KELOMPOK A7 - OTOMATA E

Muhammad Razan Athallah	5025211008
Alexander Weynard Samsico	5025211014
Apta Rasendriya Wijaya	5025211139

Laporan Praktikum - Palindrome

Soal:
Buatlah sebuah program komputer untuk pengenal string string palindrome. Dengan himpunan input string dibentuk dari (letter + digit)*

Solusi Program:

Program ini menggunakan bahasa pemrograman Python dan menerapkan konsep stack untuk memeriksa apakah suatu string adalah palindrome atau tidak. Stack diimplementasikan dengan menggunakan list bawaan Python. Program ini memiliki tiga fungsi utama:

1. **push(element: str)**: Fungsi ini menerima satu parameter **`element`** berupa string, yang kemudian ditambahkan ke dalam stack. Konsep yang digunakan adalah Last In First Out (LIFO), yang berarti elemen terakhir yang dimasukkan akan menjadi elemen pertama yang dikeluarkan.
2. **pop()**: Fungsi ini tidak menerima parameter dan mengembalikan elemen terakhir dari stack dengan menggunakan metode **`pop()`** pada list. Elemen yang diambil dari stack akan dihapus dari list.
3. **isPalindrome(string: str) -> bool**: Fungsi ini menerima satu parameter **`string`** berupa string dan mengembalikan nilai boolean. Fungsi ini digunakan untuk memeriksa apakah string yang diberikan adalah palindrome atau tidak. Palindrome adalah string yang jika dibalik akan tetap sama dengan string aslinya. Fungsi ini pertama-tama menghitung panjang string dan menemukan titik tengahnya. Kemudian, karakter dari setengah pertama string dimasukkan ke dalam stack menggunakan fungsi **`push()`**. Jika panjang string ganjil, karakter tengah diabaikan. Selanjutnya, karakter dari setengah kedua string dibandingkan dengan elemen yang diambil dari stack menggunakan fungsi **`pop()`**. Jika ada ketidakcocokan, fungsi akan mengembalikan **`False`**. Jika semua karakter cocok, fungsi akan mengembalikan **`True`**.

Contoh case input palindrom:

a. **Palindrom Ganjil** "a1b2c2b1a"

- Length = 9
- Mid = 9 // 2 = 4
- First loop (pushing to stack):
    - i = 0: push 'a' to stack. Stack: ['a']
    - i = 1: push '1' to stack. Stack: ['a', '1']
    - i = 2: push 'b' to stack. Stack: ['a', '1', 'b']
    - i = 3: push '2' to stack. Stack: ['a', '1', 'b', '2']
- Skip the middle character 'c'.
- Second loop (popping from stack and comparing):
    - i = 5: pop '2' from stack, compare with '2' -> match
    - i = 6: pop 'b' from stack, compare with 'b' -> match
    - i = 7: pop '1' from stack, compare with '1' -> match
    - i = 8: pop 'a' from stack, compare with 'a' -> match
- Result: "a1b2c2b1a" is a Palindrome

b. **Palindrom Genap** “123abcba321”

- Length = 12
- Mid = 12 // 2 = 6
- First loop (pushing to stack):
    - i = 0: push '1' to stack. Stack: ['1']
    - i = 1: push '2' to stack. Stack: ['1', '2']
    - i = 2: push '3' to stack. Stack: ['1', '2', '3']
    - i = 3: push 'a' to stack. Stack: ['1', '2', '3', 'a']
    - i = 4: push 'b' to stack. Stack: ['1', '2', '3', 'a', 'b']
    - i = 5: push 'c' to stack. Stack: ['1', '2', '3', 'a', 'b', 'c']
- No middle character to skip.
- Second loop (popping from stack and comparing):
    - i = 6: pop 'c' from stack, compare with 'c' -> match
    - i = 7: pop 'b' from stack, compare with 'b' -> match
    - i = 8: pop 'a' from stack, compare with 'a' -> match
    - i = 9: pop '3' from stack, compare with '3' -> match
    - i = 10: pop '2' from stack, compare with '2' -> match
    - i = 11: pop '1' from stack, compare with '1' -> match
- Result: "123abcba321" is a Palindrome

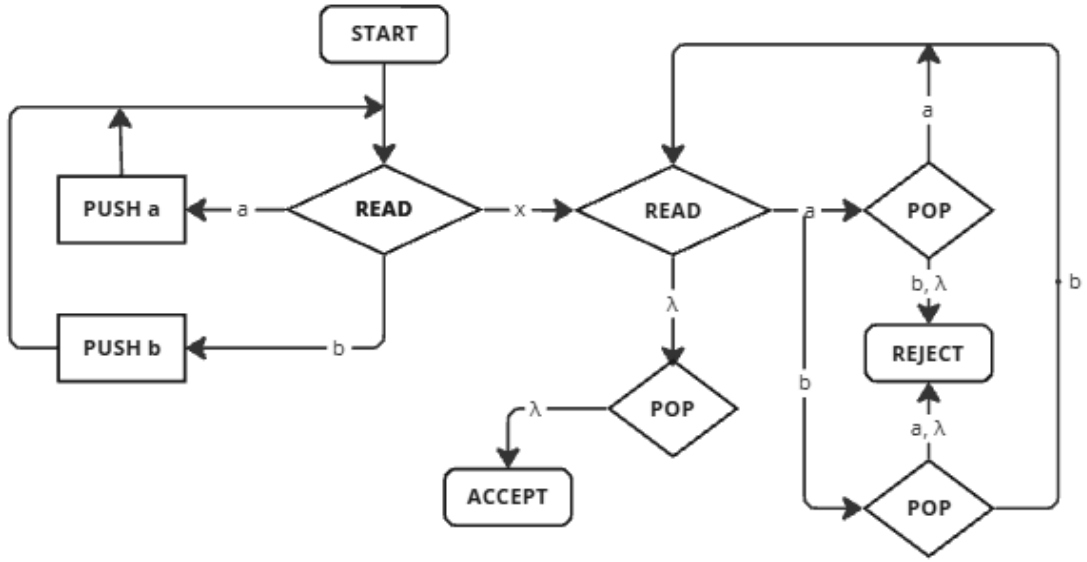
c. **Bukan Palindrom** “a1b2b1a3”

- Length = 7
- Mid = 7 // 2 = 3
- First loop (pushing to stack):
    - i = 0: push 'a' to stack. Stack: ['a']

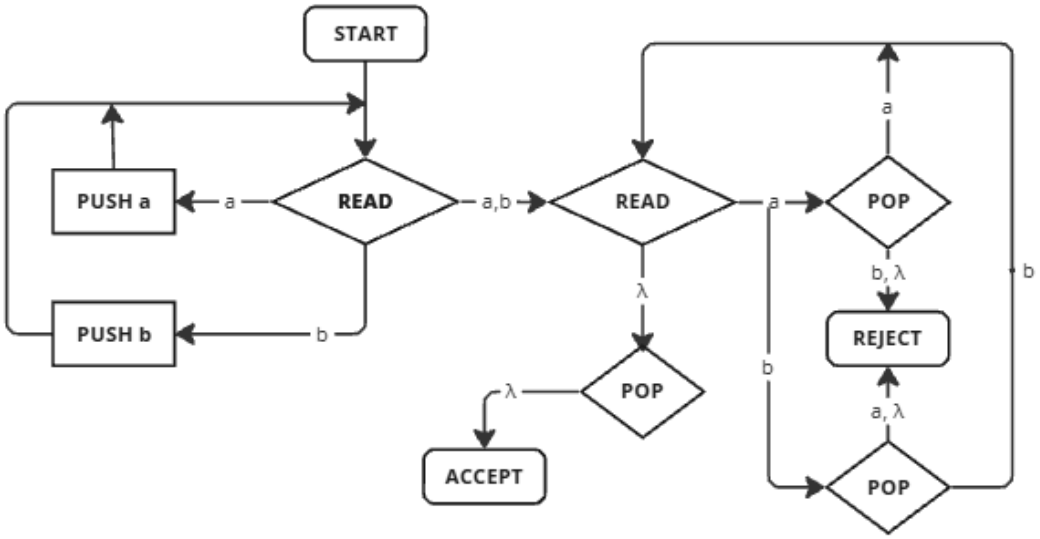
    - i = 1: push '1' to stack. Stack: ['a', '1']
    - i = 2: push 'b' to stack. Stack: ['a', '1', 'b']
- Skip the middle character '2'.
- Second loop (popping from stack and comparing):
    - i = 4: pop 'b' from stack, compare with 'b' -> match
    - i = 5: pop '1' from stack, compare with '1' -> match
    - i = 6: pop 'a' from stack, compare with '3' -> mismatch
- Result: "a1b2b1a3" is NOT a Palindrome

Konsep Otomata:
Solusi ini serupa dengan konsep Otomata yaitu pembuatan Pushdown Automata (PDA) pada palindrome. Khusus pada mesin otomata ini, palindrome merupakan bahasa dengan s X reverse(s) atau s reverse(s). Misalkan (letter + digit)* sebagai (a + b)* dan asumsikan mesin automata untuk pengecekan 2 karakter a dan b (bisa huruf atau digit) dan bisa jadi x di bagian tengah (dengan tujuan menyederhanakan mesin PDA), ada 3 PDA yang dibutuhkan

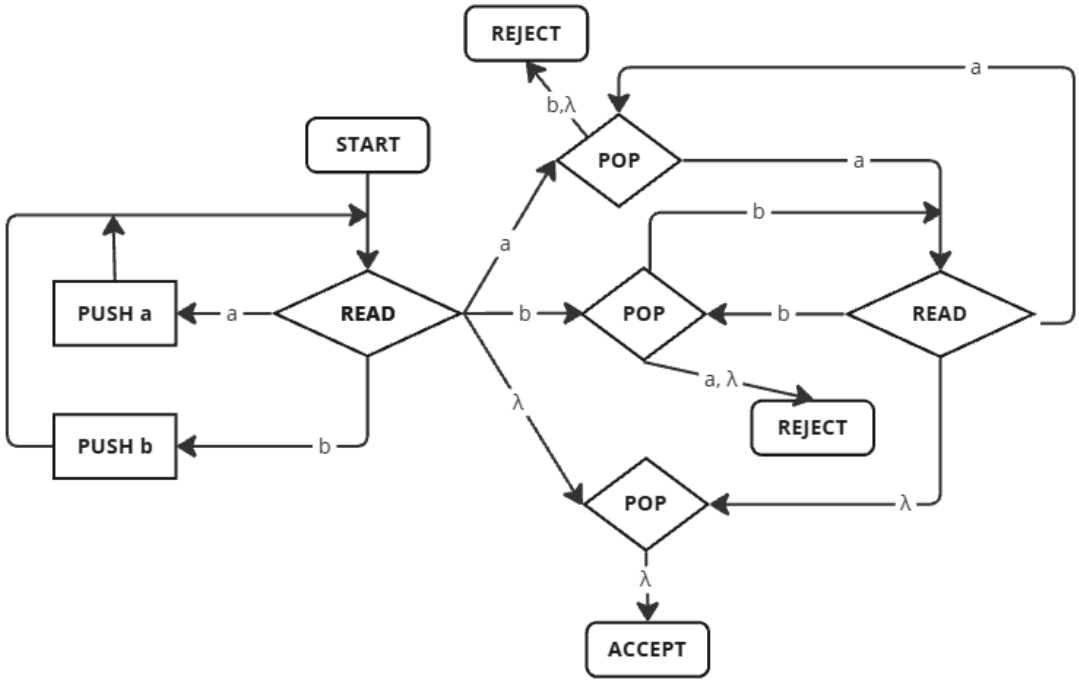
1. Palindrome ganjil dengan X di tengah



2. Palindrome ganjil



3. Palindrome genap



Mesin PDA pada (1) bersifat deterministic karena mesin dapat bergerak ke read kanan karena indikator x, sedangkan pada (2) and (3) bersifat nondeterministic karena ada 2 pilihan jalur dengan karakter yang sama pada read pertama (kiri), sehingga mesin ini harus dapat memutuskan kapan karakter bergerak ke read kanan, yaitu saat karakter berada di tengah kata, agar mendapatkan hasil **accept**. Terdapat **reject** jika READ dan POP tidak sesuai di kanan.

Contoh kasus sederhana:

A. aabxbaa (menggunakan PDA ke 1)

0. Initial

Input	a	a	b	x	b	a	a	λ
Stack	λ							

1. READ a

Input	a	b	x	b	a	a	λ	
Stack	λ							

2. PUSH a

Input	a	b	x	b	a	a	λ	
Stack	a	λ						

3. READ a

Input	b	x	b	a	a	λ		
Stack	a	λ						

4. PUSH a

Input	b	x	b	a	a	λ		
Stack	a	a	λ					

5. READ b & PUSH b

Input	x	b	a	a	λ			
Stack	b	a	a	λ				

6. READ x

Input	b	a	a	λ				
Stack	b	a	a	λ				

7. READ b and POP b

Input	a	a	λ					
Stack	a	a	λ					

8. READ a and POP a

Input	a	λ						
Stack	a	λ						

9. READ a and POP a

Input	λ							
Stack	λ							

10. READ λ and POP λ

Input								
Stack								

Result: **ACCEPT**

B. aba (menggunakan PDA ke 2)

0. Initial

Input	a	b	a	λ				
Stack	λ							

1. READ a and PUSH a

Input	b	a	λ					
Stack	a	λ						

2. READ b

Input	a	λ						
Stack	a	λ						

3. READ a and POP a

Input	λ							
Stack	λ							

4. READ λ and POP λ

Input								
Stack								

Result: **ACCEPT**

C. abaaba (menggunakan PDA ke 3)

0. Initial

Input	a	b	a	a	b	a	λ	
Stack	λ							

1. READ a and PUSH a

Input	b	a	a	b	a	λ		
Stack	a	λ						

2. READ b and PUSH b

Input	a	a	b	a	λ			
Stack	b	a	λ					

3. READ a and PUSH a

Input	a	b	a	λ				
Stack	a	b	a	λ				

4. READ a and POP a

Input	b	a	λ					
Stack	b	a	λ					

5. READ b and POP b

Input	a	λ						
Stack	a	λ						

6. READ a and POP a

Input	λ							
Stack	λ							

7. READ λ and POP λ

Input								
Stack								

Result: **ACCEPT**

D. axb

0. Initial

Input	a	x	b	λ				
Stack	λ							

1. READ a and PUSH a

Input	x	b	λ					
Stack	a	λ						

2. READ x

Input	b	λ						
Stack	a	λ						

3. READ b and POP a

Input	λ							
Stack	λ							

Result: **REJECT**