

# LAMPIRAN TUGAS

BFS, DLS, UCS



Dikerjakan Oleh:

Mochamad Haykal Alvegio Hadian

NRP :

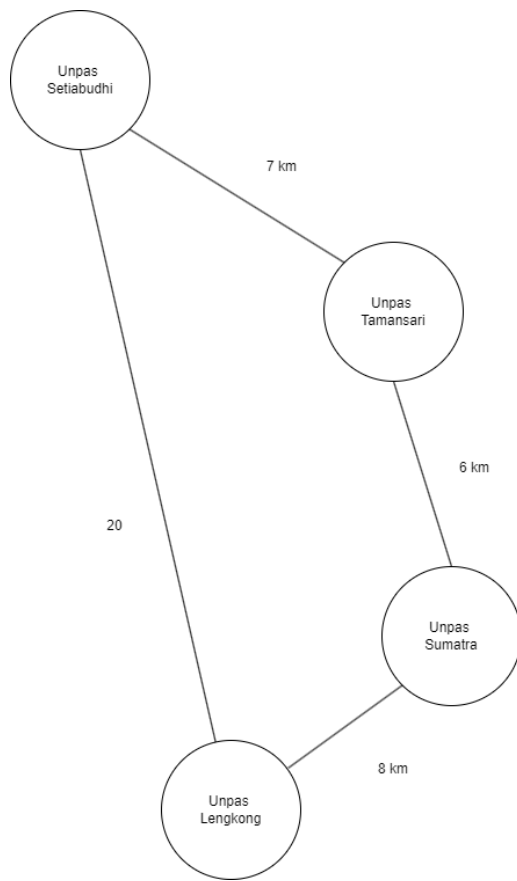
213040028

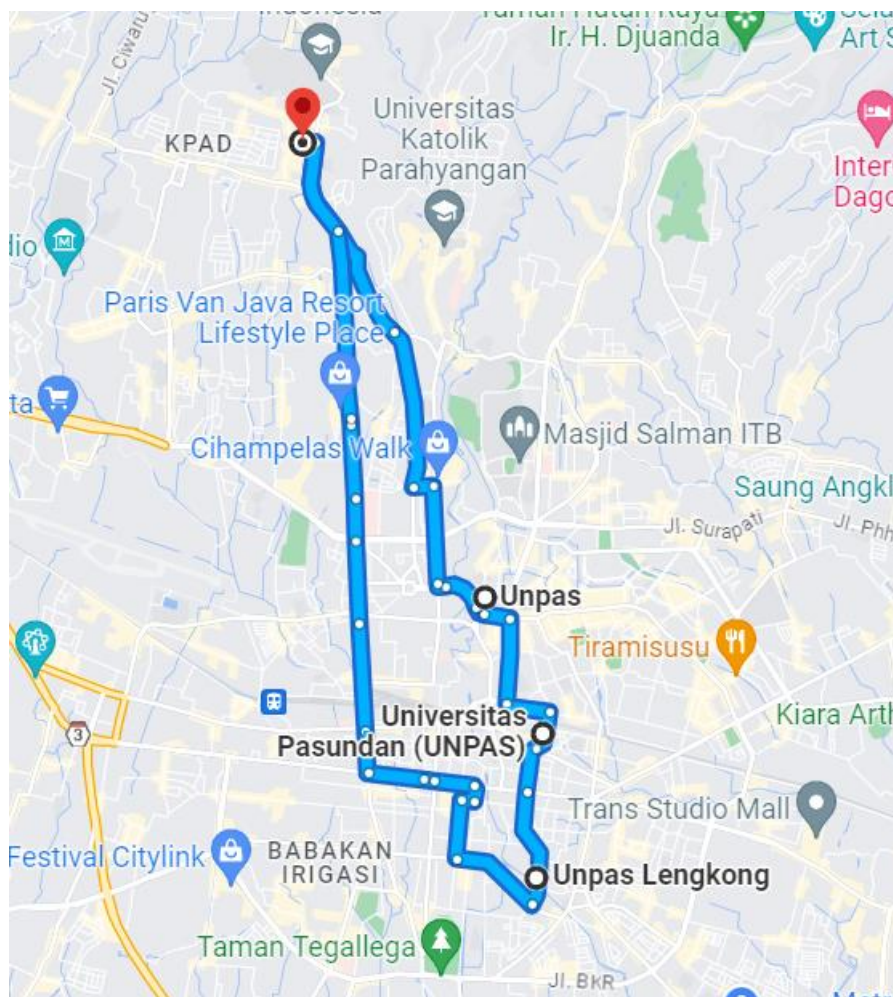
## **PRODI TEKNIK INFORMATIKA UNIVERSITAS PASUNDAN BANDUNG 2023**

### **A. Penjelasan Studi Kasus**

Studi Kasus: Pencarian rute tercepat untuk Program Mentoring  
Di Universitas Pasundan

dibuatkan studi untuk menceritakan rute tercepat dari kota asal ke kota tujuan dengan meminilisir waktu perjalanan.





## B. Kode Program

Node.java

```
package id.ac.unpas.ib;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class Node {  
    private String nilai;  
    private List<Node> tetangga;  
  
    public Node(String nilai) {  
        tetangga = new ArrayList<>();  
        this.nilai = nilai;  
    }  
  
    public void setNilai(String nilai) {  
        this.nilai = nilai;  
    }  
  
    public String getNilai() {  
        return nilai;  
    }  
}
```

```

    public void setTetangga(List<Node> tetangga) {
        this.tetangga = tetangga;
    }

    public List<Node> getTetangga() {
        return tetangga;
    }

    public void addTetangga(Node node) {
        tetangga.add(node);
    }

    @Override
    public boolean equals(Object obj) {
        if (obj instanceof Node) {
            return ((Node) obj).nilai.equals(this.nilai);
        }
        return false;
    }
}

```

Maintest.java

```

package id.ac.unpas.ib;

import id.ac.unpas.ib.bfs.BreadthFirstSearch;
import id.ac.unpas.ib.dls.DepthLimitedSearch;
import id.ac.unpas.ib.ucs.NodeUCS;
import id.ac.unpas.ib.ucs.UniformCostSearch;

public class MainTest {
    public static void main(String[] args) {
        Node unpasSetiabudhi = new Node("unpasSetiabudhi");
        Node unpasTamansari = new Node("unpasTamansari");
        Node unpasSumatra = new Node("unpasSumatra");
        Node unpasLengkong = new Node("unpasLengkong");
        // Empat objek Node yaitu unpasSetiabudhi, unpasTamansari, unpasSumatra,
        dan unpasLengkong dibuat di sini.
        // Masing-masing merepresentasikan simpul dalam graf. Setiap simpul
        memiliki nama yang diberikan sebagai argumen dalam konstruktor.

        unpasSetiabudhi.addTetangga(unpasTamansari);
        unpasSetiabudhi.addTetangga(unpasLengkong);

        unpasTamansari.addTetangga(unpasSetiabudhi);
        unpasTamansari.addTetangga(unpasSumatra);

        unpasSumatra.addTetangga(unpasTamansari);
        unpasSumatra.addTetangga(unpasLengkong);
    }
}

```

```

        unpasLengkong.addTetangga(unpasSumatra);
        unpasLengkong.addTetangga(unpasSetiabudhi);

        // Simpul-simpul dihubungkan satu sama lain dengan menggunakan metode
        addTetangga.
        // Ini membangun hubungan antara simpul-simpul dalam graf.

        System.out.println("BFS");
        BreadthFirstSearch bfs = new BreadthFirstSearch();
        bfs.search(unpasSetiabudhi, unpasLengkong);
        System.out.println();

        // Pada bagian ini, algoritma Breadth First Search (BFS) diterapkan.
        Objek bfs dibuat dari kelas BreadthFirstSearch.
        // Metode search kemudian dipanggil dengan menyediakan simpul awal
        (unpasSetiabudhi) dan simpul tujuan (unpasLengkong)

        System.out.println("DLS");
        DepthLimitedSearch dls = new DepthLimitedSearch();
        dls.setLimit(10);
        dls.search(unpasSetiabudhi, unpasLengkong);
        System.out.println();

        // Pada bagian ini, algoritma Depth Limited Search (DLS) diterapkan.
        Objek dls dibuat dari kelas DepthLimitedSearch. Batasan kedalaman (10 dalam
        contoh ini) ditetapkan dengan menggunakan metode setLimit.
        // Metode search kemudian dipanggil dengan menyediakan simpul awal
        (unpasSetiabudhi) dan simpul tujuan (unpasLengkong).

        NodeUCS unpasSetiabudhiUCS = new NodeUCS(unpasSetiabudhi, 20);
        NodeUCS unpasTamansariUCS = new NodeUCS(unpasTamansari, 7);
        NodeUCS unpasSumatraUCS = new NodeUCS(unpasSumatra, 6);
        NodeUCS unpasLengkongUCS = new NodeUCS(unpasLengkong, 0);

        // Empat objek NodeUCS yaitu unpasSetiabudhiUCS, unpasTamansariUCS,
        unpasSumatraUCS, dan unpasLengkongUCS dibuat di sini.
        // Setiap objek NodeUCS memiliki dua argumen, yaitu simpul yang
        direpresentasikan dan nilai heuristik

        unpasSetiabudhiUCS.addTetangga(unpasTamansariUCS);
        unpasSetiabudhiUCS.addTetangga(unpasLengkongUCS);

        unpasTamansariUCS.addTetangga(unpasSetiabudhiUCS);
        unpasTamansariUCS.addTetangga(unpasSumatraUCS);

        unpasSumatraUCS.addTetangga(unpasTamansariUCS);
        unpasSumatraUCS.addTetangga(unpasLengkongUCS);

        unpasLengkongUCS.addTetangga(unpasSumatraUCS);
        unpasLengkongUCS.addTetangga(unpasSetiabudhiUCS);

```

// Seperti pada poin 4, simpul-simpul dihubungkan satu sama lain dalam graf, kali ini menggunakan objek NodeUCS.

```
System.out.println("UCS");
UniformCostSearch ucs = new UniformCostSearch();
ucs.search(unpasSetiabudhiUCS, unpasLengkongUCS);
System.out.println();
```

// Pada bagian ini, algoritma Uniform Cost Search (UCS) diterapkan. Objek ucs dibuat dari kelas UniformCostSearch.

// Metode search kemudian dipanggil dengan menyediakan simpul awal (unpasSetiabudhiUCS) dan simpul tujuan (unpasLengkongUCS).

```
/*
 *
 * author :
 * mochamad haykal alvegio hadian
 * 213040028
 */
}
```

### C. Hasil Program

```
BFS
Mencari solusi dari unpaSetiabudhi ke unpaLengkong
Evaluasi: unpaSetiabudhi
Suksesor unpaSetiabudhi
unpaTamansari unpaLengkong

Evaluasi: unpaTamansari
Suksesor unpaTamansari
unpaSetiabudhi unpaSumatra

Evaluasi: unpaLengkong
Solusi ditemukan:
unpaSetiabudhi -> unpaLengkong

DLS
Mencari solusi dari unpaSetiabudhi ke unpaLengkong
Level 0 evaluasi: unpaSetiabudhi
Level 1 evaluasi: unpaTamansari
Level 2 evaluasi: unpaSetiabudhi
Level 3 evaluasi: unpaTamansari
Level 4 evaluasi: unpaSetiabudhi
Level 5 evaluasi: unpaTamansari
Level 6 evaluasi: unpaSetiabudhi
Level 7 evaluasi: unpaTamansari
Level 8 evaluasi: unpaSetiabudhi
Level 9 evaluasi: unpaTamansari

UCS
Mencari solusi dari unpaSetiabudhi ke unpaLengkong
Evaluasi: unpaSetiabudhi dengan cost 20

unpaTamansari (7), unpaLengkong (0),
Node terpilih: unpaLengkong (0)
Evaluasi: unpaLengkong dengan cost 0
Solusi ditemukan

unpaSumatra (6), unpaSetiabudhi (20),
Node terpilih: unpaSumatra (6)

Solusi ditemukan:
unpaSetiabudhi (20)
unpaLengkong (0)
```

Link Repository : [https://github.com/HLHERE/Tugas\\_IB\\_Problem\\_Solving\\_By\\_Searching](https://github.com/HLHERE/Tugas_IB_Problem_Solving_By_Searching)