# Execute, Debug Repeat

## *Part* 2: Debug Program

Setelah membuat program, saatnya validasi apakah hasilnya sesuai dengan ekspektasi.

Caranya adalah membuat serangkaian *test case*, yang terdiri dari pasangan input-output.

## Reminder: DRIVE Framework

* **D**efine: tentukan input-output-proses program
* **R**efine: jabarkan *step-by-step* logika proses
* **I**mplement/**I**terate: tulis code sampai program jalan tanpa error
* **V**alidate: cek kembali apakah input-output sudah benar untuk berbagai kasus
* **E**nhance: program sudah oke? lanjut. Belum oke? perbaiki lagi.

Bagian ini berfokus pada aspek **V**alidate dan **E**nhance.

## Praktik: *Password Strength Checker* (lanjutan...)

Membuat program untuk cek apakah password kuat/lemah.

Buat test case :

* "password" --> 3
* "p4ssword" --> 6
* "passw0rd" --> 4
* "P4ssw0rd" --> 7
* "verylongandsecure" --> 6
* "S3curedPa$$!!" --> 10
* "verylongANDs3cur3dpa$$!!" --> 10
* "" (kosong/empty) --> 0
* "p" --> 3
* "p4$$" --> 7
* "123" --> 2
* "1234567890" --> 4

Hasil koding pertama bisa jadi tidak memberikan hasil sesuai ekspektasi. Dalam *real-life*, itu hal yang wajar.

Saatnya untuk melakukan *debug*, dan memperbaiki proses dalam program.

# password strength checker

# -------- D: Define ----------

# input: password input (string)

# process: analyze password to best practice an score its strength

# output: scoring from 1 - weakest to 10 - strongest (int)

# -------- R: Refine ----------

import string

password\_input = "S3curedPa$$!!"

score = 0

# password length - 3 points

# more refine: length 0-3: 0 point, 4-7: 1 point, 8-12: 2 points, >12 : 3 points

length\_password = len(password\_input)

max\_length\_score = 3

score += min(int(length\_password / 4), max\_length\_score)

print(f"score after checking length: {score}") # quick and dirty debug, raises security concern

# password has uppercase - 1 point

has\_uppercase = any(char.isupper() for char in password\_input) # check if each char is uppercase

score += int(has\_uppercase) # add to score true = 1, false = 0

print(f"score after checking uppercase: {score}")

# password has lowercase - 1 point

has\_lowercase = any(char.islower() for char in password\_input)

score += int(has\_lowercase)

print(f"score after checking lowercase: {score}")

# password has number - 1 point

has\_digits = any(char.isdigit() for char in password\_input)

score += int(has\_digits)

print(f"score after checking digits: {score}")

# password has special character - 2 point

# char is special for each letters checked

has\_special\_chars = any(char in string.punctuation for char in password\_input)

score += int(has\_special\_chars) \* 2

print(f"score after checking special chars: {score}")

# password has no common words - 1 point

common\_words = ["pass", "password", "admin", "user", "me"]

not\_common\_words = not any(word in password\_input.lower() for word in common\_words)

score += int(not\_common\_words)

print(f"score after checking common words: {score}")

# password is not in list of leaked passwords - 1 point

leaked\_passwords = ["123456", "pass", "password", "admin", "qwerty", "yahoo"]

not\_leaked\_words = not any(word in password\_input for word in leaked\_passwords)

score += int(not\_leaked\_words)

print(f"score after checking leaked passwords: {score}")

print("password input:")

print(password\_input)

print("password score out of 10:")

print(score)

# ------ Validate input-output --------

# password --> 3

# p4ssword --> 2 + 1 + 1 + 1 + 1 = 6 X

# passw0rd --> 2 + 1 + 1 = 4 OK

# P4ssw0rd --> 2 + 1 + 1 + 1 + 1 + 1 = 7 X

# verylongandsecure --> 3 + 1 + 1 + 1 = 6 OK

# S3curedPa$$!! --> 3 + 1 + 1 + 1 + 2 + 1 + 1 = 10 OK

# verylongANDs3cur3dpa$$!! --> 3 + 1 + 1 + 1 + 2 + 1 + 1 = 10 OK

# "" (kosong/empty) --> 0 X --> bug

# "p" --> 1 + 1 + 1 = 3 OK

# "p4$$" --> 1 + 1 + 1 + 2 + 1 + 1 = 7 OK

# "123" --> 1 + 1 = 2 OK

# "1234567890" --> 2 + 1 + 1 = 4 OK

## *Extra Challenge*

1. Tambahkan lagi test case untuk menguji kendalan program

* input selain string: list, dictionary, int
* input "admin\_admin\_admin"
* input emoji: "⚠️emoji✅"

1. Kembangkan program untuk memberikan skor special character berdasarkan banyaknya

* 1-2: 1 poin
* >2: 2 poin

## Referensi

* Cara debug di VS Code: <https://learn.himpasikom.id/debugging-python-code>