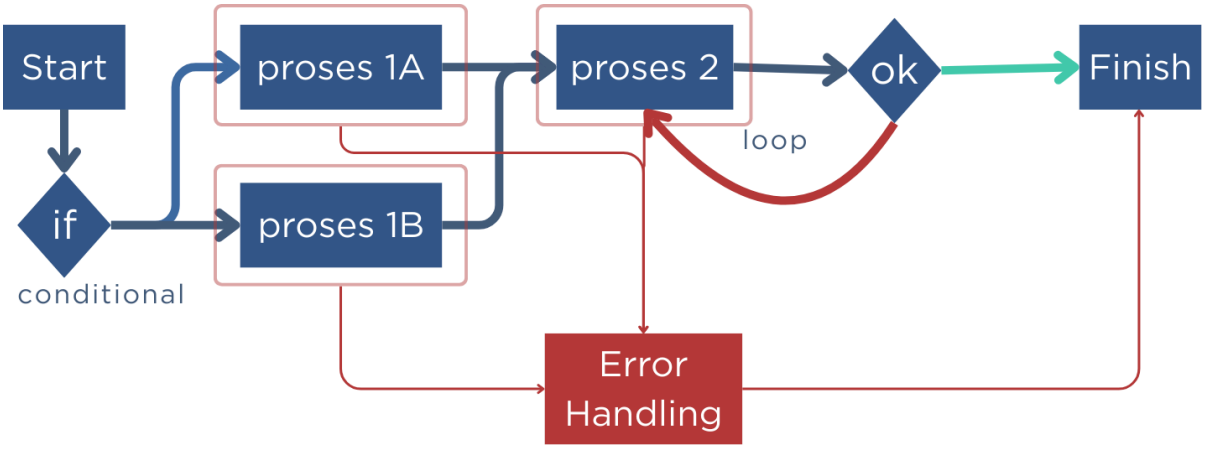
Chapter 05: Flow Control

Basic Python for Cybersecurity

Pengenalan Flow Control



- **Conditional**: logika bercabang dengan syarat
- **Loop**: perulangan dengan syarat
- **Error Handling**: mencegah aplikasi crash prematur

Syntax dalam Python untuk Flow Control

- conditional flow / branching
  - **if** : kondisi boolean yang menjadi trigger
  - **elif** : kondisi tambahan untuk jadi trigger (opsional)
  - **else** : kalau kondisi if/elif tidak memenuhi, masuk sini (opsional)
- loop
  - **for** : berulang berdasarkan counting (int, panjang list, dsb)
  - **while** : berulang berdasarkan kondisi boolean (status, limit, dsb)
- error handling
  - **try ... except** : jalankan proses utama, kalau gagal jalankan plan cadangan

Praktik: *Password Strength Checker* Fase 2

- Penyempurnaan untuk implementasi password strength checker dengan control flow.
- Mengaplikasikan aspek E (enhance) dalam DRIVE Framework.

```
# -------- E: Enhance ----------
# Phase 1 done.
# Need improvement:
# 1. Handling all test cases in the program
# 2. Improve program logic
#   - if input is empty "", output = 0
#   - rule: only check leaked password & common words if password length > 6
```

```
import string

# little syntax spoiler for future chapter: for now, just follow

def check_password_strength(input): # expected input: string, otherwise erro
    score = 0

    length_password = len(input)

    if length_password <= 0 :
        return score

    max_length_score = 3
    score += min(int(length_password / 4), max_length_score)

    has_uppercase = any(char.isupper() for char in input)
    score += int(has_uppercase)

    has_lowercase = any(char.islower() for char in input)
    score += int(has_lowercase)

    has_digits = any(char.isdigit() for char in input)
    score += int(has_digits)

    has_special_chars = any(char in string.punctuation for char in input)
    score += int(has_special_chars) * 2

    limit_compare_password = 6
    if length_password > limit_compare_password:
        common_words = ["pass", "password", "admin", "user", "me"]
        not_common_words = not any(w1 in input.lower() for w2 in common_word
        score += int(not_common_words)

        leaked_passwords = ["123456", "pass", "password", "admin", "qwerty"]
        not_leaked_words = not any(w1 in input for w2 in leaked_passwords)
        score += int(not_leaked_words)

    return score
```

```python
# ------ Enhance: Run Test Cases --------

test_cases = [
    ("password",3),
    ("p4ssword",6),
    ("passw0rd",4),
    ("P4ssw0rd",7),
    ("verylongandsecure",6),
    ("S3curedPa$$!!",10),
    ("verylongANDs3cur3dpa$$!!",10),
    ("",0),
    ("p",1),
    ("p4$$",5),
    ("123",1),
    ("1234567890",4),
    (999, 1) # challenge: handle it
] # list input-output pair

# How to test
# wrap as function (chapter 7)
# call function with input, compare output
# loop through all test cases
# wrap in try...except logic

try:
    test_success = 0
    test_total = 0
    for test_input, expected_output in test_cases:
        program_output = check_password_strength(test_input)
        test_total += 1
        if program_output == expected_output:
            print(f"test case {test_input} --> {expected_output} valid.")
            test_success += 1
        else:
            print(f"test case {test_input} --> {expected_output} invalid.")

    print(f"{test_success} test success out of {test_total}")
except Exception as e:
    print("Some error happened...")
```

Extra Challenge

- Dalam proses pemeriksaan test case, ubah iterasi `for ...` menjadi `while`
- perbaiki program untuk handling input non-string

Referensi

- Dokumentasi Python 3: https://docs.python.org/3/tutorial/controlflow.html
- https://syahrulhamdani.github.io/data-scientist-starter-kit/python/control-flow/index.html