Analisis Kekuatan Kata Sandi melalui Prinsip Logika Proposisional

Natalia Desiany Nursimin - 13523157[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*nataliadesianyy@gmail.com*, *13523157@std.stei.itb.ac.id*

***Abstrak*— Keamanan kata sandi sangat penting dalam melindungi informasi pribadi dan aset digital, terutama di tengah meningkatnya ancaman siber di era digital. Meskipun demikian, banyak pengguna yang masih cenderung menggunakan kata sandi yang lemah, sehingga membuat mereka rentan terhadap berbagai metode serangan siber seperti *brute force, phishing*, dan *dictionary attack*. Makalah ini akan membahas mengenai penerapan logika proposisional dalam menganalisis dan meningkatkan kekuatan kata sandi. Pendekatan berbasis logika proposisional memberikan kerangka sistematis untuk mengevaluasi elemen-elemen penting pembentuk kata sandi, seperti panjang, kompleksitas karakter, pola pengulangan, serta pola keyboard yang dapat mengurangi tingkat keamanan. Dengan menggunakan formula logika, kelemahan kata sandi dapat teridentifikasi secara lebih tepat dibandingkan dengan pendekatan tradisional yang hanya mengandalkan entropi. Hasil analisis menunjukkan bahwa penerapan logika proposisional dapat secara efektif mengevaluasi kekuatan kata sandi, serta memberikan wawasan lebih dalam perancangan algoritma pemeriksaan keamanan kata sandi. Makalah ini diharapkan dapat meningkatkan kesadaran pengguna terhadap pentingnya memilih kata sandi yang kuat, serta memberikan kontribusi dalam perancangan sistem keamanan yang lebih adaptif .**

***Kata kunci*—Kata sandi, logika proposisional, keamanan data, evaluasi kata sandi**

I. PENDAHULUAN

Dalam era digital yang semakin berkembang, keamanan kata sandi telah menjadi salah satu komponen yang krusial dalam menjaga privasi dan melindungi informasi pribadi. Kata sandi memiliki peran yang sangat penting, yaitu sebagai alat perlindungan utama untuk melindungi berbagai aset digital, mulai dari akses ke perangkat, akun media sosial, hingga data perusahaan. Namun, efektivitas kata sandi sebagai sebuah benteng keamanan sangat bergantung pada kekuatan dan keunikannya. Sayangnya, banyak pengguna yang masih menggunakan kata sandi yang lemah dan mudah ditebak, seperti kombinasi angka berurutan atau kata yang sangat umum, sehingga rentan dieksploitasi oleh pihak-pihak yang tidak bertanggung jawab.

Ketergantungan pada kata sandi yang lemah telah menciptakan celah besar dalam sistem keamanan. Pola seperti "123456," atau "password123," masih sering digunakan oleh para pengguna. Padahal, serangan terhadap kata sandi dan pencurian data sudah menjadi semakin canggih seiring dengan kemajuan teknologi. Berbagai metode serangan, seperti *phishing*, *brute force attack*, *dictionary attack*, *vishing*, dan *baiting* menjadi ancaman nyata bagi para pengguna internet.

Salah satu metode serangan yang paling umum dilakukan merupakan *phishing*, di mana penyerang menggunakan email atau situs web palsu untuk mencuri informasi sensitif seperti kata sandi dan detail login. Selain itu, terdapat juga metode serangan *brute force* yang dilakukan dengan mencoba setiap kemungkinan kombinasi kata sandi hingga menemukan yang sesuai. Metode serangan *dictionary attack* bersifat lebih spesifik karena menggunakan daftar kata-kata yang umum digunakan sebagai kata sandi. Terdapat juga penggunaan serangan berbasis rekayasa sosial seperti *vishing* dan *baiting*. Dalam serangan *vishing*, penyerang menggunakan panggilan telepon untuk memanipulasi korban agar memberikan informasi sensitif. Sementara itu, dalam *baiting*, penyerang menawarkan iming-iming hadiah untuk memikat korban mengungkapkan data penting mereka. Hal-hal ini menjadi sebuah tantangan besar bagi keamanan data.

Fenomena lemahnya keamanan kata sandi tidak hanya disebabkan oleh kelalaian pengguna, tetapi juga oleh pendekatan tradisional yang selama ini digunakan dalam mengevaluasi kekuatan kata sandi. Hingga saat ini, penilaian kekuatan kata sandi sering kali masih didasarkan pada entropi informasi, yang mengukur panjang dan kompleksitas karakter dalam kata sandi. Namun, pendekatan ini memiliki keterbatasan karena tidak mempertimbangkan pola logis yang sering digunakan manusia dalam menciptakan kata sandi. Misalnya, pola seperti pengulangan karakter, urutan alfabet, atau penggantian karakter sederhana.

Untuk membantu mengatasi permasalahan lemahnya keamanan kata sandi, diperlukan pendekatan yang lebih efektif dan terukur dalam menganalisis kekuatan serta pola kerentanan kata sandi. Salah satu solusi yang dapat digunakan adalah dengan menerapkan prinsip logika proposisional untuk menganalisis kekuatan kata sandi. Logika proposisional merupakan sebuah sistem yang merepresentasikan proposisi-proposisi dalam bentuk formula. Formula ini dibentuk dengan menggabungkan proposisi-proposisi atomik menggunakan penghubung logis seperti konjungsi ($\wedge$), disjungsi ($\vee$) atau ingkaran ($\sim$). Pendekatan ini dapat dimanfaatkan untuk memberikan kerangka sistematis dalam menganalisis hubungan

logis antara elemen-elemen dalam kata sandi.

Dalam konteks hubungannya dengan analisis kekuatan kata sandi, logika proposisional dapat digunakan untuk memahami dan mencari pola-pola logis yang sering muncul dalam kata sandi, seperti penggunaan huruf kapital, huruf kecil, angka, dan simbol. Pendekatan ini memungkinkan dilakukannya identifikasi struktur yang biasa digunakan pengguna atau pola pengulangan karakter. Dengan menggunakan prinsip logika proposisional, analisis dapat dilakukan dengan efektif dengan cara mengevaluasi apakah pola-pola yang digunakan dalam kata sandi memilih keamanan yang kuat agar tidak dapat mudah diretas.

Selain itu, penerapan logika proposisional juga memiliki potensi untuk diintegrasikan ke dalam sistem otomatis untuk mengevaluasi kekuatan kata sandi. Sistem ini dapat memberikan peringatan kepada pengguna saat mendeteksi kata sandi dengan pola lemah yang berisiko tinggi untuk di eksploitasi oleh pihak yang tidak bertanggung jawab. Pendekatan ini juga dapat dijadikan sebuah alat bantu dalam merancang algoritma pemeriksaan kekuatan kata sandi yang lebih adaptif terhadap ancaman serangan siber yang terus berkembang.

Makalah ini bertujuan untuk membahas mengenai penerapan logika proposisional dalam meningkatkan analisis dan pengujian kekuatan kata sandi. Pendekatan ini diharapkan dapat memberikan wawasan yang lebih mendalam mengenai kelemahan kata sandi yang sering dieksploitasi dalam serangan-serangan dunia maya. Melalui penerapan prinsip logika proposisional, diharapkan dapat ditemukan metode yang lebih efektif untuk merancang solusi yang dapat meningkatkan keamanan informasi pribadi.

II. LANDASAN TEORI

Logika proposisional merupakan sebuah cabang logika matematika yang mempelajari mengenai proposisi-proposisi yang memiliki nilai kebenaran, yaitu benar atau salah. Setiap proposisi terdiri dari suatu kalimat deklaratif yang bersifat tidak ambigu dan memiliki nilai kebenaran yang pasti. Dalam konteks analisis kekuatan kata sandi, logika proposisional dapat digunakan untuk menggambarkan hubungan antara komponen-komponen pembentuk kata sandi, serta mengevaluasi kekuatan dan kerentanannya. Logika proposisional terdiri dari simbol, variabel, dan operator logika yang digunakan untuk menyusun formula yang mewakili kondisi tertentu. Formula ini dapat dirancang untuk menganalisis berbagai aspek kata sandi, seperti panjang, kompleksitas karakter, pola pengulangan, dan penggunaan urutan karakter tertentu. Penerapan ini akan membantu mengidentifikasi pola-pola umum yang berpotensi untuk dieksploitasi.

Logika proposisional terdiri atas berbagai komponen-komponen penting yang dapat dilihat sebagai berikut:

A. Proposisi

Proposisi merupakan sebuah pernyataan dasar yang memiliki nilai kebenaran yang sudah ditentukan. Contoh beberapa proposisi dalam konteks analisis kata sandi adalah sebagai berikut:

- P: "Kata sandi mengandung huruf besar."
- Q: "Kata sandi mengandung huruf kecil."
- R: "Kata sandi mengandung angka."
- S: "Kata sandi mengandung simbol."

B. Nilai kebenaran

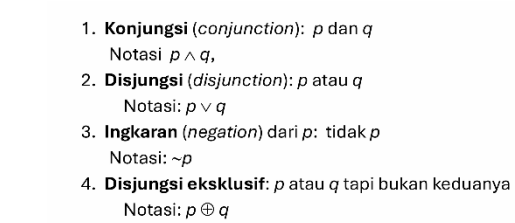
Setiap proposisi dapat memiliki dua kemungkinan nilai kebenaran, yaitu:

- Benar: Proposisi sesuai dengan kenyataan atau kondisi yang ada.
- Salah: Proposisi bertentangan dengan kenyataan atau kondisi yang ada.

C. Operator Logika

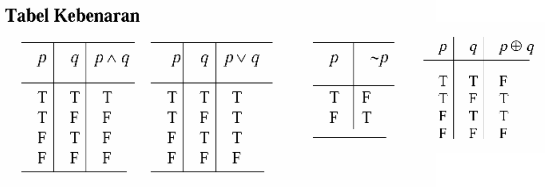
Operator logika merupakan sebuah simbol yang digunakan untuk menghubungkan proposisi-proposisi dalam sebuah formula logika. Beberaoa contoh operator logika adakah sebagai berikut:

- Konjungsi (**∧**)
  Konjungsi menggabungkan dua proposisi dengan kata "dan". Dalam konjungsi, kedua proposisi harus bernilai benar agar hasil konjungsi tersebut benar. Jika salah satu dari proposisi bernilai salah, maka hasil konjungsi akan salah.
- Disjungsi (V)
  Disjungsi menggabungkan dua proposisi dengan kata "atau". Dalam disjungsi, hasilnya akan benar jika salah satu dari proposisi p atau q bernilai benar, atau keduanya bernilai benar. Disjungsi hanya bernilai salah jika kedua proposisi bernilai salah.
- Ingkaran **(**~)
  Dalam logika proposisional, negasi digunakan untuk membalikkan nilai kebenaran sebuah proposisi. Jika proposisi p bernilai benar, maka ~p akan bernilai salah. Namun, jika p bernilai salah, maka ~p akan bernilai benar.
- Disjungsi Eksklusif (⊕)
  Disjungsi eksklusif menggabungkan dua proposisi dengan kata "atau, tapi bukan keduanya." Hasil disjungsi eksklusif benar jika salah satu proposisi benar, tetapi tidak keduanya. Jika kedua proposisi bernilai benar atau kedua proposisi bernilai salah, maka hasil disjungsi eksklusif akan bernilai salah.



Gambar 2.1 Operator-operator logika
Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/01-Logika-2024.pdf

Berikut merupakan nilai table kebenaran yang berlaku untuk operator konjungsi, disjungsi, ingkaran dan disjungsi eksklusif:



Gambar 2.2 Tabel kebenaran operator
Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/01-Logika-2024.pdf

D. Formula Logika

Formula logika merupakan ekspresi yang dibentuk dari kombinasi proposisi dan operator logika. Contoh formula logika dalam evaluasi kata sandi, sesuai yang sudah dibuat pada bagian proposisi, yaitu:

- Kata sandi harus mengandung huruf besar dan huruf kecil.
  P ∧ Q
- Kata sandi harus mengandung huruf besar, huruf kecil, dan angka
  P ∧ Q ∧ R
- Kata sandi mengandung huruf besar atau huruf kecil, atau keduanya:
  P ∨ Q
- Kata sandi mengandung huruf besar atau angka, tetapi bukan keduanya:
  P ⊕ R
- Kata sandi harus mengandung setidaknya huruf besar atau simbol:
  P ∨ S
- Kata sandi tidak boleh mengandung simbol:
  ~S
- Kata sandi mengandung huruf besar dan simbol, tetapi tidak angka:
  P ∧ S ∧ ~R
- Kata sandi mengandung huruf kecil, angka, atau simbol:
  Q ∨ R ∨ S
- Kata sandi mengandung semua jenis karakter (huruf besar, huruf kecil, angka, simbol):
  P∧Q∧R∧S
- Kata sandi tidak boleh mengandung huruf besar dan huruf kecil secara bersamaan:
  ~(P ∧ Q)

III. PEMBAHASAN

Di tengah berkembangnya berbagai ancaman siber seperti serangan brute force, phishing, dan rekayasa sosial, penggunaan kata sandi yang kuat menjadi krusial. Kata sandi yang kuat merupakan salah satu pilar utama dalam menjaga keamanan data digital. Kata sandi yang lemah sering kali menjadi celah yang dimanfaatkan oleh penyerang untuk mengakses data sensitif. Dalam upaya untuk meningkatkan ketahanan kata sandi, penerapan logika proposisional memberikan kerangka kerja yang sistematis dan terukur. Pendekatan ini memungkinkan kita untuk mengevaluasi kekuatan kata sandi berdasarkan kriteria yang jelas dan terdefinisi, seperti panjang kata sandi, kompleksitas karakter, serta keberadaan pola yang dapat dimanfaatkan oleh algoritma serangan.

Di tengah pesatnya perkembangan ancaman siber, seperti serangan brute force, phishing, dan rekayasa sosial, penggunaan kata sandi yang kuat menjadi sebuah faktor penting dalam menjaga keamanan data digital. Kata sandi yang kuat dapat membantu mencegah terjadinya eksploitasi dan pembocoran akses informasi pribadi. Untuk memperkuat perlindungan terhadap data, diperlukan suatu kerangka kerja yang lebih terstruktur dan objektif, yang memungkinkan kita untuk mengevaluasi kekuatan kata sandi secara sistematis. Hal ini dapat dicapai dengan menerapkan logika proposisional. Penerapan logika proposisional akan dapat membantu menganalisis dan mengevaluasi elemen-elemen krusial yang memengaruhi keamanan kata sandi. Berikut ini merupakan pembahasan mengenai penerapan kerangka berbasis logika proposisional untuk menganalisis dan memperkuat kekuatan kata sandi:

1) Panjang Kata Sandi

Panjang sebuah kata sandi dapat berpengaruh dalam menentukan kekuatannya. Kata sandi yang lebih panjang dan terdiri dari jumlah kombinasi yang besar akan dapat meningkatkan kesulitan bagi penyerang untuk menebaknya. Hal ini akan sangat efektif dalam mengatasi metode serangan *brute force*, di mana penyerang berusaha untuk menebak kata sandi dengan mencoba setiap kombinasi karakter. Semakin panjang sebuah kata sandi, maka semakin besar pula jumlah kombinasi yang harus dicoba, sehingga keamanannya dapat meningkat.

Sebagai contoh, kata sandi yang terdiri dari 8 karakter dengan hanya huruf kecil akan memiliki 26^8 kemungkinan kombinasi, sedangkan kata sandi dengan 12 karakter akan memiliki 26^{12} kemungkinan kombinasi. Panjang kata sandi memberikan berbagai manfaat penting dalam meningkatkan keamanan. Kata sandi yang lebih panjang dapat meningkatkan tingkat ketidakpastian, sehingga mempersulit penyerang untuk menebak kombinasi yang benar. Selain itu, penambahan panjang kata sandi memberikan peluang untuk memasukkan lebih banyak variasi dan kompleksitas karakter, yang berpengaruh dalam meningkatkan tingkat keamanan kata sandi.

Panjang kata sandi dapat direpresentasikan dalam logika proposisional sebagai berikut:

- L: Panjang kata sandi
- $L \geq 8$: Panjang kata sandi harus terdiri dari minimal 8 karakter
- Formula:
  $L \geq 8$ = "Kata sandi memenuhi panjang minimal"

2) Kompleksitas Karakter

Kompleksitas karakter merupakan penggunaan berbagai jenis karakter dalam kata sandi, seperti huruf besar, huruf kecil, angka, dan simbol. Semakin kompleks kata sandi, akan semakin sulit ditebak, baik oleh serangan manual maupun otomatis. Sebaliknya, kata sandi yang hanya menggunakan satu jenis karakter, seperti huruf kecil, lebih mudah untuk diserang karena

kombinasinya hanya sedikit.

Kompleksitas karakter dapat direpresentasikan dalam logika proposisional sebagai berikut:

- P: "Kata sandi mengandung huruf besar."
- Q: "Kata sandi mengandung huruf kecil."
- R: "Kata sandi mengandung angka."
- S: "Kata sandi mengandung simbol."
- Formula:
  $(P \land Q \land R \land S) \rightarrow$ "Kata sandi bersifat kuat.
  Jika salah satu dari P, Q, R, atau S tidak terpenuhi, maka kata sandi akan dianggap lemah.

3) Pola Pengulangan Karakter

Pola pengulangan karakter, seperti "aaaaaa" atau "111111", membuat kata sandi mudah ditebak. Hal ini dikarenakan pengulangan karakter mengurangi variasi dalam kata sandi, sehingga mempermudah penyerangan. Pola pengulangan karakter dapat direpresentasikan dalam logika proposisional sebagai berikut:

- T: Kata sandi mengandung pengulangan karakter.
- Formula:
  $T \rightarrow$ " Kata sandi bersifat lemah."
  Jika variabel T bersifat benar, maka kata sandi akan dikategorikan lemah.

4) Pola Keyboard atau Pola Umum

Pola urutan keyboard dan urutan karakter umum dan berurut, seperti "123456", "abcdef", "password" sering digunakan karena mudah diingat. Namun, pola ini sudah sangat dikenal, sehingga rentan akan *dictionary attack*. Algoritma serangan modern sering kali mencoba pola ini terlebih dahulu, karena kemungkinan besar digunakan oleh banyak pengguna.

Pola keyboard dapat direpresentasikan dalam logika proposisional sebagai berikut:

- U: Kata sandi mengandung urutan keyboard atau pola umum.
- Formula:
  $U \rightarrow$ "Kata sandi bersifat lemah."
  Jika variabel U bernilai benar, maka kata sandi akan dianggap lemah.

Dengan menggabungkan semua aspek yang telah dinyatakan sebelumnya, dapat dibangun formula logika proposisional untuk mengevaluasi kekuatan kata sandi secara menyeluruh. Formula gabungan ini mencakup panjang, kompleksitas, dan deteksi pola yang lemah, sebagai berikut:

$(L \geq 8) \land (P \land Q \land R \land S) \land (\sim T) \land (\sim U) \rightarrow$ "Kata sandi kuat."

Penjelasan variabel-variabel:

- $L \geq 8$**:** Panjang kata sandi minimal 8 karakter.
- $P \land Q \land R \land S$**:** Kata sandi harus mengandung huruf besar, huruf kecil, angka, dan simbol.
- $\sim T$**:** Tidak mengandung pengulangan karakter yang berlebihan.
- $\sim U$**:** Tidak mengandung pola keyboard atau urutan umum.

  Jika salah satu syarat tersebut tidak terpenuhi, maka kata sandi dinilai lemah..

IV. IMPLEMENTASI

Berikut merupakan kode program untuk menganalisis kekuatan password yang telah dirancang dirancang dengan pendekatan berbasis logika proposisional. Program ini dikembangkan dengan menggunakan bahasa Python. Program dirancang dengan menggunakan beberapa kriteria penting untuk menganalisis kekuatan kata sandi, seperti panjang kata sandi, keberadaan huruf besar, huruf kecil, angka, simbol, pola pengulangan, dan pola. Logika proposisional digunakan untuk mendasari pengecekan setiap kondisi tersebut, sehingga evaluasi kata sandi dapat dilakukan dengan sistematis dan terstruktur. Berikut adalah strukturnya:

- Impor Modul re
  Program ini mengimpor modul re untuk memeriksa apakah sebuah kata sandi memenuhi pola tertentu, seperti keberadaan huruf besar, angka, simbol, dan lainnya.

- Fungsi mengecek panjang kata sandi
  Fungsi pada program ini bertujuan untuk memeriksa apakah panjang kata sandi yang dimasukan memiliki minimal 8 karakter.
  Logika proposisional:
  Proposisi L: "Panjang kata sandi >= 8".
  - Jika panjang kata sandi ≥ 8, proposisi L bernilai True.
  - Jika panjang kata sandi < 8, proposisi L bernilai False.

- Fungsi Mengecek Huruf Besar
  Fungsi pada program ini bertujuan untuk memeriksa apakah kata sandi mengandung setidaknya satu huruf besar.
  Logika Proposisional:
  Proposisi P: "Mengandung huruf besar
  - Jika kata sandi mengandung huruf besar, proposisi P bernilai True.
  - Jika kata sandi tidak mengandung huruf besar, proposisi P bernilai False.

- Fungsi Mengecek Huruf Kecil
  Fungsi pada program ini bertujuan untuk memeriksa apakah kata sandi mengandung setidaknya satu huruf kecil.
  Logika Proposisional:
  Proposisi Q: "Mengandung huruf kecil"
  - Jika kata sandi mengandung huruf kecil, proposisi Q bernilai True.
  - Jika kata sandi tidak mengandung huruf kecil, proposisi Q bernilai False.

- Fungsi Mengecek Angka
  Fungsi pada program ini bertujuan untuk memeriksa apakah kata sandi mengandung setidaknya satu angka.
  Logika Proposisional:
  Proposisi R: "Mengandung angka"
  - Jika kata sandi mengandung angka, proposisi R bernilai True.
  - Jika kata sandi tidak mengandung angka, proposisi R bernilai False.

- Fungsi Mengecek Simbol

Fungsi program ini bertujuan untuk memeriksa apakah kata sandi mengandung setidaknya satu simbol khusus.
Logika Proposisional:
Proposisi S: "Mengandung simbol"
- Jika kata sandi mengandung simbol, proposisi S bernilai True.
- Jika kata sandi tidak mengandung simbol, proposisi S bernilai False.

- Fungsi Mengecek Pola Pengulangan
  Fungsi pada program ini bertujuan untuk memeriksa apakah kata sandi mengandung pengulangan karakter yang signifikan.
  Logika Proposisional:
  Proposisi T: "Ada pengulangan karakter"
  - Jika kata sandi mengandung pengulangan karakter yang signifikan, proposisi T bernilai True.
  - Jika tidak ada pengulangan karakter, proposisi T bernilai False.

- Fungsi Mengecek Pola Umum
  Fungsi pada program ini bertujuan untuk memeriksa apakah kata sandi mengandung pola umum yang mudah ditebak, seperti urutan angka atau kata umum.
  Logika Proposisional:
  Proposisi U: "Tidak ada pola umum"
  - Jika kata sandi tidak mengandung pola umum, proposisi U bernilai True.
  - Jika kata sandi mengandung pola umum, proposisi U bernilai False.

```python
import re

# Fungsi untuk mengecek panjang kata sandi
def panjang_kata_sandi(password):
    """Memeriksa apakah panjang kata sandi minimal 8 karakter."""
    return len(password) >= 8  # L: panjang kata sandi >= 8

# Fungsi untuk mengecek huruf besar
def huruf_besar(password):
    """Memeriksa apakah kata sandi mengandung huruf besar."""
    return bool(re.search(r'[A-Z]', password))  # P: Mengandung huruf besar

# Fungsi untuk mengecek huruf kecil
def huruf_kecil(password):
    """Memeriksa apakah kata sandi mengandung huruf kecil."""
    return bool(re.search(r'[a-z]', password))  # Q: Mengandung huruf kecil

# Fungsi untuk mengecek angka
def angka(password):
    """Memeriksa apakah kata sandi mengandung angka."""
    return bool(re.search(r'[0-9]', password))  # R: Mengandung angka

# Fungsi untuk mengecek simbol
def simbol(password):
    """Memeriksa apakah kata sandi mengandung simbol spesial."""
    return bool(re.search(r'[!@#$%^&*(),.?":{}|<>]', password))  # S: Mengandung simbol

# Fungsi untuk mengecek pola pengulangan
def pola_pengulangan(password):
    """Memeriksa apakah kata sandi mengandung pengulangan karakter yang signifikan."""
    return bool(re.search(r'(.)\1{2,}', password))  # T: Terdapat pengulangan karakter

# Fungsi untuk mengecek pola umum
def pola_umum(password):
    """Memeriksa apakah kata sandi mengandung pola keyboard atau urutan angka."""
    pola = [r'1234', r'password', r'qwerty', r'abc', r'1111']
    for p in pola:
        if re.search(p, password.lower()):
            return True
    return False  # U: Tidak ada pola umum
```

Gambar 4.1 Modul dan fungsi-fungsi pemeriksaan kata sandi pada program

- Fungsi analisis_kata_sandi
  Fungsi pada program yang bertujuan untuk menggunakan semua fungsi pemeriksaan untuk mengevaluasi kata sandi dan menyimpan hasilnya dalam variabel True atau False untuk setiap proposisi.

```python
def analisis_kata_sandi(password):
    """Menganalisis kekuatan kata sandi berdasarkan berbagai kriteria."""
    # Evaluasi proposisional
    panjang_valid = panjang_kata_sandi(password)  # L: Panjang kata sandi >= 8
    huruf_besar_valid = huruf_besar(password)     # P: Mengandung huruf besar
    huruf_kecil_valid = huruf_kecil(password)     # Q: Mengandung huruf kecil
    angka_valid = angka(password)                 # R: Mengandung angka
    simbol_valid = simbol(password)               # S: Mengandung simbol
    pengulangan_valid = not pola_pengulangan(password)  # T: Tidak ada pengulangan karakter
    pola_valid = not pola_umum(password)  # U: Tidak ada pola umum

    skor = 0
    rekomendasi = []
```

Gambar 4.2 Fungsi untuk menganalisis kata sandi

- Skor dan Rekomendasi
  Program akan menilai kekuatan kata sandi yang dilakukan dengan memberi skor. Setiap kriteria akan mendapatkan skor jika dipenuhi. Lalu, di akhir program akan menampilkan rekomendasi jika kata sandi lemah pada kriteria tertentu.
  Gambar 4.3 Fungsi untuk menampilkan skor dan rekomendasi pada program

```python
    skor = 0
    rekomendasi = []

    # Penilaian berdasarkan kriteria
    if panjang_valid:
        skor += 20
    else:
        rekomendasi.append("Panjang kata sandi kurang dari 8 karakter.")

    if huruf_besar_valid:
        skor += 15
    else:
        rekomendasi.append("Tambahkan setidaknya satu huruf besar.")

    if huruf_kecil_valid:
        skor += 15
    else:
        rekomendasi.append("Tambahkan setidaknya satu huruf kecil.")

    if angka_valid:
        skor += 15
    else:
        rekomendasi.append("Tambahkan setidaknya satu angka.")

    if simbol_valid:
        skor += 15
    else:
        rekomendasi.append("Tambahkan setidaknya satu simbol (misalnya: @, #, $, dll).")

    if not pengulangan_valid:
        skor += 10
        rekomendasi.append("Hindari pengulangan karakter yang signifikan (contoh: aaa, 111).")

    if not pola_valid:
        skor += 10
        rekomendasi.append("Hindari pola umum seperti '1234', 'qwerty', atau 'password'.")

    # Menampilkan hasil analisis
    print("\nHasil Analisis Kekuatan Kata Sandi:")
    print(f"Skor Akhir: {skor}/100")
```

- Kategori kekuatan kata sandi
  Program akan mengelompokkan kekuatan kata sandi ke dalam tiga kategori, sesuai dengan skor yang diperoleh.

```python
# Kategori kekuatan berdasarkan skor
if skor >= 80:
    print("Kategori Kekuatan: Kuat")
elif 60 <= skor < 80:
    print("Kategori Kekuatan: Cukup Kuat")
else:
    print("Kategori Kekuatan: Lemah")
    print("\nRekomendasi untuk memperbaiki kata sandi:")
    for r in rekomendasi:
        print(f"- {r}")
```

Gambar 4.4 Kode untuk mengelompokkan password ke dalam beberapa kategori.

- Program Utama
  Bagian kode pada program yang berfungsi untuk

mengambil input kata sandi dari pengguna.

```
if __name__ == "__main__":
    kata_sandi = input("Masukkan kata sandi Anda: ")
    analisis_kata_sandi(kata_sandi)
```

Gambar 4.5 Kode untuk meminta input kata sandi dari pengguna

- Uji Coba Program

```
Masukkan kata sandi Anda: abc123

Hasil Analisis Kekuatan Kata Sandi:
Skor Akhir: 40/100
Kategori Kekuatan: Lemah

Rekomendasi untuk memperbaiki kata sandi:
- Panjang kata sandi kurang dari 8 karakter.
- Tambahkan setidaknya satu huruf besar.
- Tambahkan setidaknya satu simbol (misalnya: @, #, $, dll).
- Hindari pola umum seperti '1234', 'qwerty', atau 'password'.
PS C:\Users\mcn0c\Downloads\Makalah matdis>
```

Gambar 4.6 Uji Coba 1

```
PS C:\Users\mcn0c\Downloads\Makalah matdis>  python passwordanalyzer.py
Masukkan kata sandi Anda: 123

Hasil Analisis Kekuatan Kata Sandi:
Skor Akhir: 15/100
Kategori Kekuatan: Lemah

Rekomendasi untuk memperbaiki kata sandi:
- Panjang kata sandi kurang dari 8 karakter.
- Tambahkan setidaknya satu huruf besar.
- Tambahkan setidaknya satu huruf kecil.
- Tambahkan setidaknya satu simbol (misalnya: @, #, $, dll).
PS C:\Users\mcn0c\Downloads\Makalah matdis>
```

Gambar 4.7 Uji Coba 2

```
Masukkan kata sandi Anda: MyStr0ng#Password!

Hasil Analisis Kekuatan Kata Sandi:
Skor Akhir: 90/100
Kategori Kekuatan: Kuat
PS C:\Users\mcn0c\Downloads\Makalah matdis>
```

Gambar 4.8 Uji Coba 3 (Pola Umum Tidak Terpenuhi)

```
PS C:\Users\mcn0c\Downloads\Makalah matdis> python passwordanalyzer.py
Masukkan kata sandi Anda: Mynameis123

Hasil Analisis Kekuatan Kata Sandi:
Skor Akhir: 65/100
Kategori Kekuatan: Cukup Kuat
```

Gambar 4.9 Uji Coba 4

```
Masukkan kata sandi Anda: -

Hasil Analisis Kekuatan Kata Sandi:
Skor Akhir: 0/100
Kategori Kekuatan: Lemah

Rekomendasi untuk memperbaiki kata sandi:
- Panjang kata sandi kurang dari 8 karakter.
- Tambahkan setidaknya satu huruf besar.
- Tambahkan setidaknya satu huruf kecil.
- Tambahkan setidaknya satu angka.
- Tambahkan setidaknya satu simbol (misalnya: @, #, $, dll).
```

Gambar 4.10 Uji Coba 5

V. KESIMPULAN

Penerapan logika proposisional dalam menganalisis kekuatan kata sandi berguna dalam memberikan kerangka yang sistematis dan terukur untuk mengevaluasi elemen-elemen penting yang dapat memengaruhi keamanan kata sandi. Analisis berbasis logika proposisional memungkinkan untuk mengidentifikasi elemen-elemen yang dapat melemahkan kata sandi yang menjadikannya rentan terhadap berbagai jenis serangan siber. Berbeda dengan pendekatan tradisional yang mengandalkan entropi, penggunaan formula logika proposisional memberikan analisis yang lebih mendalam, relevan dan akurat. . Implementasi praktis dalam bentuk perangkat lunak juga menunjukkan potensi besar untuk memperkuat kata sandi. Dengan mengintegrasikan pendekatan ini ke dalam sistem keamanan otomatis, pengguna dapat lebih mudah menciptakan kata sandi yang lebih aman, sekaligus mengurangi potensi kebocoran data akibat penggunaan kata sandi yang lemah. Pendekatan ini juga diharapkan akan dapat meningkatkan kesadaran pengguna internet tentang pentingnya pengelolaan kata sandi yang baik.

VI. SARAN

Untuk pengembangan lebih lanjut, analisis kekuatan kata sandi berbasis logika proposisional dapat diperluas dengan menambahkan lebih banyak variabel yang mencerminkan pola pembuatan kata sandi yang semakin kompleks dan beragam. Pendekatan ini juga dapat diperkuat dengan memanfaatkan teknologi yang ada, seperti model pembelajaran mesin yang dapat mendeteksi pola-pola baru yang mungkin belum teridentifikasi dalam analisis tradisional. Dengan melatih model ini menggunakan dataset kata sandi yang besar dan beragam, sistem dapat lebih efektif dalam mengidentifikasi kelemahan umum serta memberikan rekomendasi yang lebih relevan.

VII. LAMPIRAN

Video youtube :
https://youtu.be/4Qskpa2-Vl8?si=e1rivvKKSFzK_7fl

VIII. UCAPAN TERIMA KASIH

Pertama-tama, penulis ingin memanjatkan puji syukur kepada Tuhan yang Maha Esa atas seluruh berkat dan rahmat-Nya, yang telah memungkinkan penulis untuk dapat menyelesaikan makalah ini dengan tepat waktu. Penulis juga ingin menyampaikan terima kasih yang sebesar-besarnya kepada dosen mata kuliah IF2120 Matematika Diskrit, atas seluruh bimbingan, ilmu, dan arahan yang telah diberikan selama ini. Penulis juga mengucapkan rasa terima kasih yang mendalam kepada orang tua penulis yang telah senantiasa memberikan semangat dan dukungan kepada penulis selama proses pembuatan makalah ini. Akhir kata, penulis ingin mengucapkan terima kasih kepada semua pembaca makalah ini dan berharap semoga isi makalah ini dapat memberikan manfaat bagi para pembacanya.

REFERENSI

[1] Ahmad, M., Saragih, Z. K., Akbar, N., & Gunawan, I. P. (2023). Konsep logika. Kajian Stratejik Ketahanan Nasional, Jurnal Kajian Stratejik Ketahanan Nasional, Volume J. https://repository.um.ac.id/5508/
Diakses pada 7 Januari 2025.

[2] Hanief Amarullah, A., Josias Simon Runturambi, A., & Widiawan, B. (2021). Analisis Ancaman Kejahatan Siber Bagi Keamanan Nasional Pada Masa Pandemi COVID-19.

Stratejik Ketahanan Nasional, 4(2). https://doi.org/10.7454/jkskn.v4i2.10052 Diakses pada 7 Januari 2025.

[3] Hu, G. (2018). On password strength: a survey and analysis. Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 165-186. https://www.researchgate.net/profile/Gongzhu-Hu/publication/318154948_On_Password_Strength_A_Survey_and_Analysis/links/5c3c9dfa458515a4c7259bea/On-Password-Strength-A-Survey-and-Analysis.pdf Diakses pada 7 Januari 2025.

[4] Munir, R. (2024-2025). Logika Matematika. https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/01-Logika-2024.pdf Diakses pada 7 Januari 2025.

[5] Yamin, M., Malethi, T. T., & Natali, S. (2023). Evaluasi risiko pada penggunaan password yang lemah: Analisis kasus penggunaan password umum. Jurnal Ilmiah Multidisiplin Ilmu Komputer, 1(1), 41-48. https://www.semanticscholar.org/paper/EVALUASI-RISIKO-PADA-PENGGUNAAN-PASSWORD-YANG-KASUS-M.Yamin-Malethi/6f48d8aab868ca9cef2f991550c886db04d717ac Diakses pada 7 Januari 2025.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Januari 2025

Natalia Desiany Nursimin
13523157