# Identitas

|  |
| --- |
| ***Kajian***  Graf dan Pohon  ***Topik***  Implementasi Graf dan Pohon dalam dunia IT  ***Referensi***   1. Munir, R., Matematika Diskrit, Edisi kedua, untuk Infomatika, Bandung, 2003 2. Rosen, K. H., Discrete Mathematics and Its Applications, 5th edition, McGraw-Hill, Singapore, 2003   ***Kompetensi Utama***   1. Mahasiswa mampu menyelesaikan kasus yang terkait jalur terpendek dengan menggunakan algoritma djikstra dan algoritma “hapus” 2. Mahasiswa mampu menyelesaikan kasus yang berkaitan dengan pewarnaan graf 3. Mahasiswa mampu menyesaikan kasus yang berkaitan dengan kode huffman   ***Lama Pengerjaan***  100 Menit  ***Jenis Pengerjaan \*(bisa dipilih lebih dari 1)***  □Individu □Kelompok □Mandiri □Terbimbing |
|
|
|
|
|

# Pertanyaan Pendahuluan

1. Sebutkan contoh penggunaan graf dalam kehidupan sehari-hari (4 contoh) yang erat kaitannya dengan pencarian jalur terpendek!
2. Sebutkan langkah-langkah menentukan jalur terpendek dengan menggunakan algoritma djikstra!
3. Sebutkan langkah-langkah menentukan jalur terpendek dengan menggunakan algoritma “hapus”!
4. Sebutkan contoh penggunaan graf dalam kehidupan sehari-hari (3 contoh) yang erat kaitannya dengan pewarnaan graf!
5. Sebutkan langkah-langkah menentukan bilangan kromatik dalam pewarnaan graf!
6. Sebutkan contoh penggunaan graf dalam kehidupan sehari-hari (2 contoh) yang erat kaitannya dengan kode huffman!
7. Sebutkan langkah-langkah membuat sebuah kode Huffman!

# Ringkasan Teori

**Lintasan Terpendek (*Shortest Path)***

* Graf berbobot (*weighted graph*)
* Lintasan terpendek: lintasan yang memiliki total bobot minimum.

Contoh aplikasi:

* Menentukan jarak terpendek/waktu tempuh tersingkat/ongkos termurah antara dua buah kota
* Menentukan waktu tersingkat pengiriman pesan (*message*) antara dua buah terminal pada jaringan komputer.

**Algoritma Djikstra**

* Algoritma Dijkstra adalah sebuah prosedur iterasi yang mencari lintasan terpendek antara a dan z dalam graf dengan pembobot.
* Prosesnya dengan cara mencari panjang lintasan terpendek dari sebuah simpul pendahulu dan menambahkan simpul-simpul tersebut ke set simpul S.
* Algotirma berhenti setelah mencapai simpul z.

**Contoh**

Tentukan lintasan terpendek dari **A** ke **Z**



**Solusi**

* Mulai dari simpul A (lingkari) sebagai simpul awal
* Tentukan jalur dengan bobot terpendek yang menghubungkan A dengan simpul yang lain. Jika jalurnya lebih dari satu, pilih jalur dengan bobot terendah
  + ~~AB (4)~~
  + AC (2)
* Karena AC bobotnya lebih rendah maka lingkari simpul C.
* Dari simpul C terdapat beberapa jalur, yaitu:
  + CB (1)
  + ~~CD(8)~~
  + ~~CE(10)~~
* Karena CB bobotnya lebih rendah maka lingkari simpul B
* Dari simpul B, terdapat beberapa jalur, yaitu:
  + ~~BA (4)~~ 🡪 tidak mungkin karena setiap jalur hanya boleh dilewati satu kali, jika jalur ini digunakan maka ada jalur yang akan dilewati lebih dari satu kali
  + BD(5)
* Karena jalur BD yang di pilih maka lingkari simpul D
* Dari simpul D terdapat beberapa jalur, yaitu:
  + ~~DC(8)~~
  + DE(2)
  + ~~DZ(6)~~
* Karena DE bobotnya lebih rendah, maka linkari simpul E
* Dari simpul E, ada beberapa jalur, yaitu:
  + EC (10) 🡪 tidak mungkin dipilih
  + EZ (3)
* Karena jalur EZ yang di pilih dan Z merupakan simpul tujuan maka setiap simpul yang dipilih dalam langkah tersebut akan menjadi jalur terpendek dari A 🡪 Z



* Jadi jalur terpendek adalah ACBDEZ dengan bobot 13

**Algoritma “Hapus”**

Algoritma hapus merupakan salah satu algoritma atau cara untuk memperoleh jalur terpendek dari sebuah graf berbobot. Langkah-langkah yang dilakukan untuk menggunakan algoritma hapus adalah sebagai berikut.

1. Tentukan simpul awal
2. Hapus, sisi-sisi dengan bobot paling tinggi dengan syarat jika sisi-sisi ini dihapus graf awal tidak terbagi menjadi dua bagian atau lebih.
3. Proses penghapusan sisi selesai setelah tidak ada lagi sisi yang dapat di hapus.

**Contoh**

Tentukan lintasan terpendek dari **A** ke **Z**



**Solusi**

* Simbol awal (A)
* Hapus secara berurutan sisi CE (10), CD (8), DZ(6), AB(4) 🡪 sisi BD (5) tidak dapat dihapus setelah sisi DZ dihapus karena akan menyebabkan graf terbagi menjadi 2 bagian.

|  |  |
| --- | --- |
|  |  |
|  |  |

**Pewarnaan Graf**

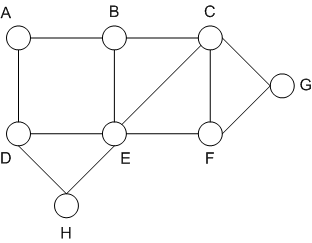
* Sebuah pewarnaan dari graph G adalah sebuah pemetaan warna-warna ke simpul-simpul dari G sedemikian hingga simpul relasinya (yang bertetangga) mempunyai warna warna yang berbeda.
* **Bilangan kromatik** dari G adalah jumlah warna minimum yang diperlukan untuk mewarnai graph G, dilambangkan dgn χ(G) (chi G)
* Untuk mendapatkan bilangan kromatik dapat menggunakan Algoritma Welch Powel
* Algoritma Welch-Powell adalah sebuah cara efisien untuk mewarnai sebuah graph G

**Langkah Algoritma Welch-Powell :**

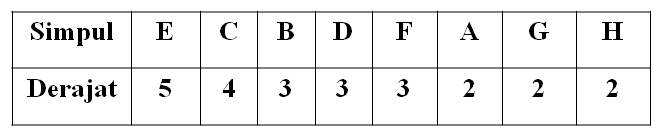
* Urutkan simpul-simpul G dalam derajat yang menurun. Urutan ini mungkin tidak unik karena beberapa simpul mempunyai derajat sama
* Gunakan satu warna untuk mewarnai simpul pertama (yang mempunyai derajat tertinggi) dan simpul-simpul lain (dalam urutan yang berurut) yang tidak bertetangga dengan simpul pertama.
* Mulai lagi dengan dengan simbul dengan derajat tertinggi dan ulangi proses pewarnaan simpul yang tidak berwarna sebelumnya dengan menggunakan warna kedua.
* Terus ulangi dengan penambahan warna sampai semua simpul telah diwarnai

**Contoh**

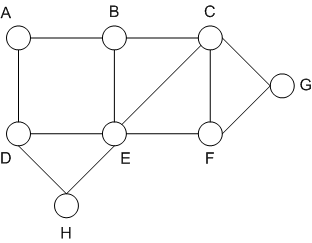
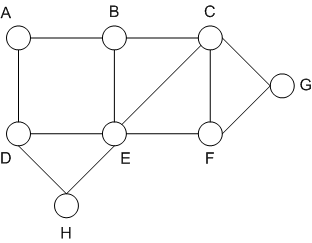
Tentukan banyaknya bilangan kromatik pada graf berikut!

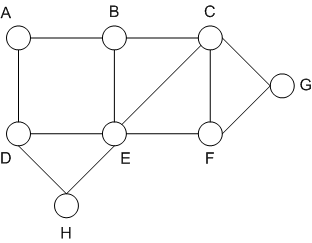
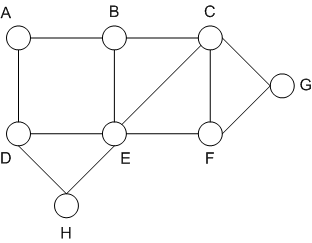
****

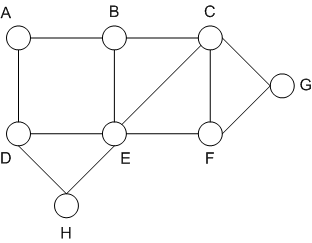
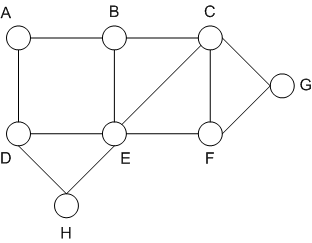
**Solusi**

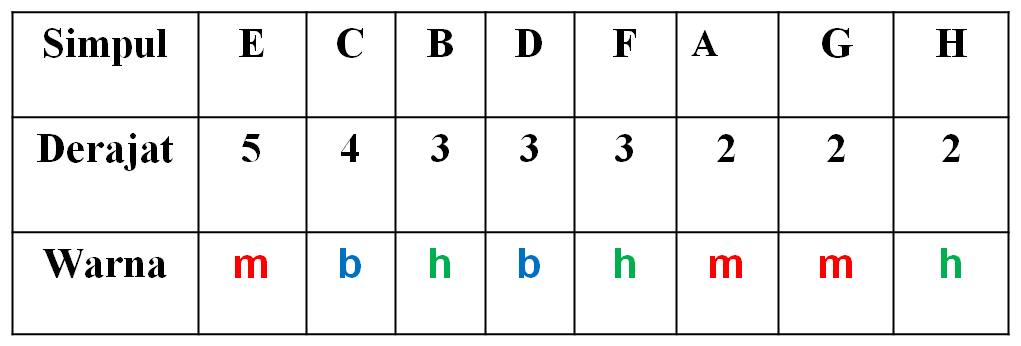
Tentukan derajat masing-masing simpul

* d(A) = 2 ; d(B) = 3 ; d(C) = 4 ;
* d (D) = 3; d(E) = 5 ; d(F) = 3 ;
* d(G) = 2 ; d (H) = 2

****

****

****

****



**Pohon Biner**

* Adalah pohon *n-ary* dengan *n* = 2.
* Pohon yang paling penting karena banyak aplikasinya.
* Setiap simpul di dalam pohon biner mempunyai paling banyak 2 buah anak.
* Dibedakan antara anak kiri (*left child*) dan anak kanan (*right child*)

**Terapan Pohon Biner**

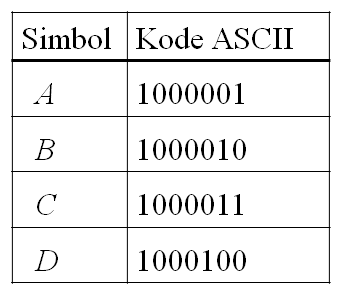
* Kode Awalan
* Kode awalan adalah himpunan kode (misalnya kode biner) sedemikian sehingga tidak ada anggota kumpulan yang merupakan awalan dari anggota lainnya
* Contoh:
  + { 000, 001, 01, 10, 11} : himpunan kode awalan
  + { 1, 00, 01, 000, 0001 } : bukan himpunan kode awalan
* Kode awalan mempunyai pohon biner yang bersesuaian
* Sisi diberi 0 atau 1
* Semua sisi kiri diberi label 0 saja (atau 1 saja)
* Semua sisi kanan diberi label 1 ( atau 0 saja)
* Barisan sisi-sisi yang dilalui oleh lintasan dari akar ke daun menyatakan kode awalan

**Terapan Pohon Biner**

* Kompresi Data
* Bagaimana sebenarnya program kompresi data itu bekerja?
* Mengapa sebuah file yang ukurannya sudah baku ternyata bisa diperkecil tanpa mengurangi isinya?
* Misalkan kita hendak menyimpan sebuah huruf A. Komputer akan mengenali huruf A ini sebagai karakter dengan urutan ke-65, maka huruf A disimpan di harddisk sebagai 100 0001 atau merupakan kode biner dari angka 65.
* Berarti dibutuhkan 7 digit biner untuk menyimpan huruf A.
* Kompresi data berusaha menyimpan sebuah data dengan jumlah digit biner seminimum mungkin.
* **ASCII** (*American Standard Code for Information Interchange*)
* Character encoding ASCII terdiri atas 95 karakter yang bisa dicetak dan 33 karakter perintah. Jadi total terdapat 128 karakter yang diwakili dengan 7 bit bilangan biner
* Rangkaian bit untuk string ‘*ABACCDA’* dengan menggunakan kode standar ASCII

01000001010000010010000010100000110100000110100010001000001

atau 7 × 8 = 56 bit (7 *byte*).



* Dengan menggunakan **Kode Huffman** rangkaian string string ‘*ABACCDA’* dapat di kompresi menjadi hanya 13 bit (0110010101110). Bagaimana caranya
* Hitung kekerapan kemunculan setiap simbol dalam teks
* Untuk yang kekerapannya lebih tinggi diberi nilai 1
* Untuk yang kekerapannya sama jika:
  + Keduanya adalah karater tunggal: urutan alafabet (a-z) yang lebih dulu disebut maka diberi nilai 0, lainnya 1
  + Salah satunya adalah karakter tunggal sedangkan yang lain karakter gabungan 🡪 karakter tunggal diberi nilai 0, yang lainnya 1
  + Kedua-duanya karakter gabungan 🡪 jumlah urutan alphabet (gabungan) yang lebih rendah diberi nilai 0, yang lainnya 1
* Bentuk pohon biner (pohon Huffman) dengan langkah-langkah berikut:
* Pilih dua simbol dengan peluang paling kecil
* Kombinasikan kedua simbol tersebut sehingga diperoleh karakter baru
* Pilih simbol lain kemudian kombinasikan dengan karakter baru yg dihasilkan pada point b
* Iterasi poin c sehingga seluruh simbol telah dipilih

**Contoh**

Tentukan Kode Huffman untuk **TELKOMSEL**

**Solusi**

Kekerapan

|  |  |
| --- | --- |
| **Huruf** | **Kekerapan** |
| **K** | **1/9** |
| **M** | **1/9** |
| **O** | **1/9** |
| **S** | **1/9** |
| **T** | **1/9** |
| **E** | **2/9** |
| **L** | **2/9** |

# Studi Kasus (INDIVIDU)

1. Jika diketahui graf berbobot berikut ini!



Dengan menggunakan algoritma Djigstra dan Algoritma “Hapus” tentukan jalur terpendek dari

1. A 🡪 Z
2. Z 🡪 A
3. B 🡪 G
4. F 🡪 C
5. Tentukan banyaknya bilangan kromatik dari graf berikut ini!

|  |  |
| --- | --- |
|  |  |
|  |  |
|  | 1. asa |

1. Tentukan kode hufman dari:
2. MATEMATIKA
3. POLTEK TELKOM
4. GIVING AND CARING