

LAPORAN AKHIR
TEORI BAHASA FORMAL DAN OTOMATA

“Pengembangan Mesin Turing untuk Verifikasi Palindrom pada String”



Dosen Pengampu : Nurul Hayaty, S.T., M.Cs.

Kelompok 2 :

Alvira Angraini	(2201020046)
Grachiella Dian Malemukur	(2201020045)
Mohd Allifiyan Baitul Nesam	(2201020023)
Aunurrafiq R	(221055201091)
Novatul Hidayah	(221055201040)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN TEKNOLOGI KEMARITIMAN
UNIVERSITAS MARITIM RAJA ALI HAJI
TANJUNG PINANG-KEPULAUAN RIAU
TA 2023

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era modern ini, pengolahan informasi menjadi bagian integral dari kehidupan sehari-hari, terutama dengan meningkatnya ketergantungan pada teknologi informasi. Salah satu aspek penting dari pengolahan informasi adalah manipulasi dan verifikasi teks atau string. Mesin Turing, yang diperkenalkan oleh Alan Turing pada tahun 1936, telah menjadi dasar bagi banyak pengembangan dalam teori komputasi dan pengolahan bahasa algoritma.

Palindrom adalah sebuah pola yang menarik dalam dunia komputasi dan matematika. Sebuah string dikatakan palindrom jika dapat dibaca dengan cara yang sama dari kiri ke kanan maupun sebaliknya, tanpa memperhatikan spasi, tanda baca, atau huruf besar-kecil. Penerapan pengembangan Mesin Turing untuk verifikasi palindrom pada string menjadi relevan mengingat banyaknya aplikasi praktis dari konsep ini dalam pemrosesan dan validasi data.

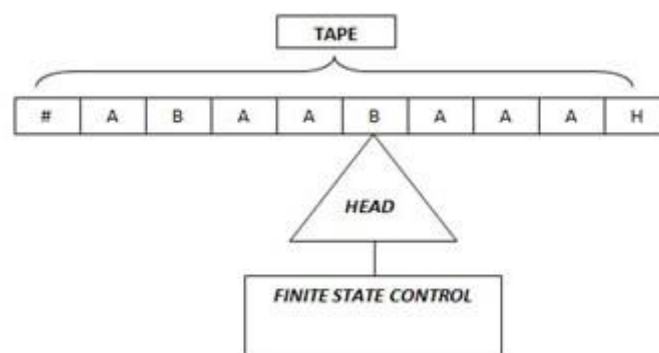
Dalam latar belakang ini, kami tertarik untuk membuat mesin sederhana yang dapat memverifikasi apakah suatu string merupakan palindrom atau tidak. Meskipun konsep ini tampak sederhana, implementasinya melibatkan penggunaan dasar pemrograman dan logika algoritma. Pentingnya pengembangan mesin palindrom adalah sebagai dasar untuk pemahaman algoritma string dan kemampuan pengolahan data dasar. Proyek ini memberikan peluang bagi pengembang pemula untuk memahami bagaimana sebuah mesin sederhana dapat memberikan output yang bermakna berdasarkan aturan sederhana.

Laporan ini akan mencakup langkah-langkah dasar pengembangan mesin palindrom, serta contoh aplikasi sederhana. Kami berharap laporan ini dapat memberikan pemahaman praktis tentang konsep palindrom dan kemampuan dasar dalam pengembangan mesin sederhana.

1.2 Landasan Teori

a. Konsep Mesin Turing

Mesin Turing adalah model komputasi teoritis yang dikemukakan oleh Alan Turing, secara esensial, mesin Turing adalah sebuah finite automaton yang memiliki sebuah pita dengan panjang tak terhingga yang dapat membaca dan menulis data. Mesin Turing menggunakan notasi seperti instantaneuous descriptions (ID) pada Push Down Automata (PDA) untuk menyatakan konfigurasi dari komputasinya. Stack pada PDA memiliki keterbatasan akses. Elemen yang dapat diakses hanya elemen yang ada pada top stack. Pada Mesin Turing, memori akan berupa suatu pita yang pada dasarnya merupakan sebuah array dari sel-sel penyimpanan (Saepudin, 2015).



Gambar 1. Representasi Mesin Turing

Di dalam teori mesin Turing, infinite tape, head, dan finite state control merupakan komponen-komponen utama:

1. Infinite Tape (Pita Tak Terhingga) :

Berfungsi sebagai media penyimpanan informasi dalam mesin Turing. Setiap sel di pita dapat menyimpan simbol, dan pita secara teoritis tidak memiliki batas panjangnya. Pita ini memungkinkan mesin Turing untuk membaca, menulis, dan menggeser posisi head.

2. Head (Kepala) :

Fungsi Head adalah komponen yang membaca dan menulis simbol pada pita. Head juga bisa menggeser posisinya ke kiri atau ke kanan. Head berinteraksi dengan simbol pada pita dan bergerak sesuai dengan instruksi dari finite state control, memungkinkan mesin Turing melakukan komputasi.

3. Finite State Control (Kontrol Status Terbatas) :

Finite state control mengatur perilaku mesin Turing. Ini mencakup keadaan-keadaan yang dapat diambil oleh mesin dan aturan perpindahan antara keadaan-keadaan tersebut. Finite state control menentukan respons mesin terhadap simbol yang dibaca dan instruksi yang diberikan, membimbing jalannya komputasi.

Ketiga komponen ini bersama-sama memungkinkan mesin Turing untuk menjalankan komputasi universal, mengonsep dasar bagi konsep komputer dan penghitungan di dunia komputasi teoritis.

Notasi formal pada mesin turing sama dengan finite automata atau PDA. Terdiri dari 7 tuple, yaitu :

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

Komponen-komponennya adalah:

- Q: Himpunan berhingga dari state dari finite control.
- Σ : Himpunan berhingga dari simbol-simbol input.
- Γ : Himpunan dari simbol pada pita. Σ merupakan subset dari.
- δ : Fungsi transisi. Argumen (q, X) adalah sebuah state q dan sebuah simbol pada pita X . Nilai dari (q, X) , jika nilai tersebut didefinisikan, adalah triple (p, Y, D) , dimana:
 - p adalah next state dalam Q
 - Y adalah simbol, dalam Γ , ditulis dalam sel yang sedang dibaca, menggantikan simbol apapun yang ada dalam sel tersebut.
 - D adalah arah, berupa L atau R, berturut-turut menyatakan kiri atau kanan, dan menyatakan arah dimana head bergerak.

- q0: start state, sebuah anggota dari Q, dimana pada saat awal finite control ditemukan.
- B: simbol blank. Simbol ini ada dalam Γ tapi tidak dalam Σ , yaitu B bukan sebuah simbol input.
- F: Himpunan dari final state, subset dari Q.

b. Palindrom

Palindrom berasal dari bahasa Yunani, yakni palin dan dromos. Palin artinya “lagi” dan dromos artinya “arah”. Sementara itu menurut Kamus Besar Bahasa Indonesia (KBBI) Palindrom adalah rangkaian kata, atau bilangan yang terbaca sama, baik dari depan maupun dari belakang. Palindrom dapat kalian temukan di sebuah kata, frasa, kalimat hingga angka. Hal ini menarik tentunya, sebab Palindrom bisa dibacadengan sama, walau dari belakang. Palindrom dapat kalian temukan tidak hanya pada bahasa Indonesia saja.

Salah satu Palindrom yang sangat unik berasal dari bahasa latin, yaitu “Sator Arepo Tenet Opera Rotas”. Palindrom yang satu ini sangatlah unik sebab akan mengulang kalimat lagi jika kalian membentuk kata dari huruf pertama setiap kata kemudian disambung dengan huruf kedua setiap kata, dst. Apabila disusun seperti pada susunan berikut, dapat kalian baca secara vertikal maupun horizontal.

S A T O R
A R E P O
T E N E T
O P E R A
R O T A S

Berikut contoh Palindrom dalam bahasa Indonesia:

1. Kasur ini rusak
2. Kasur nababan rusak
3. Katak
4. Kodok
5. Malam
6. Radar

Sedangkan dalam bilangan untuk membuat suatu palindrome adalah sangat sederhana yaitu kita jumlahkan bilangan palindrome satu dengan kebalikan palindrome lain dan dijumlahkan terus menerus sampai terbentuk bilangan palindrome. Misalnya $94 + 49 = 143$, $143 + 341 = 484$

Bilangan palindrome juga dapat dibuat dari penjumlahan

kuadrat bilangan berurutan misalnya $11^2 + 12^2 + 13^2 = 434$, $11^2 + 12^2 + 13^2 + 14^2 + 15^2 + 16^2 = 1111$, dan sebagainya. Dalam pemrograman untuk mengetahui apakah suatu bilangan palindrome adalah dengan algoritma sederhana sebagai berikut :

- a. Masukkan bilangan
- b. Reverse bilangan tersebut
- c. Bandingkan dengan bilangan yang menjadi masukannya
- d. Jika sama maka cetak bilangan tersebut
- e. Jika tidak cetak “bukan bilangan palindrome”.

c. Bahasa Pemrograman Python

Python merupakan bahasa pemrograman komputer yang biasa dipakai untuk membangun situs, software/aplikasi, mengotomatiskan tugas dan melakukan analisis data. Bahasa pemrograman ini termasuk bahasa tujuan umum. Artinya, ia bisa digunakan untuk membuat berbagai program berbeda, bukan khusus untuk masalah tertentu saja. Karena sifatnya yang serba guna dan mudah digunakan, ia menjadi bahasa pemrograman yang paling banyak digunakan. Terutama untuk mereka yang masih pemula.

Berdasarkan survei pengembang Stack Overflow tahun 2022, Python menjadi bahasa pemrograman terpopuler keempat. Sebanyak hampir 50% dari responden mengatakan bahwa mereka menggunakan hampir setengah dari waktu kerja mereka dengan menggunakan bahasa pemrograman ini. Nama Python sendiri berasal dari Monty Python. Ketika Guido van Rossum membuatnya, dia juga sedang membaca skrip Sirkus Terbang Monty Python BBC. Menurutnya nama itu singkat dan sedikit misterius. Karena itulah, sang kreator memilih menggunakan nama tersebut untuk bahasa pemrograman yang dibuatnya itu.

BAB II

ISI

2.1 Penyelesaian Kasus

Dalam menyelesaikan kasus yang ada perlu nya langkah-langkah dibawah ini :

a. Memahami Konsep Mesin Turing

Memperoleh pemahaman konsep dasar mesin Turing sebagai model teoretis komputer, terutama pada bagian tabel transisis dan diagram automata nya.

b. Implementasi dalam Bahasa Python

Mengimplementasikan mesin Turing dalam bahasa pemrograman Python. Membuat program yang melakukan verifikasi palindrom dengan membaca input, melakukan instruksi konversi, dan melanjutkan ke langkah berikutnya.

c. Penanganan Karakter Tidak Diperbolehkan

Menangani karakter di luar daftar karakter yang diperbolehkan, termasuk simbol dan non-alfabet, Program memberikan pesan error jika karakter yang dimasukkan tidak diperbolehkan.

d. Verifikasi Palindrom

Jika string yang dimasukkan merupakan palindrom, program mencetak "Is a palindrome!" sebaliknya Jika bukan palindrom, program mengeluarkan output "is a not palindrome! "

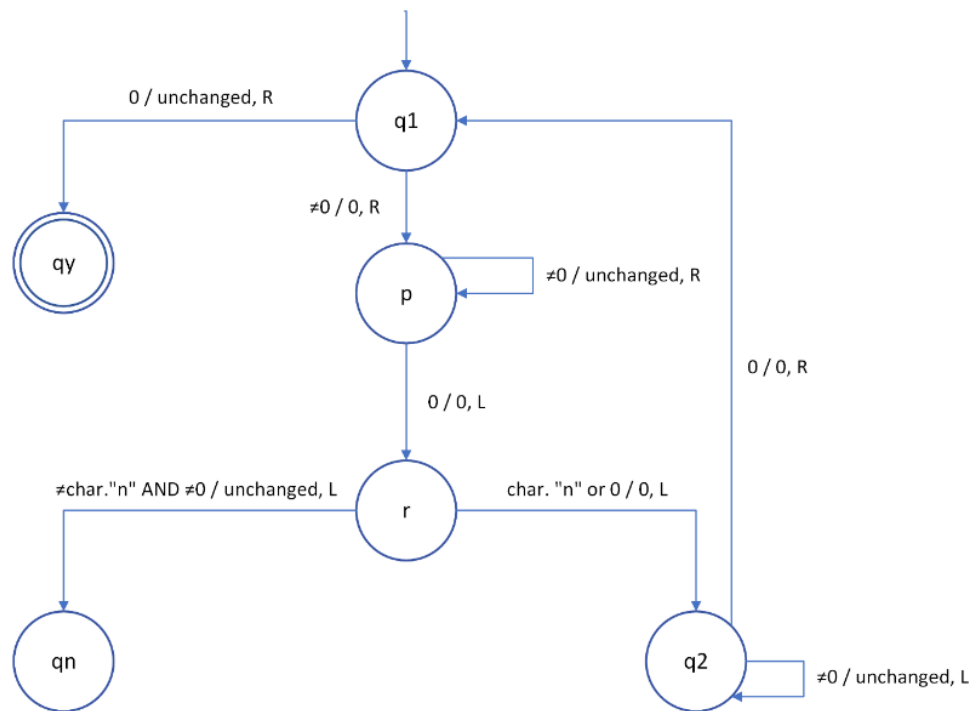
e. Pengujian dan Debugging

Melakukan pengujian menyeluruh untuk memastikan program berfungsi tanpa kesalahan serta memeriksa output program untuk berbagai input dan memastikan keakuratannya.

f. Penyusunan laporan

Penyusunan laporan dilakukan setelah semua langkah-langkah di atas diterapkan, laporan ini berisikan hal-hal yang sesuai dengan hasil dan segala hal yang berkaitan dengan judul laporan.

2.2 Diagram Automata



Gambar 2. *Diagram Automata*

Pada Gambar 2 diatas dapat di tarik beberapa kesimpulan, yaitu :

- State Awal ($q1$):
 - Simbol: Lingkaran dengan label " $q1$ ".
 - Transisi:
 - Menerima input $\neq 0$: Pindah ke State p .
 - Menerima input 0 : Pindah ke State qy .
 - Menerima input $\neq n$ dan $\neq 0$: Tidak ada transisi.
 - Menerima input n : Tidak ada transisi.
- State p :
 - Simbol: Lingkaran dengan label " p ".
 - Transisi:
 - Menerima input $\neq 0$: Tetap di State p .
 - Menerima input 0 : Pindah ke State r .
 - Menerima input $\neq n$ dan $\neq 0$: Tidak ada transisi.
 - Menerima input n : Tidak ada transisi.

- State r:
 - Simbol: Lingkaran dengan label "r".
 - Transisi:
 - Menerima input $\neq 0$: Tidak ada transisi.
 - Menerima input 0: Tidak ada transisi.
 - Menerima input $\neq n$ dan $\neq 0$: Pindah ke State qn.
 - Menerima input n: Pindah ke State q2.

- State qn:
 - Simbol: Lingkaran dengan label "qn".
 - Transisi:
 - Menerima input apa pun: Tetap di State qn.

- State q2:
 - Simbol: Lingkaran dengan label "q2".
 - Transisi:
 - Menerima input $\neq 0$: Tetap di State q2.
 - Menerima input 0: Pindah ke State q1.
 - Menerima input $\neq n$ dan $\neq 0$: Tidak ada transisi.
 - Menerima input n: Tidak ada transisi.

- State qy:
 - Simbol: Lingkaran dengan label "qy".
 - Transisi:
 - Menerima input apa pun: Tetap di State qy.

- Panah Transisi:
 - Panah dari q1 ke p dengan label input " $\neq 0$ ".
 - Panah dari q1 ke qy dengan label input "0".
 - Panah dari p ke p dengan label input " $\neq 0$ ".
 - Panah dari p ke r dengan label input "0".
 - Panah dari r ke qn dengan label input " $\neq n$ dan $\neq 0$ ".
 - Panah dari r ke q2 dengan label input "n".

- Panah dari q2 ke q2 dengan label input " $\neq 0$ ".
- Panah dari q2 ke q1 dengan label input "0".

2.3 Tabel Transisi

Tabel 1. *Tabel Transisi*

δ	$\neq 0$	0	$\neq n$ and $\neq 0$	n
q1	p	qy	-	-
p	p	r	-	-
r	-	-	qn	q2
qn	-	-	-	-
q2	q2	q1	-	-
qy	-	-	-	-

Penjelasan Tabel Transisi untuk Mesin Keadaan Terbatas Verifikasi Palindrom diatas ialah:

- Simbol dan Keadaan:
 - δ : Representasi transisi antar keadaan
 - $\neq 0$: Simbol input untuk karakter tidak sama dengan 0.
 - 0: Simbol input untuk karakter 0
 - $\neq n$ and $\neq 0$: Simbol input untuk karakter tidak sama dengan n dan tidak sama dengan 0.
 - n: Simbol input untuk karakter n.
 - q1, p, qy, r, qn, q2: Keadaan-keadaan yang mungkin merepresentasikan kondisi saat memeriksa palindrom.

- Transisi Antara Keadaan:

Dalam tabel ini, setiap sel memberikan informasi tentang bagaimana mesin berpindah dari satu keadaan ke keadaan lainnya berdasarkan input yang diberikan.

- Penjelasan Transisi:
 - Jika mesin berada di keadaan q_1 dan menerima input $\neq 0$, maka pindah ke keadaan p .
 - Jika mesin berada di keadaan q_1 dan menerima input 0 , maka pindah ke keadaan q_y .
 - Jika mesin berada di keadaan p dan menerima input $\neq 0$, maka tetap berada di keadaan p .
 - Jika mesin berada di keadaan p dan menerima input 0 , maka pindah ke keadaan r .
 - Jika mesin berada di keadaan r dan menerima input $\neq n$ and $\neq 0$, maka pindah ke keadaan q_n .
 - Jika mesin berada di keadaan r dan menerima input n , maka pindah ke keadaan q_2 .
 - dan seterusnya untuk setiap kombinasi input dan keadaan lainnya

BAB III

PEMBAHASAN

3.1 Penjelasan Program

1) Program Bagian 1

```
1 import string
2 import sys
3 from turing import Turing_Machine
4
5 def check_palindrome(initial_string):
6
7     character_list = list(string.ascii_lowercase)
8     character_list.append(' ')
9
10    print('Checking: ' + initial_string)
11    print('- - -')
12    initial_list = list(initial_string)
13
14    for i in initial_list:
15        if i not in character_list:
16            print('Error! Initial character >',i,< not in allowed character list!')
17            sys.exit()
18
19
20
21    initial_list.append(0)
22
23    i_write_head = 0
24    i_state = 'q1'
25    i_tape_list = initial_list
26
27
28    runMachine = Turing_Machine(i_state,i_write_head,i_tape_list)
29    print(runMachine.getState(),runMachine.getHead(),runMachine.getList())
30
31    # Run the machine
32    ctr=0
33    while runMachine.getState() != 'qy' and runMachine.getState() != 'qn' and ctr < 1000:
34        runMachine.updateMachine(character_list)
35        print(runMachine.getState(),runMachine.getHead(),runMachine.getList())
36        ctr += 1
37    print('- - -')
38
39
40    if (runMachine.getState() == 'qy'):
41        print(initial_string,'is a palindrome! Steps:',ctr)
42    else:
43        print(initial_string,'is NOT a palindrome! Steps:',ctr)
```

Gambar 3. Program bagian 1

Dari gambar kodingan di atas dapat di jelaskan, sebagai berikut :

- Impor Pustaka:
 - string dan sys diimpor untuk manipulasi string dan mengakses fungsi sistem.
 - Turing_Machine diimpor dari pustaka yang tidak terlihat (mungkin kode terpisah).
- 2. Inisialisasi Variabel:
 - Daftar karakter diinisialisasi dengan huruf kecil ASCII dan ditambahkan karakter kosong.
 - String input dipecah menjadi daftar karakter.

- Cek Karakter Awal:
 - Setiap karakter dalam string input diperiksa untuk memastikan hanya karakter dari daftar karakter yang diizinkan yang digunakan. Jika ditemukan karakter tidak valid, pesan kesalahan dicetak, dan program dihentikan.
- Konfigurasi Mesin Turing:
 - Variabel dan kondisi awal mesin Turing diatur.
 - Mesin Turing diinisialisasi dengan status awal, kepala penulisan, dan daftar tape.
- Eksekusi Mesin Turing:
 - Mesin Turing dijalankan dalam loop while hingga mencapai kondisi akhir atau mencapai maksimum iterasi.
 - Pada setiap iterasi, mesin diperbarui dan statusnya dicetak.
- Output Hasil:
 - Hasil akhir dicetak, menyatakan apakah string adalah palindrom atau bukan, bersama dengan jumlah langkah yang diperlukan.

2) Program Bagian 2



```

1 import sys
2
3 from palindrome import check_palindrome
4
5 if __name__ == "__main__":
6
7     if len(sys.argv) != 2:
8         print("Usage:")
9         print("python test.py word")
10        print("where word is the word to be tested as a palindrome")
11        sys.exit(0)
12
13    word = sys.argv[1]
14    check_palindrome(word)

```

Gambar 4. Program bagian 2

Kodingan bagian ini berguna untuk memanggil fungsi atau logika turing ke dalam file palindrom dan di gunakan untuk menyelesaikan logika palindrom. Jadi, skrip ini digunakan untuk memeriksa apakah suatu kata (diambil dari argumen baris perintah)

adalah palindrom atau bukan, dengan memanfaatkan fungsi `check_palindrome` yang diharapkan ada di dalam modul `palindrome`.

3) Program Bagian 3

```
1 class TuringMachine:
2
3     def __init__(self, state, write_head, tape_list):
4         self.state = state
5         self.write_head = write_head
6         self.tape_list = tape_list
7
8     def getState(self):
9         return self.state
10
11    def getHead(self):
12        return self.write_head
13
14    def getTape(self):
15        return self.tape_list
16
17    # Table of rules!
18    def updateMachine(self, character_list):
19
20        # Initial State
21        if (self.state == 'q1'):
22            if (self.tape_list[self.write_head] != 0):
23                ## STATE ## (q)
24                char_read = self.tape_list[self.write_head]
25                char_index = character_list.index(char_read)
26                self.state = ''.join(['p', str(char_index)])
27                ## WRITE ## (zero)
28                self.tape_list[self.write_head] = 0
29                ## MOVE ## (right)
30                self.write_head += 1
31            else:
32                ## STATE ## (qy)
33                self.state = 'qy'
34                ## WRITE ## (zero, unchanged)
35                self.tape_list[self.write_head] = 0
36                ## MOVE ## (right) (doesn't matter)
37                self.write_head += 1
38
39        elif (self.state.startswith('p')):
40            if (self.tape_list[self.write_head] != 0):
41                ## STATE ## (unchanged)
42                self.state = self.state
43                ## WRITE ## (unchanged)
44                self.tape_list[self.write_head] = self.tape_list[self.write_head]
45                ## MOVE ## (right)
46                self.write_head += 1
47            else:
48                ## STATE ## (r)
49                self.state = ''.join(['r', self.state[1:]])
50                ## WRITE ## (zero, unchanged)
51                self.tape_list[self.write_head] = 0
52                ## MOVE ## (left)
53                self.write_head -= 1
54
55        elif (self.state.startswith('r')):
56            char_read = character_list[self.state[1]]
57            if (self.tape_list[self.write_head] != char_read and self.tape_list[self.write_head] != 0): # zero is needed for strings of odd length
58                ## STATE ## (q0)
59                self.state = 'q0'
60                ## WRITE ##
61                self.tape_list[self.write_head] = self.tape_list[self.write_head]
62                ## MOVE ## (left) (doesn't matter)
63                self.write_head -= 1
64            else:
65                ## STATE ##
66                self.state = 'q2'
67                ## WRITE ## (zero)
68                self.tape_list[self.write_head] = 0
69                ## MOVE ## (left)
70                self.write_head -= 1
71
72        elif (self.state == 'q2'):
73            if (self.tape_list[self.write_head] != 0):
74                ## STATE ## (unchanged)
75                self.state = 'q2'
76                ## WRITE ## (unchanged)
77                self.tape_list[self.write_head] = self.tape_list[self.write_head]
78                ## MOVE ## (left)
79                self.write_head -= 1
80            else:
81                ## STATE ## (q1)
82                self.state = 'q1'
83                ## WRITE ## (zero)
84                self.tape_list[self.write_head] = 0
85                ## MOVE ## (right)
86                self.write_head += 1
```

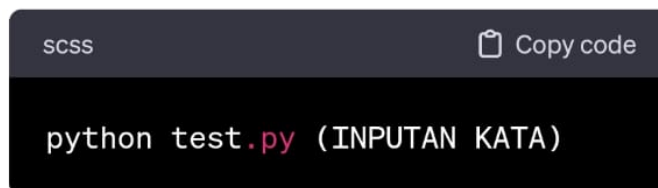
Gambar 5. Program bagian 3

kode ini mencoba untuk mensimulasikan perilaku mesin Turing dengan mengimplementasikan aturan-aturan tertentu untuk membaca dan memodifikasi daftar tape berdasarkan kondisi saat ini. Kode ini sebagai pendeklarasian kode untuk menjalankan 2 kode program sebelumnya

3.2 Langkah Menjalankan program yang telah dibuat

Berikut adalah langkah-langkah dalam menjalankan program yang sudah dijelaskan diatas :

1. Unduh dan Ekstrak File Zip :
 - Unduh file zip yang diberikan.
 - Ekstrak isi file zip tersebut ke dalam sebuah folder.
2. Buka Terminal atau CMD :
 - Buka terminal atau cmd di komputer Anda.
3. Navigasi ke Folder Hasil Ekstraksi :
 - Gunakan perintah cd untuk berpindah ke direktori tempat Anda mengekstrak file zip.
4. Jalankan Program:
 - Setelah berada di dalam direktori program, jalankan perintah:

A screenshot of a terminal window with a dark background. The top bar is dark gray and contains the text 'scss' on the left and a 'Copy code' button with a clipboard icon on the right. The main area of the terminal is black and displays the command 'python test.py (INPUTAN KATA)' in a light gray monospace font. The '.py' part of the command is highlighted in red.

Gambar 6. *Perintah di cmd*

Gantilah (INPUTAN KATA) dengan kata atau frase yang ingin Anda verifikasi sebagai palindrom.

BAB IV

PENUTUP

4.1 Kesimpulan

Berdasarkan pembahasan diatas diambil beberapa hal umum, yaitu :

- Mesin turing dapat di *design* untuk menyelesaikan permasalahan – permasalahan matematis
- Mesin turing adalah merupakan model matematis sederhana untuk komputer
- Jika suatu permasalahan dapat dimodelkan oleh mesin turing satu pita maka dapat pula dibuat untuk mesin turing dengan pita banyak (*multiple*), begitupula sebaliknya.
- Fungsi transisi yang terjadi pada mesin turing banyak pita lebih sedikit daripada yang dilakukan pada mesin turing satu pita.

Penelitian ini menghasilkan pemahaman yang lebih mendalam tentang implementasi mesin Turing dalam konteks verifikasi palindrom dengan mengimplementasikan bahasa pemrograman python. Mesin sederhana ini dapat menjadi dasar untuk pengembangan lebih lanjut dalam memahami dan merancang algoritma yang lebih kompleks. Dengan demikian, laporan ini memberikan kontribusi pada pemahaman tentang implementasi mesin Turing dengan menerapkan konsep tersebut pada kasus khusus, yaitu verifikasi palindrom. Kami berharap bahwa proyek sederhana ini dapat memberikan inspirasi dan bahan referensi bagi mereka yang tertarik pada pemrograman dan teori komputasi.

DAFTAR PUSTAKA

Abdullah, R. D. (t.thn.). Pengertian Palindrom, Penjelasan, Serta Contoh-contohnya. *Jejak Pustaka*(Jejak Pustaka). Dipetik April 2022

Saepudin, N. (2016). *Aplikasi Simulator Turing Pita Tunggal*. Bandung: Institut Teknologi Bandung.

H, F. S. (2015). *Mesin Turing Dan Polindrome*. Bandung.