

Lembar Laporan UAS

(nama matakuliah)
(nama dosen pengampu)



Program Studi Teknik Informatika

Semester Ganjil Tahun Ajaran 2025/2026

Tanggal Pengumpulan

The Logic Loopers
Fadhil Ariq Adi (251552010014)
M.IhsanAdzikra (NIM)
Dika Aldiansyah ()
Ikhwan Ahmad Yasin ()

Sekolah Tinggi Manajemen Informatika dan Komputer TAZKIA Jl. Raya Dramaga Blok Radar Baru No.8, RT.03/RW.03, Margajaya, Kec. Bogor Baru.,
Kota Bogor, Jawa Barat 16116, Indonesia

1. Daftar Isi

Daftar Isi.....

1. Abstrak.....

2. Pendahuluan.....

- 1.1. Latar Belakang, berisi narasi tentang mengapa project ini dibuat baik dari kebutuhan akademis maupun kebutuhan peningkatkan pengetahuan.....
- 1.2. Tujuan Project, berisi narasi deskriptif tentang alasan apa yang ingin dicapai dalam pelaksanaan project ini.....
- 1.3. Rumusan Masalah, berisi rangkaian deskriptif rumusan permasalahan apa yang ditemui dan hipotesa awal dari jawaban terhadap rumusan masalah ini.....
- 1.4. Manfaat Proyek, tuangkan manfaat apa yang didapat dari pelaksanaan proyek ini baik dari sisi individu, akademis maupun aspek lain.....
- 1.5. Ruang Lingkup, jelaskan ruang lingkup apa saja yang tercakup dalam project yang dilakukan termasuk bahasa pemrograman, metode dan teknik pemrograman yang dicapai, output yang dihasilkan.....

2. Tinjauan Pustaka.....

3. Metodologi.....

- 3.1. Analisa dan Design, berisi analisa kebutuhan yang menggambarkan masalah yang dihadapi serta desain untuk fitur-fitur apa saja yang akan dibutuhkan atau dibuat.....
- 3.2.1. Pseudocode, tuangkan semua pseudocode dan algortima yang akan digunakan selama proses pembuatan project ini.....
- 3.2.2. Diagram Alir, tuangkan diagram alir atau flowchart yang sesuai bukan dalam bentuk diagram organisasi. Gunakan simbol dan komponen yang sesuai dengan.....
- 3.3. Langkah pengembangan, berisi narasi proses pengembangan aplikasi mulai dari analisa kebutuhan, hipotesa, design, pengembangan, pengujian dan lain sebagainya. Gunakan diagram atau alat bantu lain yang dibutuhkan dalam penjelasan proses pengembangannya.....
- 3.4. Desain Antar Muka (jika ada).....

4. Hasil dan Pembahasan.....

- 4.1. Bahasa Pemrograman dan Tools, jelaskan argumentasi, alasan analisa dari bahasa Bahasa pemrograman yang digunakan dalam hal ini adalah Python serta Library, modul atau framework tambahan (jika ada).....
- 4.2. Source Code / Program, tampilkan kode program utama (bisa dilampirkan dalam bentuk file terpisah). Jelaskan bagian-bagian kode yang penting, terutama yang berkaitan dengan.....
- 4.3. Pembahasan,.....
- 4.3.1. Test Case.....
- 4.3.2. Hasil Pengujian, berisi analisa terhadap hasil pengujian yang dibandingkan dengan hipotesis awal atau ekspektasi awal. Tuangkan analisa jika ada hasil pengujian tidak sesuai dengan ekpektasi Gunakan tabel sebagai bantuan untuk membantu dalam menuangkan hasil pengujian sebagai contoh terlihat dibawah ini :.....

5. Kesimpulan.....

6. Daftar Pustaka.....

7. Lampiran.....

2.

1. Abstrak

Proyek ini bertujuan untuk menciptakan sebuah permainan tebak angka interaktif yang dikembangkan menggunakan bahasa pemrograman Python. Permainan ini mengandalkan konsep rekursi sebagai metode utama. Tujuan dari game ini adalah peserta akan berusaha menebak angka acak antara 1 hingga 100, di mana mereka akan diberikan petunjuk dan text quote random agar semangat menyelesaikan game python ini. Program di design untuk mencapai kepuasan penggunaan seperti statistik permainan,sistem hint yang dinamis, dan tampilan visual terminal yang menggunakan ASCII dan emoji. Lebih detail, program ini berhasil mengimplementasikan program yang aman dari stack overflow, percabangan, perulangan dalam satu system yang menyenangkan.

2. Pendahuluan

2.1. Latar Belakang,

Proyek ini bertujuan untuk menciptakan sebuah permainan tebak angka interaktif yang dikembangkan menggunakan bahasa pemrograman Python. Permainan ini mengandalkan konsep rekursi sebagai metode utama. Tujuan dari game ini adalah peserta akan berusaha menebak angka acak antara 1 hingga 100, di mana mereka akan diberikan petunjuk dan text quote random agar semangat menyelesaikan game python ini. Program di design untuk mencapai kepuasan penggunaan seperti statistik permainan,sistem hint yang dinamis, dan tampilan visual terminal yang menggunakan ASCII dan emoji. Lebih detail, program ini berhasil mengimplementasikan program yang aman dari stack overflow, percabangan, perulangan dalam satu system yang menyenangkan.

2.2. Tujuan Project,

Tujuan dari project ini adalah untuk meingimplementasikan konsep dari rekursi yang telah dipelajari dalam penyelesaian masala iteratif, hal ini dengan membuat sebuah system game sederhana menggunakan python yang menerapkan struktur data array untuk pencatatan hisotry, serta percabangan dan validsi input. Hal ini dipercaya akan membuat pengalaman permainan lebih baik dan dopamin pengguna meningkat.

2.3. Rumusan Masalah,

Permasahan yang berhasil di identifikasi dalam project ini meliputi masalah rekursif yang harus menggantikan perulangan / loop iteratif dalam game tebak angka, hal ini meliputi system yang harus dirancang untuk menagani riwayat tebakan pengguna, struktur data array, dan handling kasus error

Hipotesa awal dari permasalahan ini adalah rekursi dapat digunakan secara efektif untuk menggantikan loop dalam game berbasis turn. Penggunaan array diperhitungkan dapat menyimpan semua riwayat tebakan untuk analisis statistik history pengguna, penerapan sistem hint berbasis perhitungan jarak dapat membantu pemain menebak lebih cepat, serta validasi input untuk mencegah jika ada crash pada program, dan tampilan penggunaan karakter ASCII.

2.4. Manfaat Proyek

Dari sisi tim kelompok, proyek ini meningkatkan pemahaman lebih dalam mengenai konsep rekursi dan implementasi penggunaanya, yang meliputi: mengasah kemampuan problem-solving dalam pemrograman, melatih kemampuan debugging dan error handling. Dari sisi akademis, proyek ini dapat menjadi portfolio praktik, serta sebagai dasar fundamentals untuk mahasiswa untuk dapat pengembangan proyek yang lebih kompleks di masa depan. Secara umum, proyek ini mengembangkan kemampuan merancang user experience yang baik dan meningkatkan kreativitas dalam pemrograman console-based application.

2.5. Ruang Lingkup

Ruang lingkup proyek ini mencakup penggunaan bahasa pemrograman Python 3.x sebagai bahasa utama. Kami menggunakan Metode dan teknik pemrograman meliputi fungsi rekursi sebagai metode utama iterasi, struktur data array/list untuk penyimpanan riwayat, percabangan untuk validasi dan kontrol alur, exception handling untuk penanganan error. Hal ini membuat output yang dihasilkan berupa lebih menarik dengan sistem hint dinamis berdasarkan jarak tebakan, statistik permainan, serta opsi mengulang permainan

3. Tinjauan Pustaka

Rekursi adalah teknik pemrograman di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan masalah yang lebih kecil dari masalah utama. Menurut Cormen et al. dalam "Introduction to Algorithms", rekursi efektif digunakan ketika masalah dapat dibagi menjadi sub-masalah yang serupa dengan masalah asal. Dalam konteks game tebak angka, setiap percobaan tebakan dapat dianggap sebagai sub-masalah yang identik, sehingga rekursi menjadi pilihan yang tepat.

Struktur data array atau list dalam Python merupakan koleksi terurut yang dapat menyimpan multiple values. Menurut dokumentasi Python, list bersifat mutable dan dapat digunakan untuk menyimpan data dinamis. Dalam proyek ini, array digunakan untuk menyimpan riwayat semua tebakan pemain, yang kemudian dianalisis untuk memberikan statistik seperti nilai minimum, maksimum, dan rata-rata.

4. Metodologi

4.1. Analisa dan Design

Kalau kita lihat dari analisa kebutuhan, game tebak angka membutuhkan beberapa fitur penting. seperti harus ada sistem yang bisa membuat angka random antara 1 sampai 100 sebagai target. dimana, pemain harus memasukkan tebakan mereka, dan sistem mesti akan melakukan check-in terlbih dahulu apakah angkanya valid dan masuk dalam rentang angka atau tidak. System didesign untuk memberikan feedback jika tebakan terlalu besar atau kecil serta plus seberapa jauh jaraknya dari target. Semua tebakan akan dicatat didalam array, untuk dianalisis secara statistik. Dari sisi lainnya , tampilan game sebaiknya lebih visual untuk menari minat pengguna , misalnya karakter ASCII, emoji, dengan formatting yang tampak lebih human-desgin. Terakhir, setelah satu ronde selesai, pemain dapat memilih lanjut atau stop dari permainan.

4.2. Desain Sistem

4.2.1. Pseudocode

```
FUNCTION print_header():
    PRINT header dengan ASCII art dan judul game
    PRINT instruksi singkat

FUNCTION print_hint(tebakan, target):
    CALCULATE selisih = |tebakan - target|
    IF tebakan < target THEN
        PRINT "Terlalu kecil"
        IF selisih > 30 THEN PRINT "Masih jauh"
        ELSE IF selisih > 15 THEN PRINT "Lumayan jauh"
        ELSE PRINT "Sudah dekat"
    ELSE
        PRINT "Terlalu besar"
        (sama seperti di atas)

FUNCTION print_victory(target, jumlah_percobaan, riwayat):
    PRINT ASCII art "VICTORY"
    PRINT jumlah percobaan
    CALCULATE rating berdasarkan jumlah percobaan
    PRINT riwayat tebakan dalam format tabel
    CALCULATE dan PRINT statistik (min, max, rata-rata)

FUNCTION tebak_rekursif(target, percobaan, riwayat):
    TRY:
        PRINT header percobaan
        INPUT tebakan dari user

        IF tebakan < 1 OR tebakan > 100 THEN
            PRINT error
            CALL tebak_rekursif(target, percobaan, riwayat)
            RETURN

        ADD tebakan ke riwayat

        IF tebakan == target THEN
            CALL print_victory(target, percobaan, riwayat)
            RETURN
        ELSE
            CALL print_hint(tebakan, target)
            CALL tebak_rekursif(target, percobaan+1, riwayat)

    CATCH ValueError:
        PRINT error "Input harus angka"
        CALL tebak_rekursif(target, percobaan, riwayat)

    CATCH KeyboardInterrupt:
        PRINT goodbye message
        RETURN

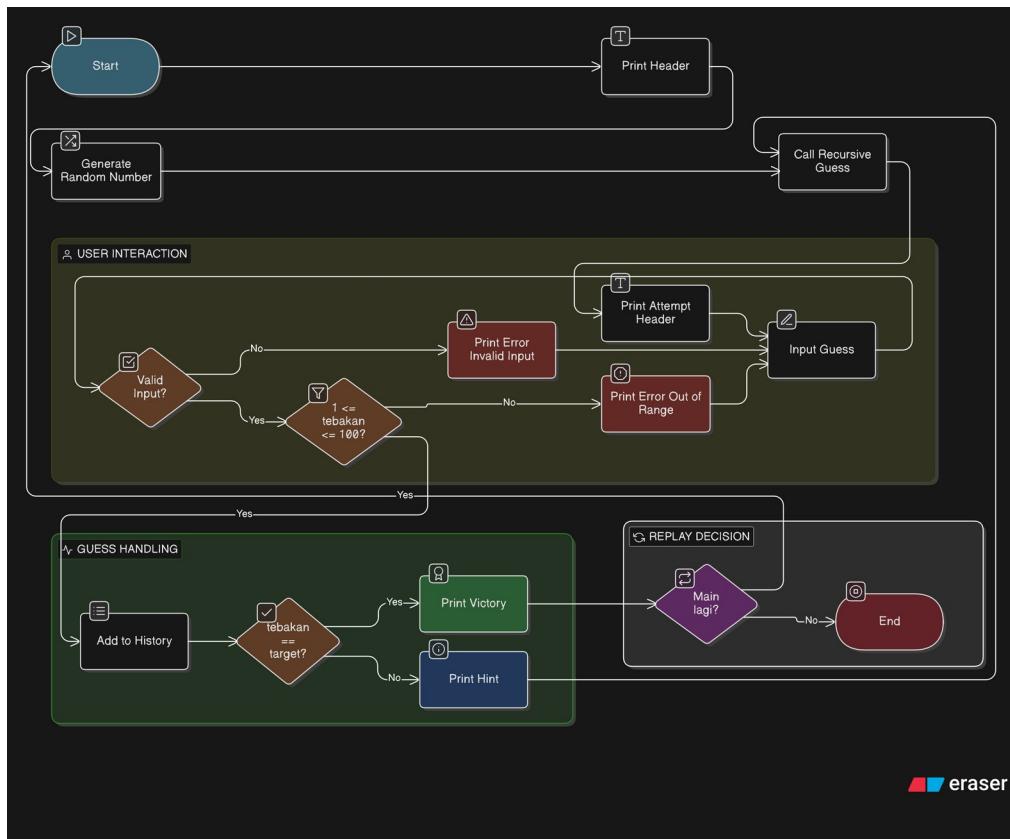
FUNCTION play_game():
    CALL print_header()
    PRINT random fun fact

    target = RANDOM(1, 100)
    CALL tebak_rekursif(target, 1, [])

    INPUT main_lagi
    IF main_lagi == "y" THEN
        CALL play_game()
    ELSE
        PRINT goodbye message

MAIN:
TRY:
    CALL play_game()
CATCH Exception:
    PRINT error message
```

4.2.2. Diagram Alir,



4.3. Langkah pengembangan

Pengembangan aplikasi ini berjalan lewat serangkaian tahap yang jelas. Pertama, analisa kebutuhan. Di sini, tim kelompok berdiskusi untuk mengidentifikasi fitur yang harus ada: generate random number, validasi input, sistem hint, dan pelacakan statistik. Setelah itu kami, masuk ke tahap desain. Masing-masing anggota kelompok akan berbagi tugas untuk mulai menyusun struktur fungsi yang modular, alur rekursi, dan tampilan visualnya.

- 4.4. Tahapan berikutnya, implementasi. Anggota kelompok yang bertugas akan membangun berbagai fungsi utility untuk tampilan, kemudian dilanjutkan oleh anggota lainnya untuk mengerjakan fungsi rekursi utama, menambahkan exception handling, dan mempercantik output terminal dengan ASCII art dan emoji. Saat testing system, dilakukan beberapa proses debugging oleh Tim yang bertugas menangani error, dan mengetes system apakah system penting sudah valid, seperti input yang tidak valid, kasus-kasus seperti overflow, sampai alur pengalaman pengguna. Terakhir, tahap refinement. Di sini, tim mulai membuat tampilan visual yang lebih baik, mengoptimalkan sistem hint, menambah statistik, dan menghadirkan fitur main ulang. Setiap tahap saling

melengkapi, memastikan aplikasi tidak hanya berfungsi, tapi juga nyaman digunakan.

4.5. Desain Antar Muka

Antar muka dirancang khusus di terminal dengan prinsip visual hierarchy menggunakan karakter box-drawing, emoji, spacing dan divider yang konsisten. Komponen utama interface meliputi header dengan ASCII art "TEBAK ANGKA GAME", attempt header dengan nomor percobaan, input prompt dengan emoji, feedback hint dengan simbol visual, victory screen dengan ASCII art "VICTORY", serta statistik tabel riwayat tebakan, dan prompt pertanyaan main ulang.

5. Hasil dan Pembahasan

5.1. Bahasa Pemrograman dan Tools,

Proyek ini memakai Python versi 3.14.2 sebagai bahasa pemrograman utama. Python juga mendukung rekursi dengan baik, meski ada batasan recursion depth default di angka 1000. Library bawaannya, seperti modul random, sudah cukup untuk kebutuhan proyek, sehingga ketergantungan untuk eksternal dependencies tidak diperlukan. Selain itu, sistem exception handling dari python sudah solid dengan try-except block, dan dukungan Unicode-nya memungkinkan kita pakai emoji atau karakter khusus untuk mempercantik tampilan.

Project ini, hanya menggunakan random module dari standard library yang dipakai untuk menghasilkan angka acak. Framework tambahan tidak diperlukan karena aplikasinya berbasis console/ terminal interface dan strukturnya sederhana.

5.2. Source Code

link : <https://github.com/Fadhil-Ariq-Id/uasPyhton2026/blob/main/la.py>

Kode program terdiri dari beberapa fungsi utama dengan penjelasan sebagai berikut.

a. Rekursi: Implementasi utama rekursi ada pada fungsi `tebak_rekursif()`. Fungsi ini memanggil dirinya sendiri dalam dua kondisi, yaitu ketika input tidak valid dan invalid. Base case dari rekursi adalah ketika tebakan sama dengan target, atau ketika terjadi KeyboardInterrupt. Contoh:

```
def tebak_rekursif(number_rahasia, percobaan=1, riwayat=None):
    # ... kode validasi ...
    if tebak == number_rahasia:
        print_victory(number_rahasia, percobaan, riwayat)
        return # Base case
    else:
        print_hint(tebak, number_rahasia)
        tebak_rekursif(number_rahasia, percobaan + 1, riwayat) # Recursive
call
```

b. Percabangan: Struktur if-else digunakan secara ekstensif untuk validasi input dan kontrol alur program. Contoh pada fungsi `print_hint()`:

```
if tebak < number_rahasia:
    diff = number_rahasia - tebak
    if diff > 30:
        print("Masih jauh banget")
    elif diff > 15:
        print("Lumayan jauh")
    else:
        print("Udah deket")
```

Percabangan juga digunakan untuk validasi range input dan menentukan rating pemain.

C. Array: List Python digunakan sebagai array dinamis untuk menyimpan riwayat tebakan:

```
riwayat = []
riwayat.append(tebak) # Menambah
min(riwayat) # Analisis statistik
max(riwayat)
sum(riwayat)/len(riwayat)
```

Array ini kemudian diiterasi untuk menampilkan riwayat lengkap di victory screen.

d. Input dari User: Program menerima input pada beberapa titik menggunakan fungsi `input()` dengan validasi menggunakan exception handling:

```
try:
    tebak_str = input("Tebak angkanya (1-100): ")
    tebak = int(tebak_str)
except ValueError:
    print("Input tidak valid")
    tebak_rekursif(number_rahasia, percobaan,
    riwayat)
```

5.3. Pembahasan,

Program tebak angka ini memiliki struktur modular dengan 7 fungsi utama, yaitu `print_header()`, `print_divider()`, `print_box()`,

`print_attempt_header()`, `print_hint()`, `print_victory()`, dan `tebak_rekursif()`. Fungsi `play_game()` bertindak sebagai controller utama.

Program start dimulai dari `play_game()`, yang memanggil `print_header()`. Setelah itu, `tebak_rekursif()` berjalan secara rekursif, terus memanggil dirinya sendiri sampai pemain berhasil menebak atau base case terpenuhi. Fungsi-fungsi utilitas seperti `print_hint()` dan `print_victory()` muncul sesuai kebutuhan, jadi pemain selalu dapat feedback yang pas di setiap langkah.

Seluruh tim kelompok sudah menguji program ini dengan berbagai skenario, dan hasilnya solid—program tetap stabil dalam berbagai kondisi. Target utama proyek ini, yaitu mengganti loop tradisional dengan rekursi, tercapai dengan mulus. Fungsi `tebak_rekursif()` berhasil menggantikan while loop dengan konsep pemanggilan diri sendiri untuk setiap percobaan.

Sebelumnya ada beberapa kendala, terutama risiko stack overflow jika pemain terus gagal menebak. Tapi, dengan range angka 1-100 dan sistem hint yang cukup membantu, rata-rata pemain sudah bisa menebak dalam 6-7 kali. Itu jauh di bawah batas recursion default Python, yang 1000. Solusi alternatif adalah meningkatkan recursion limit menggunakan `sys.setrecursionlimit()`.

Tim kelompok berhasil mengimplementasi rekursi yang bersih, tanpa side effect, sistem hint yang adaptif untuk pengguna, tampilan visual terminal yang menarik, serta exception handling yang baik.

Kedepanya beberapa hal dapat ditingkatkan mulai dari level kesulitan dengan range yang berbeda, implemnetasi system untuk scoring dan level global dalam game, penyimpanan local database seperti mysql atau lite-sql untuk penyimpanan highscore dan penambahan system ads yang dapat dihubungkan dengan google AdMob , AdSense, dan pihak iklan ketiga seperti hoeygain atau monetag.

5.3.1. Test Case

Berikut adalah test case yang telah dirancang untuk menguji berbagai aspek program:

Test Case 1: Input Valid dalam Range

- Skenario: User memasukkan angka 50 ketika target adalah 75
- Expected: Program menerima input, menampilkan hint "terlalu kecil", dan melanjutkan ke percobaan berikutnya

Test Case 2: Input Valid - Tebakan Benar

- Skenario: User menebak angka yang sesuai dengan target -
- Expected: Program menampilkan victory screen dengan statistik lengkap

Test Case 3: Input Invalid - Bukan Angka

- Skenario: User memasukkan string "abc"
- Expected: Program menampilkan error "Input tidak valid" dan meminta input ulang tanpa menambah counter percobaan

Test Case 4: Input Invalid - Di Luar Range

- Skenario: User memasukkan angka 150
- Expected: Program menampilkan warning "Angka harus antara 1-100" dan meminta input ulang

Test Case 5: Input Invalid - Angka Negatif

- Skenario: User memasukkan -5
- Expected: Program menampilkan warning dan meminta input ulang

Test Case 6: Keyboard Interrupt

- Skenario: User menekan Ctrl+C
- Expected: Program keluar dengan graceful dengan menampilkan angka rahasia

Test Case 7: Hint System - Jauh (>30)

- Skenario: Tebakan berjarak >30 dari target
- Expected: Hint menampilkan "Masih jauh banget"

Test Case 8: Hint System - Sedang (15-30)

- Skenario: Tebakan berjarak 15-30 dari target

Expected: Hint menampilkan "Lumayan jauh"

Test Case 9: Hint System - Dekat (<15)

- Skenario: Tebakan berjarak <15 dari target
- Expected: Hint menampilkan "Udah deket"

Test Case 10: Main Ulang – Yes

- Skenario: User memilih "y" setelah game selesai
- Expected: Game restart dengan angka random baru

Test Case 11: Main Ulang – No

-Skenario: User memilih "n" setelah game selesai

- Expected: Program menampilkan goodbye message dan exit

Test Case 12: Statistik Riwayat

- Skenario: Setelah 5 percobaan dengan nilai [30, 60, 45, 52, 50]

- Expected: Program menampilkan min=30, max=60, rata-rata=47.4

5.3.2. Hasil Pengujian, berisi analisa terhadap hasil pengujian yang dibandingkan dengan hipotesis awal atau ekspektasi awal. Tuangkan analisa jika ada hasil pengujian tidak sesuai dengan ekspektasi. Gunakan tabel sebagai bantuan untuk membantu dalam menuangkan hasil pengujian sebagai contoh terlihat dibawah ini :

No	Input	Output yang diharapkan	Hasil Aktual	Keterangan
1	Angka: 50 (target: 75)	"Terlalu kecil", hint +25	Sesuai ekspektasi	Berhasil
2	Angka: 75 (target: 75)	Victory screen dengan statistik	Sesuai ekspektasi	Berhasil

Analisis Hasil Pengujian:

Semua test case berhasil dilalui sesuai dengan hipotesis awal. Sistem telah tervalidasi baik untuk menangani berbagai jenis error input. serta beberapa sistem penting seperti : hint adaptif dalam feedback yang akurat berdasarkan jarak tebakan, rekursi berfungsi dengan sempurna tanpa stack overflow untuk range percobaan normal, exception handling mencegah crash program pada semua skenario error yang diuji. Statistik yang ditampilkan akurat. Tidak ditemukan hasil pengujian yang tidak sesuai ekspektasi. Program berjalan dengan stabil

6. Kesimpulan

Proyek game tebak angka berbasis rekursi telah memenuhi semua tujuan awal yang sudah ditetapkan. Implementasi rekursif sebagai pengganti iterasi berhasil diterapkan dengan baik pada fungsi `tebak_rekursif()` yang jadi main core logic dari game. Konsep-konsep dasar pemrograman seperti percabangan, array hingga exception handling terhubung dengan fungsional.

Sistem hint adaptif juga dibuat agar efektif terhadap pemain dalam menebak angka dengan lebih cepat. Penggunaan array untuk fitur tracking riwayat tebakan memberikan insight menarik melalui statistik yang ditampilkan di akhir permainan. Tampilan visual yang dikembangkan dengan ASCII art dan emoji berhasil menciptakan user experience yang menyenangkan meskipun terbatas pada console interface.

Dari sisi pembelajaran, project ini memberikan pemahaman mendalam tentang cara kerja rekursi, pentingnya base case, dan bagaimana parameter dilewatkannya antar pemanggilan rekursif. Exception handling yang comprehensive mengajarkan pentingnya defensive programming untuk menciptakan aplikasi yang robust (kode yang sehat).

7. Daftar Pustaka

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press.
- Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media.
- Python Software Foundation. (2024). *Python Documentation: Built-in Functions*. Retrieved from <https://docs.python.org/3/library/functions.html>
- Python Software Foundation. (2024). *Python Documentation: Random Module*. Retrieved from <https://docs.python.org/3/library/random.html>
- Zelle, J. M. (2016). *Python Programming: An Introduction to Computer Science* (3rd ed.). Franklin, Beedle & Associates.
- Matthes, E. (2019). *Python Crash Course* (2nd ed.). No Starch Press.
- Downey, A. B. (2015). *Think Python: How to Think Like a Computer Scientist* (2nd ed.). O'Reilly Media.

Lampiran

A: Kode Program Lengkap

potong bagi 2 disini

output
=>

```
towea@fedora:~/Downloads/develop/uasPython$ py la.py
```

```
=====
```

```
🎮 TEBAK ANGKA GAME 🎮
```

```
=====
```

```
✨ Game super simple: Tebak angka antara 1-100! ✨
```

```
=====
```

```
💡 Fun fact: Lucky number-mu adalah 7! 🌸
```

```
=====
```

```
◉ PERCOBAAN KE-1 ◉
```

```
◉ Tebak angkanya (1-100): 12
```

```
✗ ▼▼▼ TERLALU KECIL! ▼▼▼
```

```
💡 Hint: Coba angka yang LEBIH BESAR!
```

```
★ Udah deket! (+4)
```

```
=====
```

```
◉ PERCOBAAN KE-2 ◉
```

```
◉ Tebak angkanya (1-100): 232
```

```
⚠️ Angka harus antara 1-100!
```

```
=====
```

```
◉ PERCOBAAN KE-2 ◉
```

```
◉ Tebak angkanya (1-100): 12
```

```
✗ ▼▼▼ TERLALU KECIL! ▼▼▼
```

```
💡 Hint: Coba angka yang LEBIH BESAR!
```

```
★ Udah deket! (+4)
```

```
=====
```

```
◉ PERCOBAAN KE-3 ◉
```

```
◉ Tebak angkanya (1-100): ^C
```

```
👋 Keluar dari game. Bye bye!
```

```
💔 Angka rahasianya adalah: 16
```

```
MAIN LAGI?
```

```
🎮 Main lagi? (y/n): n
```

```
*****
```

```
TERIMA KASIH SUDAH BERMAIN!
```

```
Sampai jumpa lagi! 🙋
```

```
Keep gaming, keep smiling!
```

```
*****
```