Tugas TBO 6

Davidson Rafael Krisman Nugroho - 412024030

October 8, 2025

Soal

Di Kutub Selatan terdapat enam binatang yang terdiri dari induk pinguin, anak pinguin, induk singa laut, anak singa laut, induk beruang kutub dan anak beruang kutub. Semua binatang ini hendak menyeberang ke gunung es pada sisi lainnya. Diantara dua gunung es ini terdapat balok es yang hanya bisa ditempati untuk dua binatang saja. Namun terdapat suatu masalah yaitu anak binatang tidak boleh ditinggal bersama induk binatang yang berlainan jenis dan singa laut harus menjadi yang pertama sampai di sisi lainnya karena anak singa laut sedang terluka sehingga harus cepat sampai di kelompoknya. Bantulah mereka untuk menyeberang ke gunung es pada sisi lainnya.

Jawaban

Untuk menyelesaikan masalah ini, kita dapat menggunakan pendekatan langkah demi langkah untuk memeastikan bahwa semua binatang dapat menyeberang dengan aman sesuai dengan aturan yang diberikan. Dan saya juga memberikan kode python untuk mensimulasikan proses penyeberangan ini, serta JFLAP untuk memvisualisasikan solusi. Jawaban lengkapnya ada di halaman berikutnya.

Penjelasan

Untuk dapat memahami solusi dari masalah ini, kita perlu memahami aturan-aturan yang ada:

- Anak binatang tidak boleh ditinggal bersama induk binatang yang berlainan jenis.
- Singa laut harus menjadi yang pertama sampai di sisi lainnya.
- Balok es hanya bisa menampung dua binatang saja.

Kemudian kita permudah dengan memberikan notasi pada setiap binatang:

Notasi	Deskripsi
B	Induk Beruang Kutub
b	Anak Beruang Kutub
P	Induk Pinguin
p	Anak Pinguin
S	Induk Singa Laut
s	Anak Singa Laut

Table 1: Notasi Setiap Binatang

State	KIRI	KANAN	Keterangan
q0	BbPpSs	0	State Awal
q1	BbPp	Ss	
q2	BbPpS	s	
q3	BPS	bps	
q4	BPSs	bp	
q5	Ss	BbPp	
q6	BSbs	Pp	
q7	bs	BPSp	
q8	bps	BPS	
q9	p	BPSbs	
q10	Pp	BSbs	
q11	0	BbPpSs	State Final

Table 2: Transisi antar state

Code Python

```python
"""
Problem: Menyeberangkan 6 binatang di Kutub Selatan
- Induk pinguin (P), anak pinguin (p)
- Induk singa laut (S), anak singa laut (s)
- Induk beruang kutub (B), anak beruang kutub (b)

Constraint:
1. Balok es hanya muat 2 binatang
2. Anak tidak boleh ditinggal dengan induk jenis lain
3. Singa laut (S) harus sampai pertama kali di sisi lain
"""

from collections import deque
from copy import deepcopy

class State:
    def __init__(self, left, right, boat_pos, path=[], first_trip_done=False):
        self.left = set(left)  # Binatang di sisi kiri
        self.right = set(right)  # Binatang di sisi kanan
        self.boat_pos = boat_pos  # 'left' atau 'right'
        self.path = path.copy()  # Riwayat perpindahan
        self.first_trip_done = first_trip_done  # Apakah perjalanan pertama sudah dilakukan

    def __eq__(self, other):
        return (self.left == other.left and
                self.right == other.right and
                self.boat_pos == other.boat_pos and
                self.first_trip_done == other.first_trip_done)

    def __hash__(self):
        return hash((frozenset(self.left), frozenset(self.right), self.boat_pos, self.first_trip_done))

    def is_valid(self):
        """Cek apakah state valid (anak tidak sendirian dengan induk lain)"""
        for side in [self.left, self.right]:
            if len(side) == 0:
                continue

            # Cek anak pinguin
            if 'p' in side and 'P' not in side:
                # Ada anak pinguin tanpa induknya
                if 'S' in side or 'B' in side:
                    return False

            # Cek anak singa laut
            if 's' in side and 'S' not in side:
                # Ada anak singa laut tanpa induknya
                if 'P' in side or 'B' in side:
                    return False

            # Cek anak beruang kutub
            if 'b' in side and 'B' not in side:
                # Ada anak beruang kutub tanpa induknya
                if 'P' in side or 'S' in side:
                    return False

        return True

    def is_goal(self):
        """Cek apakah semua binatang sudah di sisi kanan"""
        return len(self.left) == 0

    def get_next_states(self):
        """Generate semua state berikutnya yang mungkin"""
        next_states = []

        if self.boat_pos == 'left':
            current_side = self.left
            other_side = self.right
            new_boat_pos = 'right'
        else:
            current_side = self.right
            other_side = self.left
            new_boat_pos = 'left'

        animals = list(current_side)

        # Coba pindahkan 1 binatang
        for animal in animals:
            new_current = current_side - {animal}
            new_other = other_side | {animal}

            # Tentukan apakah perjalanan pertama sudah dilakukan
            new_first_trip = self.first_trip_done or (self.boat_pos == 'left' and new_boat_pos ==
'right')

            if self.boat_pos == 'left':
                new_state = State(new_current, new_other, new_boat_pos,
                                  self.path + [f"Pindahkan {animal}"], new_first_trip)
            else:
                new_state = State(new_other, new_current, new_boat_pos,
                                  self.path + [f"Kembali {animal}"], new_first_trip)

            # Constraint: Perjalanan pertama HARUS membawa S
            if not self.first_trip_done and self.boat_pos == 'left' and new_boat_pos == 'right':
                if animal != 'S':
                    continue

            if new_state.is_valid():
                next_states.append(new_state)

        # Coba pindahkan 2 binatang
        for i in range(len(animals)):
            for j in range(i + 1, len(animals)):
                animal1, animal2 = animals[i], animals[j]
                new_current = current_side - {animal1, animal2}
```

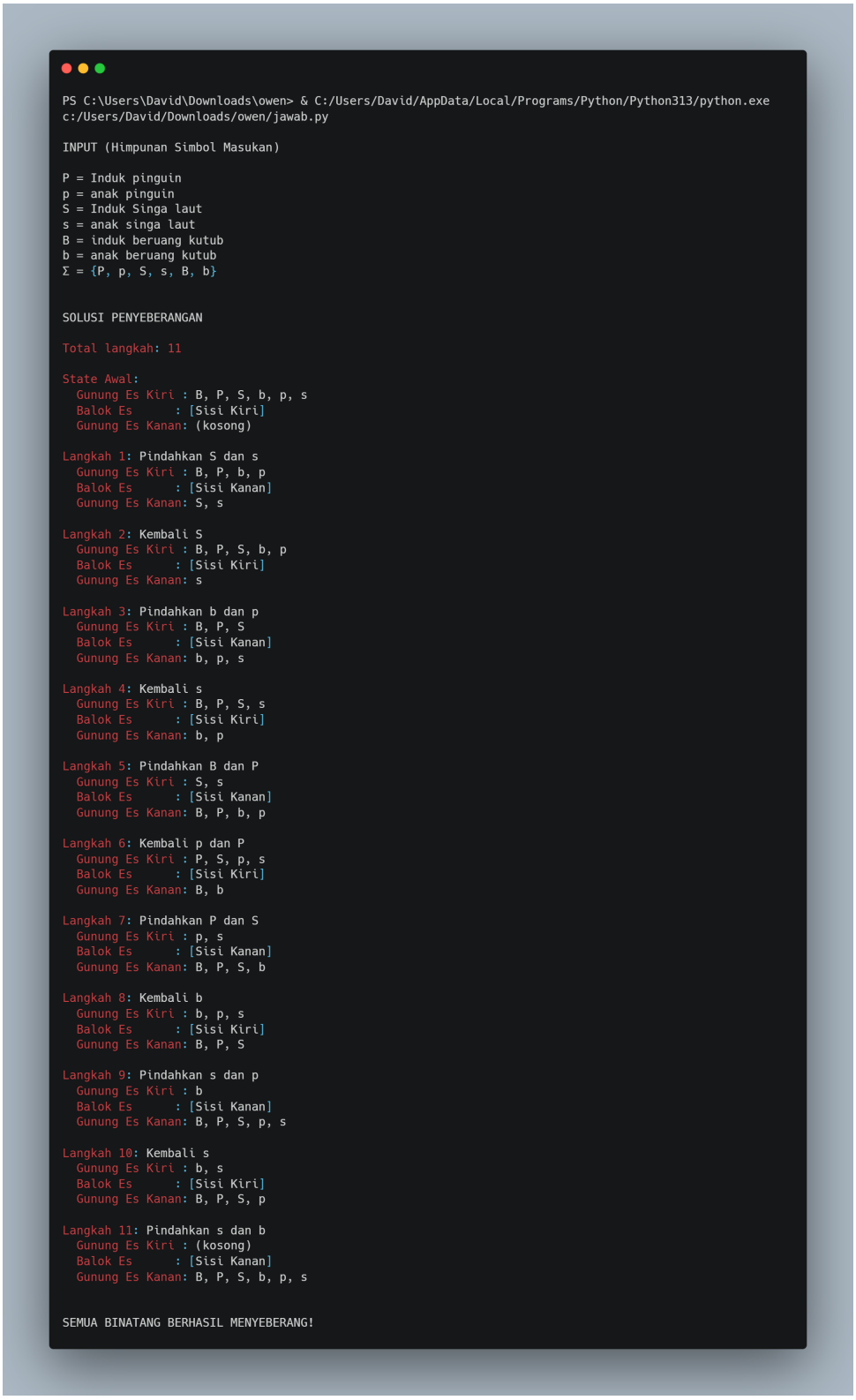

Figure 2: Output Simulasi Penyeberangan Binatang dengan Python

JFLAP

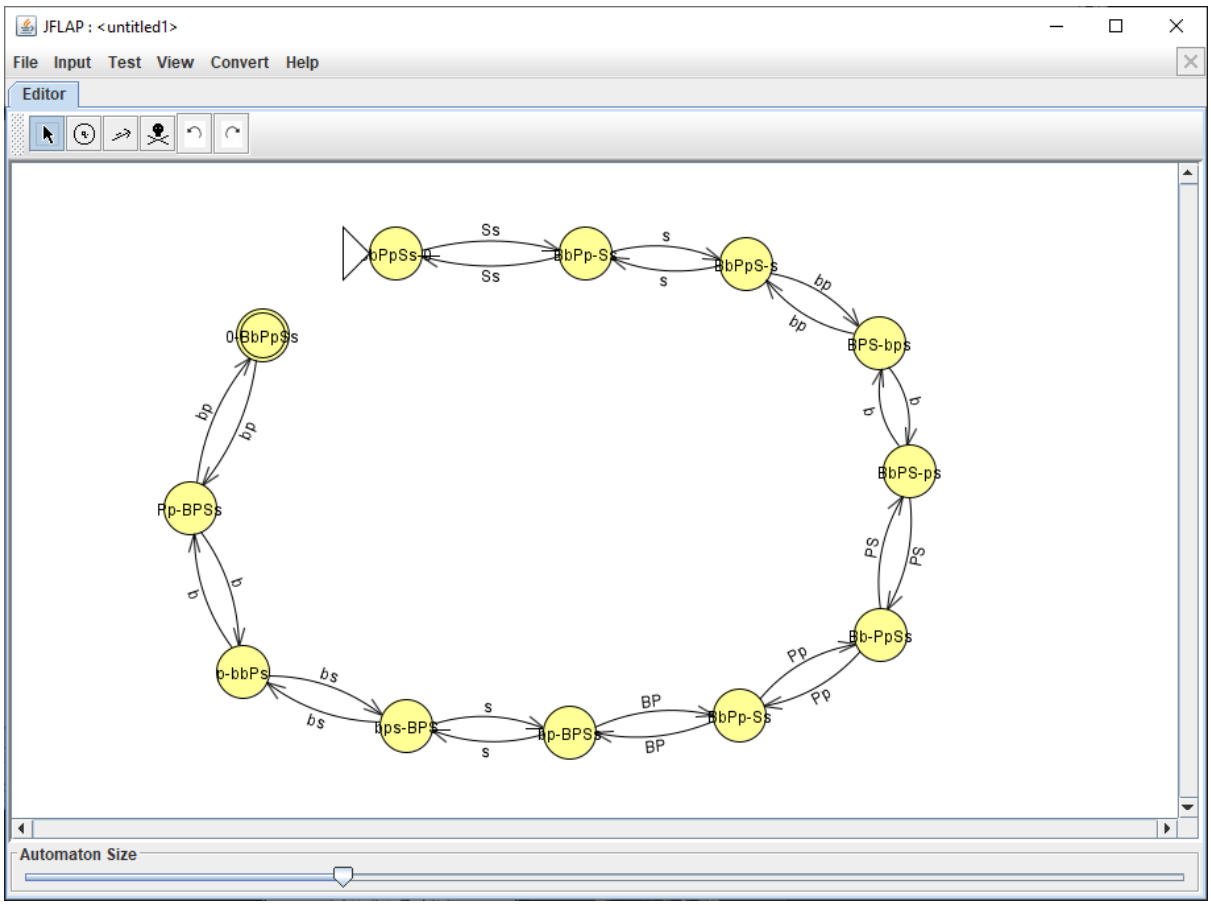

Figure 3: Visualisasi Penyeberangan Binatang dengan JFLAP (Bagian 1)

Terminologi

- **State**: Kondisi atau situasi tertentu dalam proses penyeberangan binatang.
- **Transisi**: Perpindahan dari satu state ke state lainnya berdasarkan aturan yang telah ditentukan.
- **Notasi**: Simbol atau singkatan yang digunakan untuk mewakili binatang dalam proses penyeberangan.