata sandi dan label, tetapi juga mempelajari karakteristik lain yang berhubungan dengan kekuatan kata sandi.

Berbeda dengan beberapa penelitian terdahulu yang masih menggunakan label bawaan *dataset*, penelitian ini melakukan pelabelan ulang terlebih dahulu sebelum proses pelatihan model. Langkah tersebut membuat proses pelatihan, evaluasi, dan interpretasi dilakukan

menggunakan label yang lebih sesuai dengan kondisi keamanan kata sandi. Apabila pelabelan ulang tidak dilakukan, model kemungkinan hanya akan mempelajari aturan panjang kata sandi seperti pada label asli. Akibatnya, model dapat memperoleh performa yang tinggi, tetapi belum tentu mampu menggambarkan kekuatan kata sandi secara lebih baik.

4.7.2 Performa Model dan Pengaruh *Balancing*

Hasil pengujian menunjukkan bahwa *Random Forest* dan XGBoost memberikan performa terbaik dibandingkan *Logistic Regression* dan *Support Vector Machine* (SVM) *Linear*. Perbedaan *F1-Score* makro antara *Random Forest* dan XGBoost juga sangat kecil, sehingga keduanya sama-sama memiliki kemampuan klasifikasi yang baik pada penelitian ini.

Performa kedua algoritma tersebut kemungkinan dipengaruhi oleh kemampuannya dalam menangkap hubungan yang lebih kompleks antarfitur. Berbeda dengan *Logistic Regression* dan SVM *Linear* yang membentuk batas keputusan secara linier, *Random Forest* dan XGBoost dapat mempelajari hubungan nonlinier, misalnya interaksi antara panjang kata sandi, entropi, dan fitur lain yang digunakan dalam penelitian ini. Hasil ini sejalan dengan beberapa penelitian terdahulu yang menunjukkan bahwa metode berbasis *ensemble* sering memberikan performa yang baik pada klasifikasi kekuatan kata sandi.

Selain membandingkan algoritma, penelitian ini juga mengevaluasi pengaruh penanganan ketidakseimbangan kelas menggunakan *random undersampling*. Berdasarkan hasil pengujian, penerapan *balancing* justru menurunkan *F1-Score makro* pada seluruh algoritma yang diuji. Penurunan tersebut diduga terjadi karena proses *random undersampling* mengurangi jumlah data pada kelas mayoritas. Akibatnya, sebagian informasi yang sebenarnya dapat digunakan selama proses pelatihan menjadi hilang. Selain itu, setelah dilakukan *relabeling* menggunakan *zxcvbn*, distribusi kelas pada *dataset* menjadi sedikit lebih seimbang dibandingkan label asli. Oleh karena itu, tingkat ketidakseimbangan yang tersisa tidak cukup besar sehingga penanganan menggunakan *random undersampling* tidak memberikan peningkatan performa.

Berdasarkan hasil tersebut, *Random Forest without balancing* dipilih sebagai model final karena memperoleh *F1-Score* makro tertinggi pada data validasi serta memiliki nilai *cross-validation* yang konsisten. Waktu pelatihan tidak dijadikan kriteria pemilihan karena proses pelatihan dilakukan satu kali secara *offline*, sehingga selisih waktu antaralgoritma tidak memengaruhi proses klasifikasi ketika sistem digunakan.

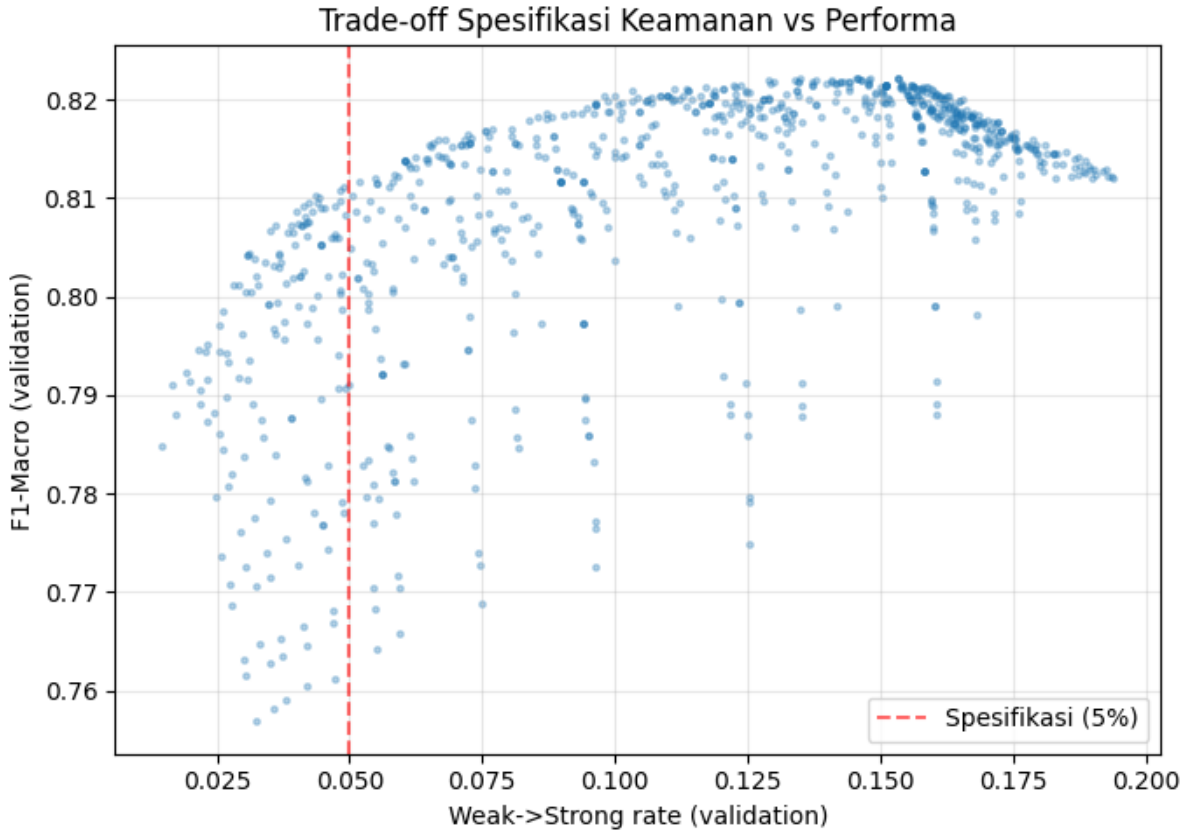
4.7.3 Pengaruh Threshold Asimetris terhadap Performa dan Keamanan

Salah satu hasil utama penelitian ini adalah penerapan *threshold* asimetris untuk memenuhi batas keamanan yang telah ditetapkan pada Bab III. Dengan menggunakan *threshold* [0,25; 0,25; 0,55], proporsi kesalahan klasifikasi kata sandi *weak* menjadi *strong* berhasil ditekan hingga 4,88 persen pada data validasi dan 4,73 persen pada data uji. Sebelum optimasi *threshold* dilakukan, proporsi kesalahan tersebut mencapai 15,04 persen pada data uji.

Penerapan *threshold* tersebut juga memengaruhi performa model. Pada data validasi, *F1-Score* makro turun dari 0,8215 menjadi 0,8111, atau berkurang sebesar 0,0104. Penurunan ini terjadi karena model dibuat lebih berhati-hati dalam memberikan prediksi *strong*. Akibatnya, jumlah kata sandi *weak* yang salah diklasifikasikan sebagai *strong* dapat dikurangi.

Hasil pada *confusion matrix* juga menunjukkan perubahan pola kesalahan setelah optimasi *threshold*. Ketika kesalahan *weak→strong* berkurang, jumlah kesalahan *weak→medium* meningkat menjadi 19,26 persen. Meskipun demikian, kesalahan tersebut masih lebih aman karena kata sandi tidak langsung dikategorikan sebagai *strong*. Jadi, pengguna tetap memperoleh informasi bahwa kata sandinya belum cukup kuat.

Analisis sensitivitas pada Subbab 4.3.2 juga menunjukkan bahwa terdapat hubungan antara batas keamanan dan performa model. Semakin ketat batas keamanan yang digunakan, semakin rendah *F1-Score* makro yang dapat dicapai. Dengan kata lain, peningkatan keamanan perlu diimbangi dengan penurunan performa klasifikasi. Sebaran hasil optimasi *threshold* pada berbagai batas keamanan ditunjukkan pada Gambar 4.9. Setiap titik pada gambar tersebut menunjukkan satu kombinasi *threshold* yang diuji. Sumbu horizontal menunjukkan proporsi kesalahan *weak→strong*, sedangkan sumbu vertikal menunjukkan *F1-Score makro*. Garis putus-putus merah menunjukkan batas keamanan sebesar 5 persen.



Gambar 4.9 Trade-off spesifikasi keamanan terhadap performa pada data validasi

Berdasarkan Gambar 4.9, terlihat bahwa semakin kecil proporsi kesalahan *weak→strong*, semakin rendah *F1-Score* makro yang dapat dicapai. Model final dipilih dari kombinasi *threshold* yang masih memenuhi batas keamanan 5 persen dan memiliki *F1-Score* makro tertinggi, yaitu 0,8111. Hasil ini berbeda dengan penelitian-penelitian terdahulu yang telah dibahas pada Bab II. Sebagian besar penelitian sebelumnya hanya melaporkan nilai akurasi atau *F1-Score* tanpa mengendalikan jenis kesalahan klasifikasi tertentu. Penelitian oleh Thai et al. (2024) memang menekankan pentingnya *recall* pada kelas *weak*, tetapi belum mengatur secara khusus proporsi kesalahan klasifikasi *weak* menjadi *strong*.

Pada penelitian ini, optimasi *threshold* digunakan untuk mengendalikan jenis kesalahan tersebut sehingga model tidak hanya menghasilkan performa klasifikasi yang baik, tetapi juga memenuhi batas keamanan yang telah ditentukan. Pendekatan ini memberikan fleksibilitas dalam memilih keseimbangan antara performa dan keamanan sesuai dengan kebutuhan sistem. Secara keseluruhan, hasil penelitian menunjukkan bahwa penggunaan *threshold* asimetris dapat menjadi salah satu cara untuk mengurangi kesalahan klasifikasi yang paling berisiko, yaitu ketika kata sandi *weak* dinilai sebagai *strong*, meskipun terdapat sedikit penurunan pada *F1-Score* makro.

4.7.4 Interpretasi Model dan Fitur Berpengaruh

Hasil interpretasi pada Subbab 4.5 menunjukkan bahwa *length* merupakan fitur yang memiliki pengaruh terbesar terhadap keputusan model dengan nilai *feature importance* sebesar 0,3581. Nilai ini lebih tinggi dibandingkan fitur lainnya sehingga menunjukkan bahwa panjang kata sandi masih menjadi salah satu faktor utama dalam proses klasifikasi. Meskipun demikian, model tidak hanya menggunakan panjang kata sandi sebagai dasar pengambilan keputusan. Hal ini terlihat dari nilai *feature importance* fitur lainnya yang secara keseluruhan masih memberikan kontribusi yang cukup besar. Fitur-fitur tersebut antara lain berkaitan dengan kemunculan kata sandi pada basis data kata sandi yang bocor, nilai entropi, serta hasil interaksi antarfitur.

Hasil SHAP juga memperlihatkan bahwa pengaruh setiap fitur berbeda pada masing-masing kelas. Pada kelas *weak*, fitur yang berkaitan dengan basis data RockYou, seperti *rockyou_core_match* dan *in_rockyou*, memberikan pengaruh yang lebih besar. Sebaliknya, pada kelas *strong*, fitur yang berkaitan dengan panjang kata sandi dan turunannya menjadi lebih dominan. Perbedaan ini menunjukkan bahwa model menggunakan kombinasi beberapa fitur untuk membedakan karakteristik setiap kelas.

Analisis terhadap kesalahan klasifikasi juga memberikan gambaran yang sama. Kata sandi *weak* yang masih salah diklasifikasikan sebagai *strong* umumnya memiliki panjang yang lebih tinggi dibandingkan kata sandi *weak* yang berhasil dikenali dengan benar. Namun, pada saat yang sama, kata sandi tersebut memiliki tingkat kemunculan yang lebih rendah pada basis data kata sandi yang bocor. Kondisi ini menunjukkan bahwa model tidak hanya mempertimbangkan panjang kata sandi, tetapi juga memperhatikan informasi lain yang tersedia pada fitur-fitur yang digunakan.

Hasil tersebut sejalan dengan pembahasan pada Bab II yang menjelaskan bahwa kekuatan kata sandi tidak hanya dipengaruhi oleh panjang karakter, tetapi juga oleh tingkat ketertebakan (*predictability*), pola yang digunakan, serta kemunculan pada basis data kata sandi yang telah bocor. Hasil ini juga sejalan dengan rekomendasi NIST SP 800-63B yang menekankan pentingnya panjang kata sandi dan pemeriksaan terhadap basis data kata sandi yang telah bocor sebagai bagian dari evaluasi kekuatan kata sandi. Dengan demikian, panjang kata sandi memang menjadi fitur yang paling berpengaruh dalam model. Namun, keputusan klasifikasi tetap ditentukan oleh kombinasi beberapa fitur sehingga model tidak hanya bergantung pada satu karakteristik saja.

4.7.5 Analisis Kesalahan Model dan Batas Kemampuannya

Meskipun penelitian ini menggunakan *zxcvbn* pada tahap pelabelan ulang data, model klasifikasi yang dikembangkan tidak mengimplementasikan mekanisme internal *zxcvbn* secara langsung. Model hanya memanfaatkan fitur-fitur struktural yang diekstraksi dari kata sandi untuk mempelajari hubungan dengan label yang dihasilkan *zxcvbn*. Oleh karena itu, masih dimungkinkan terdapat pola-pola yang dikenali oleh *zxcvbn* tetapi belum sepenuhnya direpresentasikan oleh fitur yang digunakan dalam penelitian ini.

Meskipun model telah memenuhi seluruh target penelitian, masih terdapat 1.207 kata sandi *weak* yang salah diklasifikasikan sebagai *strong*. Berdasarkan hasil analisis, kesalahan tersebut tidak terjadi secara acak, tetapi banyak ditemukan pada beberapa pola kata sandi yang memiliki karakteristik serupa. Pola pertama adalah penggunaan *leet substitution*, yaitu penggantian huruf dengan angka atau simbol yang bentuknya menyerupai huruf, seperti *hollyho0d* dan *blacksou1*. Pola kedua adalah penggunaan urutan tombol pada *keyboard*, misalnya *q1w2e3qwe*. Pola ketiga adalah kata sandi yang tidak terdapat pada kamus referensi yang digunakan dalam penelitian, seperti *stoneforge7* dan *akatsuki@@*.

Kesalahan pada ketiga pola tersebut menunjukkan bahwa fitur struktural yang digunakan dalam penelitian ini belum sepenuhnya merepresentasikan mekanisme penilaian yang digunakan oleh *zxcvbn*. Sebagai contoh, *zxcvbn* memiliki mekanisme untuk mengenali pola *leet*, pola *keyboard*, serta kamus yang lebih luas dibandingkan kamus RockYou yang digunakan dalam penelitian ini. Hasil ini juga sesuai dengan penjelasan Wheeler (2016) yang menyebutkan bahwa *zxcvbn* menggunakan beberapa metode pencocokan, seperti pencocokan kamus, pola *leet*, dan pola *keyboard*. Sementara itu, penelitian ini hanya merepresentasikan sebagian mekanisme tersebut melalui fitur-fitur struktural yang dirancang sebagai masukan model klasifikasi.

Berdasarkan hasil tersebut, masih terdapat beberapa peluang untuk meningkatkan performa model pada penelitian selanjutnya. Beberapa di antaranya adalah menambahkan fitur untuk mengenali pola *leet*, memperluas deteksi pola *keyboard*, serta menggunakan kamus referensi yang lebih lengkap. Namun, perlu diperhatikan bahwa penambahan seluruh mekanisme tersebut akan membuat pendekatan yang dikembangkan semakin menyerupai mekanisme internal *zxcvbn*. Oleh karena itu, pengembangan tersebut berada di luar ruang lingkup penelitian ini, yang berfokus pada pembangunan model klasifikasi berbasis fitur struktural.

Temuan ini menunjukkan bahwa kesalahan yang masih terjadi lebih banyak disebabkan oleh keterbatasan fitur yang digunakan daripada keterbatasan kemampuan model dalam mempelajari pola pada data. Oleh karena itu, aspek tersebut menjadi salah satu keterbatasan yang dibahas pada subbab berikutnya.

4.7.6 Keterbatasan Penelitian

Penelitian ini masih memiliki beberapa keterbatasan yang perlu diperhatikan. Pertama, label yang digunakan merupakan hasil penilaian *zxcvbn*. Dalam penelitian ini, label tersebut digunakan sebagai acuan (*ground truth*) karena merepresentasikan mekanisme penilaian yang menjadi target pembelajaran model. Namun, label tersebut tidak dapat dianggap sebagai representasi yang sepenuhnya menggambarkan kekuatan kata sandi dalam seluruh konteks penggunaan. Kedua, fitur struktural yang digunakan belum mampu mengenali seluruh pola yang digunakan oleh *zxcvbn*, seperti pola *leet substitution*, pola *keyboard* yang lebih kompleks, maupun kata sandi yang tidak terdapat pada kamus referensi. Kondisi ini terlihat pada hasil analisis kesalahan yang telah dibahas pada Subbab 4.7.5. Ketiga, kamus referensi yang digunakan hanya berasal dari RockYou, sehingga belum mencakup seluruh variasi kata sandi yang mungkin digunakan di dunia nyata.

Keempat, *dataset* utama dan kamus referensi yang digunakan berasal dari kumpulan kata sandi publik yang didominasi oleh pengguna internasional. Oleh karena itu, penelitian ini belum mengevaluasi secara khusus karakteristik maupun pola pembentukan kata sandi pengguna Indonesia sehingga kemampuan generalisasi model terhadap konteks pengguna Indonesia masih memerlukan validasi lebih lanjut. Kelima, penelitian ini membatasi panjang kata sandi yang diproses hingga 100 karakter. Pembatasan tersebut dilakukan untuk menyesuaikan dengan kebutuhan sistem yang dikembangkan pada penelitian ini. Keenam, interpretasi menggunakan SHAP hanya diterapkan pada model berbasis pohon (*tree-based model*), sehingga hasil interpretasi pada penelitian ini belum dapat langsung diterapkan pada seluruh jenis algoritma *machine learning*.

Ketujuh, penerapan *threshold* asimetris berhasil mengurangi kesalahan klasifikasi kata sandi *weak* menjadi *strong*, tetapi di sisi lain menyebabkan penurunan nilai *F1-Score* makro. Temuan ini menunjukkan adanya *trade-off* antara peningkatan keamanan dan performa klasifikasi yang perlu dipertimbangkan dalam penentuan nilai *threshold*. Keterbatasan-keterbatasan tersebut juga menunjukkan beberapa peluang untuk penelitian selanjutnya. Pengembangan fitur yang lebih lengkap, penggunaan kamus referensi yang lebih besar, serta

evaluasi menggunakan *dataset* yang lebih beragam berpotensi meningkatkan performa model pada penelitian berikutnya.

Terkait kelayakan penerapan sistem pada konteks pengguna Indonesia, perlu dipahami bahwa model melakukan deteksi melalui dua mekanisme utama, yaitu karakteristik struktural kata sandi dan fitur berbasis referensi. Kata sandi yang menggunakan nama lokal dengan panjang yang pendek masih cenderung diklasifikasikan sebagai lemah melalui fitur panjang kata sandi (*length*), yang merupakan fitur paling berpengaruh pada model dengan nilai *importance* sebesar 0,3581. Keterbatasan utama terdapat pada pola kata sandi yang memanfaatkan nama atau frasa lokal dengan panjang yang memadai, baik dikombinasikan dengan angka maupun tidak. Sebagai contoh, pengujian langsung pada *prototype* menunjukkan bahwa kata sandi "*subroto123*", yang menggabungkan nama keluarga Jawa dengan imbuhan angka, serta "*selamatpagidunia*", yang merupakan frasa bahasa Indonesia tanpa angka maupun simbol, keduanya diklasifikasikan sebagai *strong* oleh sistem.

Kedua kata sandi tersebut tidak tercatat pada kamus referensi yang digunakan dalam penelitian ini, yaitu RockYou maupun kamus internal *zxcvbn*, sementara panjangnya yang memadai mendorong fitur *length* untuk memberikan penilaian yang cenderung tinggi. Temuan ini menegaskan bahwa performa yang diperoleh pada penelitian ini belum dapat digeneralisasikan secara langsung pada korpus kata sandi pengguna Indonesia, dan tingkat kesalahan klasifikasi pada konteks tersebut berpotensi berbeda dari hasil yang diperoleh pada penelitian ini. Dengan demikian, sistem ini layak digunakan sebagai *prototype* demonstrasi maupun sebagai lapisan penyaring awal (*first-line screening*) terhadap kata sandi yang lemah secara struktural atau telah terpapar pada kebocoran data global. Namun, sistem ini belum layak dijadikan sebagai satu-satunya mekanisme penilaian kekuatan kata sandi pada layanan yang mayoritas penggunanya berasal dari Indonesia. Untuk konteks tersebut, sistem perlu terlebih dahulu dilengkapi dengan kamus referensi lokal yang memuat nama orang, kosakata, dan pola kata sandi yang umum digunakan oleh masyarakat Indonesia sebagaimana diuraikan pada Subbab 5.2.

BAB V
KESIMPULAN DAN SARAN

Bab ini memaparkan kesimpulan yang ditarik dari hasil dan pembahasan pada Bab 4, serta memberikan saran untuk pengembangan penelitian selanjutnya. Kesimpulan disusun untuk menjawab ketiga rumusan masalah sekaligus menegaskan ketercapaian tujuan penelitian, sedangkan saran dirumuskan berdasarkan keterbatasan dan temuan yang muncul selama penelitian.

5.1 Kesimpulan

Sebagai jawaban atas rumusan masalah pertama, penelitian ini berhasil membangun model klasifikasi kekuatan kata sandi ke dalam tiga kategori, yaitu *weak*, *medium*, dan *strong*, menggunakan fitur struktural dengan *zxcvbn* sebagai acuan pelabelan ulang. Berdasarkan perbandingan empat algoritma pada dua skenario penanganan ketidakseimbangan kelas, *Random Forest without balancing* dipilih sebagai model terbaik karena memberikan performa paling tinggi dan konsisten pada data validasi. Pada data uji, model final mencapai akurasi sebesar 0,8344, *F1-Score* makro sebesar 0,8121, dan *recall* kelas *weak* sebesar 0,7601 sehingga seluruh kriteria keberhasilan penelitian berhasil dipenuhi. Selain itu, melalui optimasi *threshold* asimetris menggunakan vektor [0,25; 0,25; 0,55], proporsi kesalahan klasifikasi kata sandi *weak* menjadi *strong* berhasil ditekan dari 15,04% menjadi 4,73%, berada di bawah batas keamanan sebesar 5%. Penurunan *F1-Score* makro sebesar 0,0104 merupakan konsekuensi yang dipilih secara sadar untuk mengurangi jenis kesalahan klasifikasi yang paling berisiko.

Menjawab rumusan masalah kedua, keputusan klasifikasi model dapat dijelaskan menggunakan SHAP, baik pada tingkat global maupun lokal. Secara global, panjang kata sandi (*length*) merupakan fitur paling berpengaruh dengan nilai 0,3581. Fitur ini diikuti oleh interaksi panjang dengan entropi dan fitur yang berkaitan dengan kemunculan kata sandi pada basis data kata sandi yang bocor. Pada tingkat lokal, SHAP mampu menunjukkan fitur yang mendorong maupun menentang suatu prediksi pada kata sandi tertentu. Karena keputusannya bisa dijelaskan, model ini tidak lagi bekerja sebagai *black box*. Panjang kata sandi dan keterpaparannya terhadap basis data kata sandi yang bocor terbukti jadi faktor paling menentukan, sejalan dengan dua syarat inti yang ditetapkan NIST SP 800-63B-4, yaitu panjang minimum dan pemeriksaan terhadap kata sandi yang telah bocor.

Akhirnya, sebagai jawaban atas rumusan masalah ketiga, model final beserta mekanisme penjelasannya berhasil diimplementasikan ke dalam sebuah *prototype* sistem berbasis web. *Prototype* ini menggunakan model dan *threshold* yang sama persis dengan yang dipakai pada eksperimen. Oleh karena itu, performa yang dilaporkan dalam penelitian ini juga merepresentasikan performa sistem yang diimplementasikan. Hasil tersebut menunjukkan bahwa pendekatan yang dikembangkan dapat dioperasikan secara utuh, mulai dari *input* kata sandi hingga menghasilkan klasifikasi beserta penjelasannya dalam sebuah sistem berbasis web.

5.2 Saran

Penelitian selanjutnya disarankan untuk mengembangkan rekayasa fitur agar mampu mengenali pola-pola yang belum sepenuhnya terwakili dalam penelitian ini. Analisis kesalahan menunjukkan bahwa sebagian besar kesalahan klasifikasi masih terjadi pada kata sandi yang menggunakan *leet substitution* (seperti *hollyho0d*), pola ketukan *keyboard* (seperti *q1w2e3qwe*), serta kata sandi yang tidak terdapat pada kamus referensi (seperti *stoneforge7*). Oleh karena itu, penambahan fitur untuk mendeteksi pola-pola tersebut serta penggunaan kamus referensi yang lebih luas berpotensi meningkatkan kemampuan model dalam mengenali kata sandi yang lemah. Selain itu, karena hasil interpretasi SHAP menunjukkan bahwa panjang kata sandi merupakan fitur yang paling berpengaruh, penelitian berikutnya juga dapat mengembangkan fitur yang lebih mampu membedakan kata sandi panjang dengan entropi rendah dari kata sandi yang benar-benar kuat sehingga penilaian tidak mudah dipengaruhi oleh panjang karakter saja.

Selain pengembangan model, validasi terhadap asumsi yang digunakan dalam penelitian ini juga perlu dilakukan. Keandalan *zxcvbn* sebagai acuan pelabelan dapat dievaluasi lebih lanjut dengan membandingkannya terhadap *password strength meter* lain atau melalui penilaian pakar. Di samping itu, kemampuan generalisasi model perlu diuji menggunakan dataset yang lebih beragam, termasuk *dataset* yang merepresentasikan karakteristik pengguna Indonesia. Pengumpulan data *primer* maupun pemanfaatan *dataset* yang relevan dengan konteks lokal dapat dilakukan untuk mengevaluasi performa model pada pola pembentukan kata sandi yang umum digunakan oleh pengguna Indonesia. Selain itu, kamus referensi dapat diperkaya dengan kosakata bahasa Indonesia, nama orang, nama daerah, maupun istilah yang umum digunakan oleh pengguna Indonesia sehingga model lebih mampu mengenali karakteristik kata sandi pada konteks lokal.

Dari sisi implementasi, *prototype* web masih berfungsi sebagai sarana demonstrasi dan belum dirancang untuk penggunaan pada lingkungan produksi. Oleh karena itu, penelitian selanjutnya dapat mengembangkan aspek keamanan infrastruktur, skalabilitas sistem, serta pengujian beban agar aplikasi lebih siap diterapkan pada penggunaan nyata. Selain itu, sistem juga dapat dikembangkan dengan menambahkan fitur rekomendasi perbaikan kata sandi berdasarkan hasil interpretasi model sehingga tidak hanya memberikan penilaian kekuatan kata sandi, tetapi juga membantu pengguna memperbaiki kata sandinya secara lebih terarah.

DAFTAR PUSTAKA

Alajaji, A., & Althaqeb, M. (2025). ROBENS: A Robust Ensemble System for Password Strength Classification. *IEEE Access*, *13*, 173665–173676. https://doi.org/10.1109/ACCESS.2025.3614377

Alkurdi, Y., Fayoumi, M. A., Al-badarneh, A., & Al-Haija, Q. A. (2023). Enhancing Password Security via Supervised Learning. *2023 International Conference on Information Technology (ICIT)*, 256–259. https://doi.org/10.1109/ICIT58056.2023.10226141

Alshammari, A. H., Bencsik, G., & Ali, A. H. (2026). A Survey of Six Classical Classifiers, Including Algorithms, Methodological Characteristics, Foundational Variants, and Recent Advances. *Algorithms*, *19*(1), 37. https://doi.org/10.3390/a19010037

Asaduzzaman, A., D'Souza, D., Uddin, M. R., & Woldeyes, Y. (2024). Increase Security by Analyzing Password Strength using Machine Learning. *2024 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, 32–37. https://doi.org/10.1109/ECTIDAMTNCON60518.2024.10479995

Bhavik Bansal. (2019). *Password Strength Classifier Dataset* [Dataset]. Kaggle. https://www.kaggle.com/datasets/bhavikbb/password-strength-classifier-dataset

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Darbutaitė, E., Stefanovič, P., & Ramanauskaitė, S. (2023). Machine-Learning-Based Password-Strength-Estimation Approach for Passwords of Lithuanian Context. *Applied Sciences*, *13*(13), 7811. https://doi.org/10.3390/app13137811

Divya, R., Gaganashree, Devamane, S. B., Dharshini, V., & Deepika, S. (2023). Performance Analysis of Machine Learning Algorithms for Password Strength Check. *2023 International Conference on Computational Intelligence for Information, Security and Communication Applications (CIISCA)*, 338–343. https://doi.org/10.1109/CIISCA59740.2023.00071

He, D., Liu, Z., Chan, S., Zhou, B., & Guizani, M. (2025). Evaluating the Impact of Special Characters on Password Security: A Comparative Analysis. *IEEE Network*, *39*(6), 218–225. https://doi.org/10.1109/MNET.2025.3540319

He, D., Liu, Z., Zhu, S., Chan, S., & Guizani, M. (2024). Special Characters Usage and Its Effect on Password Security. *IEEE Internet of Things Journal*, *11*(11), 19440–19453. https://doi.org/10.1109/JIOT.2024.3367323

Kayalvili, S., Devadharshini, T., Dharaneesh, N., & Dhanush, P. (2024). Password Strength Checker Using Machine Learning. *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–5. https://doi.org/10.1109/ICCCNT61001.2024.10725352

Khalid, N., Zahid, A., & Nisar, S. (2024). A Password Strength Classifier Using Supervised Machine Learning. *2024 21st International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, 306–314. https://doi.org/10.1109/IBCAST61650.2024.10876915

Kim, S. J., & Lee, B. M. (2024). A Novel Approach to Password Strength Evaluation Using ChatGPT-Based Prompt Metrics. *IEEE Access*, *12*, 175071–175080. https://doi.org/10.1109/ACCESS.2024.3503653

Lingamsetty, H., Sri Nitya, N. S., Teja M, B. M., Odelu, V., Mookherji, S., Narayan, K. R., Prasath, R., & Reddy, A. G. (2024). A Penalty-Based Entropy Driven Universal Password Strength for Lightweight Devices. *2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 25–30. https://doi.org/10.1109/COMSNETS59351.2024.10427138

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, *2*(1), 56–67. https://doi.org/10.1038/s42256-019-0138-9

Lundberg, S. M., & Lee, S.-I. (2017). *A Unified Approach to Interpreting Model Predictions*.

Mo, J., Kuang, H., & Li, X. (2026). *Password Strength Detection via Machine Learning: Analysis, Modeling, and Evaluation* (arXiv:2505.16439). arXiv. https://doi.org/10.48550/arXiv.2505.16439

NordPass. (2026). *Top 200 Most Common Passwords*. https://nordpass.com/most-common-passwords-list/

Opitz, J., & Burst, S. (2021). *Macro F1 and Macro F1* (arXiv:1911.03347). arXiv. https://doi.org/10.48550/arXiv.1911.03347

Pande, V. (2026). Machine Learning-Based Password Strength Classification Using Entropy and Markov Chain Analysis. *2026 Fourth International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, 533–538. https://doi.org/10.1109/ICAISS68683.2026.11525503

Pawade, P., Kulkarni, M., Naik, S., Raut, A., & Wagh, K. S. (2024). Efficiency Comparison of Dataset Generated by LLMs using Machine Learning Algorithms. *2024 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 1–6. https://doi.org/10.1109/ESCI59607.2024.10497340

Powers, D. M. W. (2011). EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION. *Journal of Machine Learning Technologies*, *2*. https://doi.org/https://doi.org/10.48550/arXiv.2010.16061

Rehman, H., Ahmed, A., Hussain, A., Umar, M., Siraj, M., Ahmed, W., Muhamnad, M., & Rehman, E. (2024). Password Strength Classification Using Machine Learning Methods. *2024 Global Conference on Wireless and Optical Technologies (GCWOT)*, 1–7. https://doi.org/10.1109/GCWOT63882.2024.10805622

Sarkar, S., & Nandan, M. (2022). Password Strength Analysis and its Classification by Applying Machine Learning Based Techniques. *2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA)*, 1–5. https://doi.org/10.1109/ICCSEA54677.2022.9936117

Shimaoka, A. M., Ferreira, R. C., & Goldman, A. (2024). The evolution of CRISP-DM for Data Science: Methods, Processes and Frameworks. *SBC Reviews on Computer Science*, *4*(1), 28–43. https://doi.org/10.5753/reviews.2024.3757

Shinde, S., Chandragiri, A., Chaudhari, A., Late, U., & Baheti, A. (2025). Quantum-Resistant Password Generation A Comprehensive Model with QNN, Simplified Entropy, and QKD Integration. *2025 Global Conference in Emerging Technology (GINOTECH)*, 1–9. https://doi.org/10.1109/GINOTECH63460.2025.11076641

Shreya, K., Divakarla, L. P., Kumar, K. J., & Vishwas, H. N. (2024). Fortifying Digital Security: A Machine Learning and Explainable AI Framework for Password Strength Assessment. *2024 8th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 1358–1364. https://doi.org/10.1109/ICECA63461.2024.10800823

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Temoshok, D., Fenton, J., Choong, Y.-Y., Lefkovitz, N., Regenscheid, A., & Richer, J. (2025). *Digital Identity Guidelines: Authentication and Authenticator Management* (NIST SP 800-63B-4 2pd; p. NIST SP 800-63B-4 2pd). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-63B-4.2pd

Thai, B. L. T., Takii, T., & Tanaka, H. (2024). Password Classification Using Machine Learning and Natural Language Processing Techniques: Methods and Evaluations. *2024 8th Cyber Security in Networking Conference (CSNet)*, 147–150. https://doi.org/10.1109/CSNet64211.2024.10851759

Thai, B. L. T., & Tanaka, H. (2024a). A statistical Markov-based password strength meter. *Internet of Things*, *25*, 101057. https://doi.org/10.1016/j.iot.2023.101057

Thai, B. L. T., & Tanaka, H. (2024b). A Study on Markov-Based Password Strength Meters. *IEEE Access*, *12*, 69066–69075. https://doi.org/10.1109/ACCESS.2024.3401195

Verizon. (2026). *2026 Data Breach Investigations Report*. https://www.verizon.com/business/resources/reports/dbir/

Wheeler, D. L. (2016). *zxcvbn: Low-Budget Password Strength Estimation*.

William J. Burns. (2019). *RockYou* [Dataset]. Kaggle. https://www.kaggle.com/datasets/wjburns/common-password-list-rockyoutxt/data

Wirth, R., & Hipp, J. (2000). *CRISP-DM: Towards a Standard Process Model for Data Mining*.

Yu, W., Ni, L., Shi, Y., Ji, Q., & Liu, C. (2024a). Sequential-Spatial-Aware Attribute Classification of Ultra-Short Password. *2024 4th International Conference on Electronic Information Engineering and Computer Science (EIECS)*, 1123–1126. https://doi.org/10.1109/EIECS63941.2024.10800624

Yu, W., Ni, L., Shi, Y., Ji, Q., & Liu, C. (2024b). Vulnerability Analysis of Passwords Based on Real Data. *2024 6th International Conference on Frontier Technologies of Information and Computer (ICFTIC)*, 63–68. https://doi.org/10.1109/ICFTIC64248.2024.10912863

LAMPIRAN

Link Repository Kode Eksperimen: https://github.com/adiprasetyo31/obsidian-vault-PSM-with-Machine-Learning

Lampiran A. Hasil optimasi *threshold* seluruh model pada data validasi

Model	**F1-Score Makro Sebelum**	**F1-Score Makro Sesudah**	**Selisih**	**weak→strong**	**Recall weak**	**Threshold**
Random Forest	0,8215	0,8111	-0,0104	4,88%	0,7530	[0,25; 0,25; 0,55]
XGBoost	0,8205	0,8094	-0,0111	4,73%	0,7558	[0,20; 0,20; 0,45]
Logistic Regression	0,7779	0,7243	-0,0536	4,65%	0,7824	[0,10; 0,20; 0,55]
SVM Linear	0,7671	0,7147	-0,0524	4,68%	0,7059	[0,15; 0,20; 0,55]