

# Penggunaan Graf dalam Menentukan Pengeluaran Minimum Kebutuhan Mahasiswa

Muhammad Syarafi Akmal - 13522076<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

<sup>1</sup>13522076@std.stei.itb.ac.id

**Abstract**—The abstract is to be in fully-justified italicized text, at the top of the left-hand column as it is here, below the author information. The abstract is to be in 9-point, single-spaced type, and may be up to 8 cm long. Define all symbols used in the abstract. Do not cite references in the abstract. Do not delete the blank line immediately above the abstract; it sets the footnote at the bottom of this column. Leave two blank lines after the index terms, then begin the main text. All manuscripts must be in English.

**Keywords**—About four key words or phrases in alphabetical order, separated by commas.

## I. PENDAHULUAN

Sebagai mahasiswa, kita selalu menghadapi kesibukan kegiatan kuliah, akademis maupun non-akademis. Salah satu tantangan dalam menjalaninya adalah mengelola kesibukan tersebut dengan baik. Sebagai mahasiswa kita harus membiasakan diri untuk selalu produktif, salah satu cara untuk meningkatkan produktivitas adalah dengan memiliki disiplin dalam mengelola aktivitas keseharian atau kemampuan manajemen. Disiplin manajemen sangat krusial bagi mahasiswa agar kegiatan keseharian mereka tertata dan tidak *chaotic*, kegiatan yang tertata akan sangat membantu mahasiswa dalam menentukan strategi aktivitas harian mereka agar dapat memaksimalkan produktivitas.

Pada makalah ini kita akan membahas dan mencari solusi permasalahan pada salah satu disiplin manajemen, yaitu manajemen keuangan. Terdapat sebuah istilah yang sangat populer di kalangan mahasiswa yaitu 'akhir bulan'. Istilah ini menggambarkan kondisi keuangan mahasiswa yang menerima uang bulanan pada akhir bulan, dimana jumlah uang mereka yang menyusut dikarenakan keputusan-keputusan finansial yang buruk. Kepopuleran istilah ini menunjukkan bahwa masih banyak mahasiswa yang mengalami permasalahan dalam mengelola keuangan mereka.

Dalam menentukan keputusan finansial, mahasiswa perlu menyusun sebuah perencanaan untuk mencari keputusan finansial yang tepat secara menyeluruh atau untuk jangka waktu per bulan.

## II. DASAR TEORI

### A. Graf

Graf adalah salah satu cabang ilmu matematika yang diajarkan pada mata kuliah Matematika Diskrit yang secara umum digunakan untuk menggambarkan/mengilustrasikan hubungan antar objek-objek diskrit. Sebuah graf terdiri atas kumpulan simpul-simpul (vertex) yang terhubung oleh sisi-sisi (edge). Secara notasi formal, sebuah graf  $G$  terdefinisi dengan notasi tuple  $(V, E)$ , dengan  $V$  sebagai himpunan tak kosong yang berisi simpul-simpul (vertex) dan  $E$  sebagai himpunan (bisa kosong) yang berisi sisi-sisi (edge).

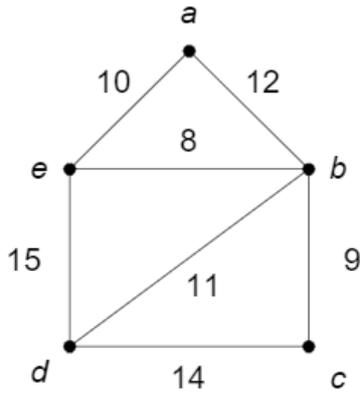
Berikut contoh notasi Graf:

$$\begin{aligned}G &= (V, E) \\V &= \{a, b, c, d, e\} \\E &= \{(a, b), (b, c), (c, d), (d, e)\}\end{aligned}$$

Teori graf memiliki banyak aplikasi di kehidupan terutama pada bidang engineering, ilmu komputer, social media, dan transportasi. Pada ilmu komputer, teori graf digunakan untuk *data mining*, *networking*, analisis sekuensi algoritma, *image capturing*, *optimal path finding* dalam sistem software. Pada transportasi, teori graf digunakan untuk *airline scheduling* untuk mencari *shortest path* yang melewati semua simpul (bandara). Pada social media, graf digunakan untuk algoritma relevansi target dari konten dengan menggunakan algoritma Karger.

*Weighted Graf* adalah salah satu bentuk graf, dimana setiap sisi dari graf memiliki sebuah nilai numerik. Nilai numerik ini melambangkan harga (bobot) dari hubungan antar simpul-simpul.

Berikut contoh *weighted graph*:

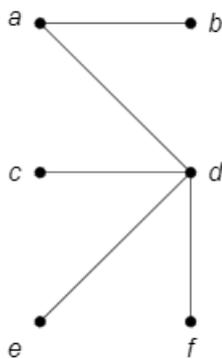


Gambar 2.1 Graf berbobot.

### B. Pohon

Pohon adalah salah satu cabang ilmu dalam matematika yang diajarkan di mata kuliah Matematika Diskrit yang merupakan turunan dari graf. Secara definisi, pohon adalah graf tak berarah yang tak memiliki sirkuit. Sirkuit adalah jalur yang dituju sedemikian rupa sehingga dapat kembali ke posisi awal.

Berikut adalah contoh Pohon:



Gambar 2. 2 Pohon.

Teori pohon memiliki banyak aplikasi dalam kehidupan, terutama di bidang ilmu komputer. Pada bidang ilmu komputer, teori pohon digunakan untuk system file, database, rute network, machine learning, syntax parsing dll.

*Minimum Spanning Tree* adalah pohon yang dibentuk dari *weighted graph* dimana pohon ini merepresentasikan jalur dari simpul awal sampai dengan simpul tujuan yang memiliki bobot akumulatif paling kecil.

Oleh karena itu, konsep teori graf dan tree sangatlah cocok untuk penyelesaian masalah ini dikarenakan manajemen keuangan akan memiliki cabang keputusan yang berjumlah banyak. Hal ini dapat direpresentasikan dengan tree dan graf yang dapat mengilustrasikan banyak objek diskrit dan hubungan diantaranya. *Weighted Graph* juga merupakan representasi yang cocok untuk permasalahan ini karena melambangkan harga dari keputusan-keputusan finansial yang akan diambil. Proses

pencarian pengeluaran minimum juga dapat dilakukan dengan mencari *minimum spanning tree*.

### A. Algoritma

Terdapat beberapa algoritma untuk mencari *minimum span tree* dari *weighted graph* yaitu:

1. Algoritma Prim, algoritma prim adalah algoritma untuk membangun *minimum spanning tree* dengan proses *node by node* dan dilakukan secara sekuensial. Secara singkat, algoritma ini mencari rute per simpul dengan harga terkecil.

Berikut adalah contoh algoritma Prim dalam pseudocode:

```

procedure Prim(input G : graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf terhubung-berbobot G.
Masukan: graf-berbobot terhubung G = (V, E), dengan |V|= n
Keluaran: pohon rentang minimum T = (V, E')
}
Deklarasi
i, p, q, u, v : integer

Algoritma
Cari sisi (p,q) dari E yang berbobot terkecil
T ← {(p,q)}
for i←1 to n-2 do
  Pilih sisi (u,v) dari E yang bobotnya terkecil namun
  bersisian dengan simpul di T dan (u,v) tidak membentuk
  sirkuit di T
  T ← T ∪ {(u,v)}
endfor
  
```

Gambar 2.3 Algoritma Prim

2. Algoritma Kruskal, berbeda dengan prim, algoritma Kruskal membangun *minimum spanning tree* dengan proses *edge by edge*. Secara singkat, algoritma ini membangun *minimum span tree* dengan terlebih dahulu melakukan sorting terhadap harga dari semua sisi pada graf yang dinotasikan dengan *tuple* (node1, node2).

Berikut adalah algoritma Kruskal dalam pseudocode:

```

procedure Kruskal(input G : graf, output T : pohon)
{ Membentuk pohon merentang minimum T dari graf terhubung -berbobot G.
Masukan: graf-berbobot terhubung G = (V, E), dengan |V|= n
Keluaran: pohon rentang minimum T = (V, E')
}
Deklarasi
i, p, q, u, v : integer

Algoritma
( Asumsi: sisi-sisi dari graf sudah diurut menaik
berdasarkan bobotnya - dari bobot kecil ke bobot
besar)
T ← {}
while jumlah sisi T < n-1 do
  Pilih sisi (u,v) dari E yang bobotnya terkecil
  if (u,v) tidak membentuk siklus di T then
    T ← T ∪ {(u,v)}
  endif
endfor
  
```

Gambar 2.4 Algoritma Kruskal

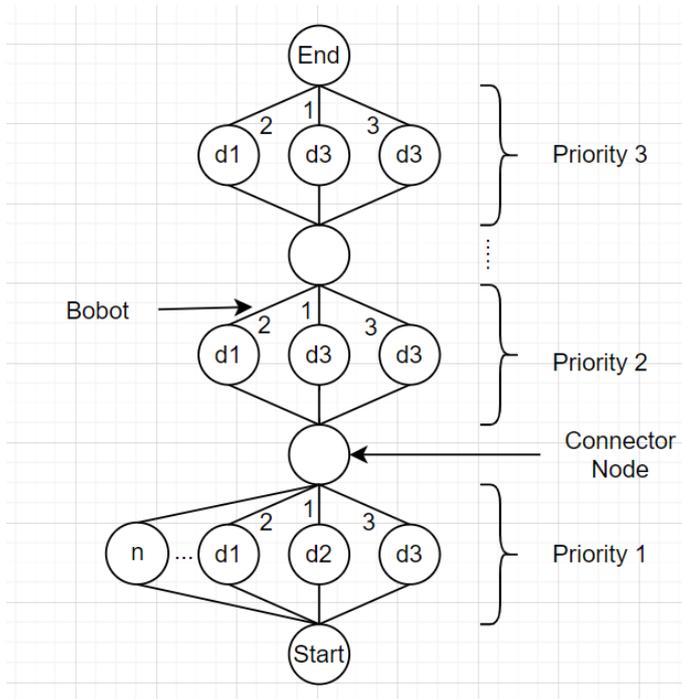
## III. METODE

### A. Struktur Graf Manajemen Uang

Dengan konteks penyelesaian masalah manajemen keuangan bulanan, struktur graf perlu disusun secara sistematis. Hal ini sebab pada manajemen keuangan terdapat prioritas pengeluaran

yang dilakukan secara sekuensial berdasarkan prioritas. Oleh karena itu, graf perlu disusun secara sistematis berdasarkan prioritas.

Berikut adalah ilustrasi dari penyusunan graf manajemen keuangan:



Gambar 3.1. Graf Manajemen Uang

Secara umum, graf akan disusun secara sistematis berdasarkan prioritas seperti pada gambar 3.1. Struktur dari sebuah prioritas terdiri dari simpul-simpul *decision* yang dapat ditambah hingga  $n$  buah beserta dengan bobot-bobotnya. Struktur graf juga memungkinkan untuk menambah sub-struktur prioritas hingga  $n$  buah prioritas. *Connector Node* bukan merupakan bagian dari struktur, tujuan ada *connector node* hanya agar graf terlihat lebih rapi dan terstruktur.

Struktur graf di atas dibuat dinamis agar memungkinkan untuk melakukan penyesuaian perencanaan manajemen keuangan yang dapat berubah seiring waktu (menambah keputusan).

### B. Algoritma Pencarian MST (minimum spanning tree)

Seperti yang sudah dibahas pada BAB II, terdapat 2 algoritma untuk mencari *minimum spanning tree*, yaitu algoritma Prim dan algoritma Kruskal. Pada konteks desain struktur graf diatas, graf akan memiliki banyak simpul dikarenakan semua kemungkinan keputusan yang dapat ditambahkan. Oleh karena itu, dibutuhkan algoritma yang melakukan pembangunan *minimum spanning tree* dengan proses algoritma yang minimum. Pada dua algoritma yang disebut, algoritma Prim merupakan algoritma yang cocok untuk konteks struktur graf manajemen uang.

Algoritma Prim akan membangun *minimum spanning tree* dengan proses *node by node* yang merupakan algoritma paling

minimum proses di antara kedua algoritma yang telah disebut. Algoritma Kruskal tidak cocok pada konteks struktur graf manajemen uang karena pada prosesnya diperlukan pendataan setiap sisi yang memungkinkan dan *sorting* terhadap bobot dari sisi-sisi tersebut. Kedua proses ini memakan waktu banyak, terutama pada pendataan semua sisi dikarenakan struktur graf yang memungkinkan untuk memiliki banyak simpul. Oleh karena itu, algoritma Prim adalah metode algoritma yang akan digunakan karena lebih efektif dan efisien.

## IV. IMPLEMENTASI DALAM BAHASA PEMROGRAMAN C

### A. Struktur Data Graf Manajemen Uang

```
#ifndef grafkeuangan_H
#define grafkeuangan_H

#include <stdlib.h>
#include "boolean.h"
#include "listlinier.c"

typedef struct np *AddressNode;
typedef struct prio
{
    List bobot;
    char priorityTag[100];
} PrioStruct;

typedef struct np
{
    PrioStruct prio;
    AddressNode next;
} NodePriority;

typedef AddressNode Graf;
#define FIRSINODE(G) (G)

AddressNode newPriorityNode(PrioStruct p);
/*Mengembalikan alamat dari alokasi struktur Graph*/

void createGraph(Graf *G);
/*Membuat Graf kosong, Graf kosong ditandai dengan elemen pertamanya kosong/bernilai NULL*/

boolean isEmptyGraph(Graf G);
/*Mengembalikan true jika Graf Kosong*/

void addPriorityNode(Graf *G);
/*Menambahkan alokasi PriorityNode baru pada Graf*/

void printPriorityTag(Graf G, int index);
/*Menambahkan alokasi PriorityNode baru pada Graf*/

void addDecision(Graf *G, int index);
/*Menambahkan decision pada isi struktur prioritas, basically insertLast buat struktur List bobot*/

int getPengeluaranMin(Graf G);
/*Mengembalikan nilai bobot minimum dari list bobot*/

void printMinDecisions(Graf G);
/*Menampilkan keputusan-keputusan yang harus diambil agar meminimalisir pengeluaran*/

#endif
```

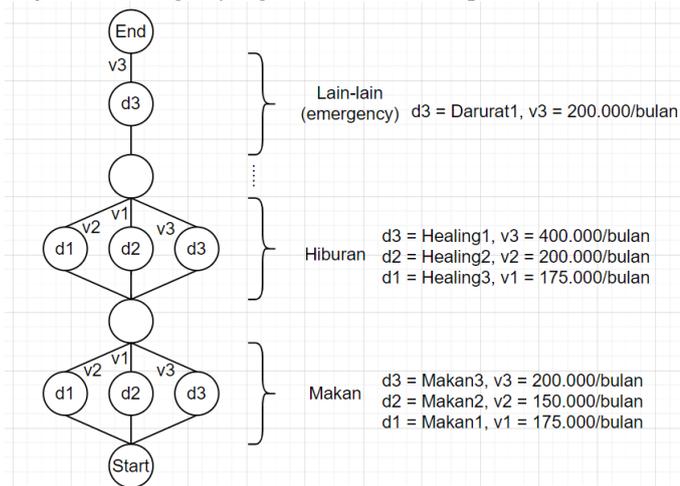
Gambar 4.1 Struktur Data Graf Keuangan

Graf dibangun menggunakan implementasi list berkait yang mengandung elemen berupa struktur tipe data *PrioStruct* dan alamat simpul berikutnya. *PrioStruct* tersusun atas tipe struktur data *List* dan *priorityTag*. Tipe data *List* merupakan list berkait yang berisi *decisionTag* (string) yang berfungsi untuk menyimpan tag dari setiap keputusan dan elemen integer untuk menyimpan nilai bobot. *PriorityTag* (string) memiliki fungsi yang sama seperti *decisionTag* namun untuk setiap node prioritas. Terdapat fungsi-fungsi primitif yang digunakan untuk mempermudah proses program yang sudah ditampilkan di gambar 4.1 beserta penjelasannya. File-file lengkapnya akan dilampirkan di bagian **Lampiran**.

## V. STUDI KASUS

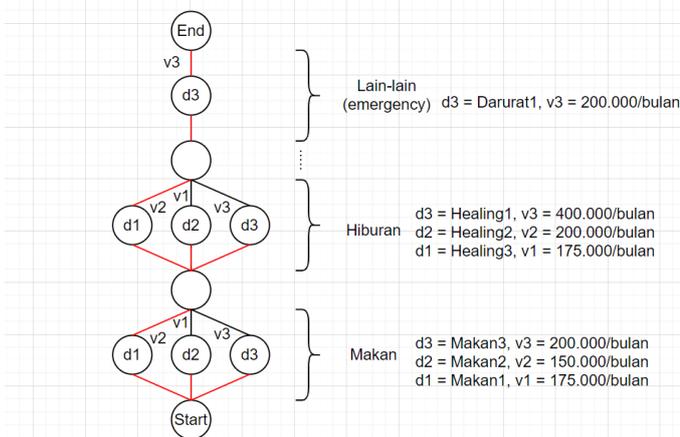
Terdapat seorang mahasiswa bernama Asep, setiap akhir bulan Asep selalu saja makan mie instan dan telur. Hal ini terjadi karena Asep tidak menerapkan disiplin manajemen uang. Suatu hari Asep sudah muak karena dia enek (ini baku) makan mie setiap akhir bulan dan memutuskan untuk menerapkan disiplin manajemen uang. Ia menggunakan graf untuk mempermudah

proses pencarian pengeluaran minimum dengan mencari *minimum spanning tree* dari graf tersebut. Berikut adalah graf manajemen keuangan yang didesain oleh Asep:



Gambar 5.1 Studi kasus Asep

Dengan membangun *minimum spanning tree* berikut adalah pohon yang terbentuk (garis merah):



Gambar 5.2 *minimum spanning tree*

Dengan mencari jalur yang dapat menuju simpul end, Asep dapat menentukan pengeluaran minimum per bulannya, yaitu sebesar 525.000(150.000 + 175.000 + 200.000) / bulan.

Berikut adalah implementasi pencarian pengeluaran minimum dalam programnya.

```

Input decision tag: Makan3
Input bobot : 150
Input decision tag: Makan2
Input bobot : 175
Input decision tag: Makan1
Priority Hiburan
Input bobot : 400
Input decision tag: Healing3
Input bobot : 200
Input decision tag: Healing2
Input bobot : 175
Input decision tag: Healing1
Priority Lain-lain
Input bobot : 200
Input decision tag: Darurat1
Berikut adalah keputusan-keputusan yang perlu diambil untuk meminimalisir pengeluaran Anda :
Makan2, harga : 150
Healing1, harga : 175
Darurat1, harga : 200
525 selamat mengirit~ :D
    
```

Gambar 5.3 Program

Dapat dilihat bahwa program menampilkan pilihan-pilihan yang harus diambil Asep untuk meminimalisir pengeluaran per bulan yaitu sebesar 525 (satuan 1000).

Program menggunakan algoritma Prim dengan terlebih dahulu membandingkan semua nilai yang ada di dalam struktur prioritas. Berikut adalah algoritma Prim yang diimplementasikan ke dalam programnya:

```

void printMinDecisions(Graf G)
/*Menampilkan keputusan-keputusan yang harus
diambil agar meminimalisir pengeluaran*/
{
    AddressNode g = G;
    printf("Berikut adalah keputusan-
keputusan yang perlu diambil untuk
meminimalisasi pengeluaran Anda :\n");
    while (g != NULL)
    {
        printf("%s, ", getDecisionTag(g-
>prio.bobot, indexOf(g->prio.bobot,
getMin(g->prio.bobot)));
        printf("harga : %d\n", getMin(g-
>prio.bobot));
        g = g->next;
    }
    printf("%d selamat mengirit~ :D",
getPengeluaranMin(G));
}
    
```

## VII. KESIMPULAN

Kesimpulannya adalah bahwa kita hidup di zaman dengan perkembangan teknologi yang sangat pesat. Ilmu pengetahuan merupakan dasar yang perlu dimiliki oleh semua orang untuk menyelesaikan masalah-masalah yang ada di kehidupan sehari-hari. Dengan mengimplementasikan teknologi, umat manusia dapat menyelesaikan masalah-masalah yang sangat kompleks demi kesejahteraan. Sebagai mahasiswa Teknik Informatika, salah satu ilmu pengetahuan yang dipelajari adalah ilmu matematika. Dengan mengimplementasikan ilmu matematika, kita dapat menyelesaikan masalah-masalah dengan mudah melalui prinsip-prinsip matematika. Graf dan Tree merupakan salah dua cabang ilmu matematika yang digunakan dalam makalah ini sebagai dasar dari solusi permasalahan manajemen uang. Sebagai mahasiswa kita juga perlu pandai dalam menerapkan disiplin manajemen uang, tapi terkadang terdapat suatu permasalahan manajemen uang yang cukup kompleks sehingga diperlukan semacam media pembantu. Makalah ini menggunakan konsep pembangunan *Minimum Spanning Tree* dari *Weighted Graph* yang disusun secara struktural sesuai dengan kebutuhan perancangan manajemen uang. Hasil dari pengeluaran minimum merupakan jalur dari simpul awal mula pohon (root) menuju *child* tujuan. Dengan menghitung akumulasi nilai pada jalur tersebut, kita bisa menentukan pengeluaran minimum beserta keputusan-keputusan finansial yang perlu diambil. Penyelesaian ini saya implementasikan ke bahasa pemrograman c, dengan tujuan mempermudah(malas).

Kompleksitas program cukup rendah berdasarkan bahasa pemrograman, yaitu  $O(n^2)$ . Kompleksitas ini dikarenakan program melakukan *nested loops* sebanyak dua buah.

## VIII. LAMPIRAN

<https://github.com/Akmal2205/GrafKeuangan>

(Struktur lengkap program)

## REFERENCES

- [1] Munir, Rinaldi. 2023. Matematika Diskrit (Edisi 3). Bandung: Informatika Bandung.
- [2] Munir, Rinaldi. 2023. Graf (Bagian 1). Diakses melalui <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/19-Graf-Bagian1-2023.pdf> Bandung: Informatika Bandung.
- [3] Munir, Rinaldi. 2023. Graf (Bagian 2). Diakses melalui <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/20-Graf-Bagian2-2023.pdf> Bandung: Informatika Bandung.
- [4] Munir, Rinaldi. 2023. Graf (Bagian 3). Diakses melalui <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/23-Pohon-Bag2-2023.pdf> Bandung: Informatika Bandung.
- [5] Munir, Rinaldi. 2023. Pohon (Bagian 1). Diakses melalui <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/23-Pohon-Bag2-2023.pdf> Bandung: Informatika Bandung.
- [6] Munir, Rinaldi. 2023. Pohon (Bagian 2). Diakses melalui <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/23-Pohon-Bag2-2023.pdf> Bandung: Informatika Bandung.

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 3 Desember 2023



Muhammad Syarafi Akmal 13522076