PROGRAM STUDI

S1 SISTEM KOMPUTER

UNIVERSITAS DIPONEGORO

Algoritma dan Pemrograman
Pendekatan Pemrograman Modular

Oky Dwi Nurhayati, ST, MT

email: okydn@undip.ac.id

Pendahuluan

Teknik Pemrograman

Penekanan pada pemrograman terstruktur

- Struktur dasar

  Menggunakan *flow chart* dan *pseudocode*

Menggunakan sistem modular.

- Program dibuat dalam bentuk modul-modul untuk fungsi tertentu maupun subroutine tertentu.
- Modul-modul dikendalikan oleh modul utama (program utama)

Pendahuluan

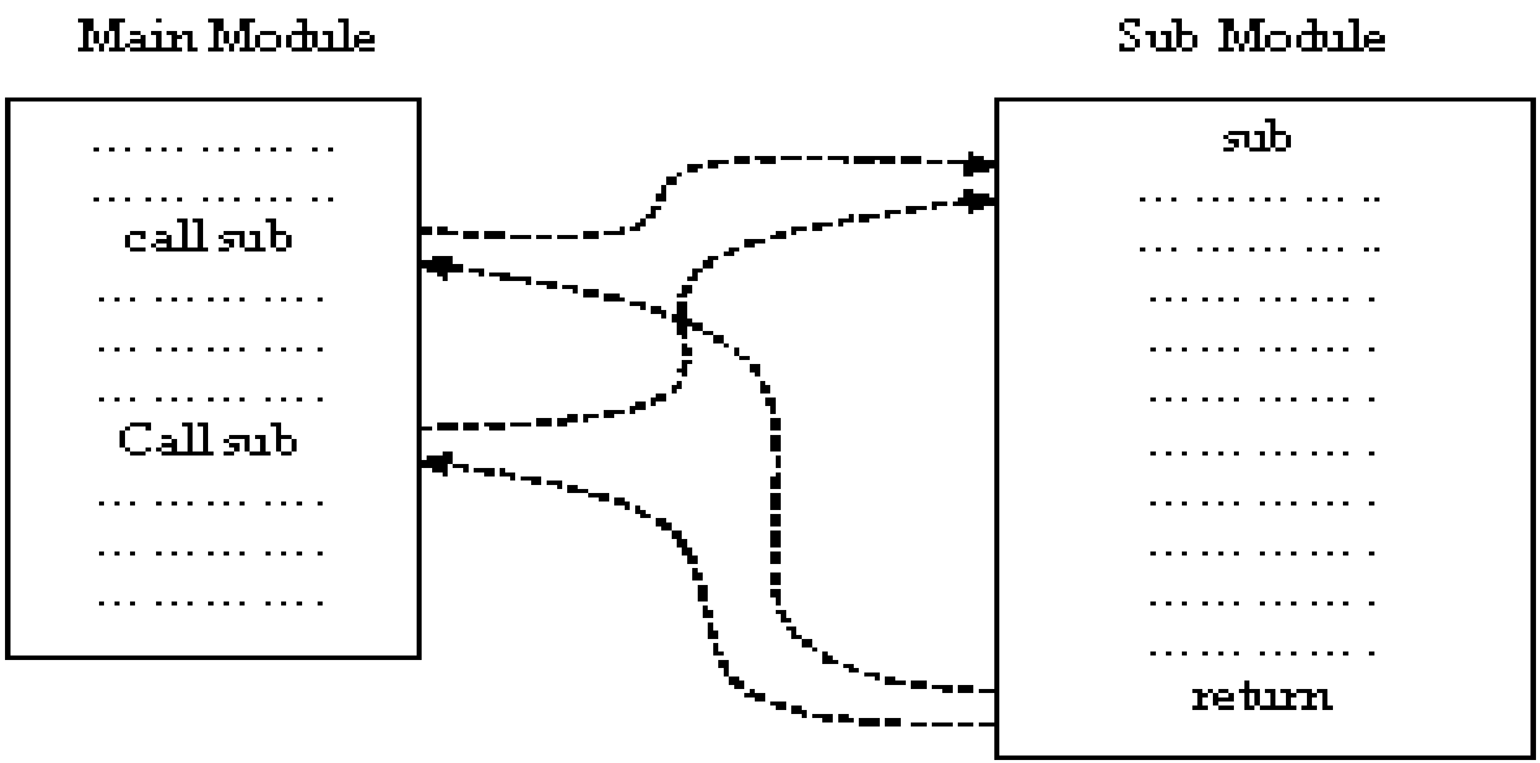
Teknik Pemrograman

- Modul bersifat independen (tidak ada modul yang dapat akses langsung ke modul lain, kecuali modul pemanggil dan submodulnya sendiri)
- Modul dapat diubah secara radikal tanpa mempengaruhi modul lain sejauh fungsi modul tidak berubah..

Pendahuluan

Teknik Pemrograman

Implementasi pemrograman modul menggunakan subroutine-subroutine digambarkan sebagai berikut;



Gambar 1. Main module & Sub Module

Pendahuluan

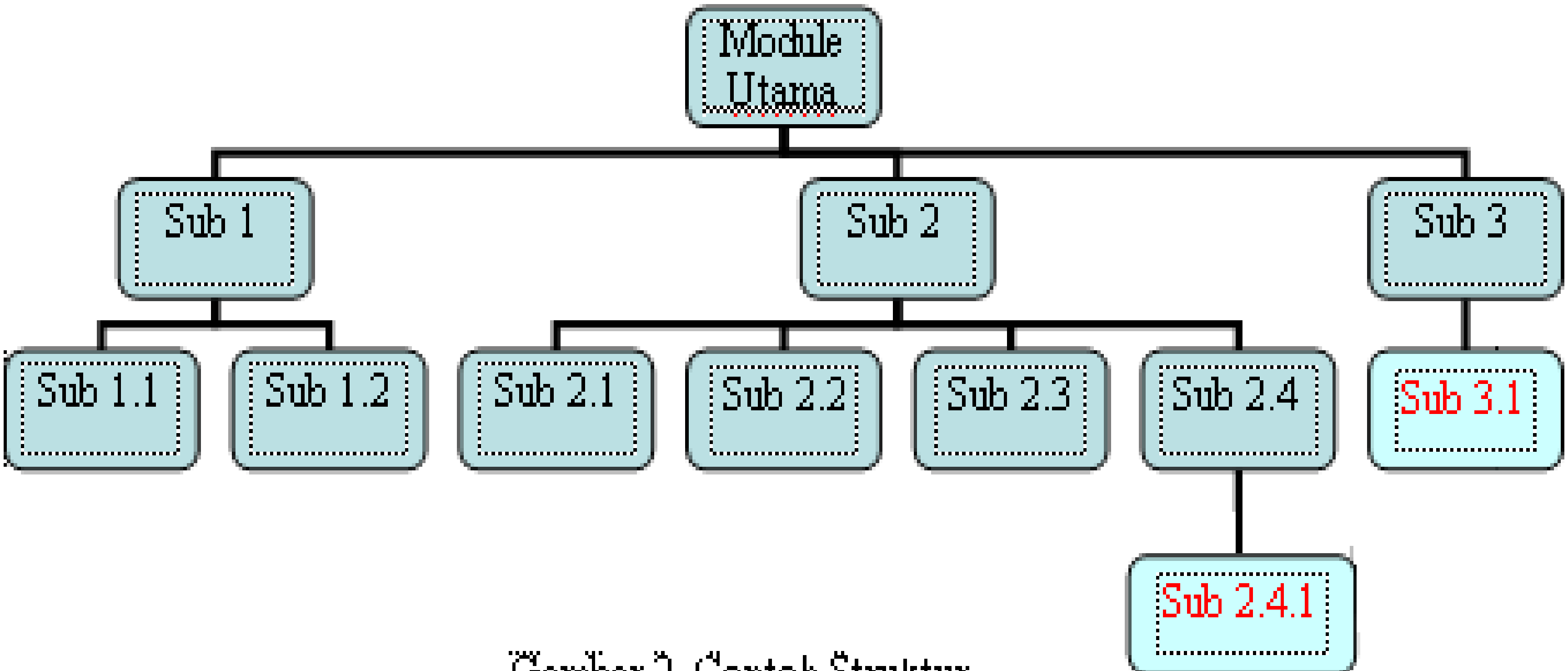
Bentuk sub module

- Internal subroutine; berupa bagian dari program yang menggunakannya
- External subroutine;
  - bukan bagian dari program yang menggunakan modul itu.
  - Modul tersimpan dalam *library* dalam bentuk "*object module*" yang siap digunakan oleh modul-modul yang akan menggunakan.
  - Modul dapat digunakan untuk tugas lebih dari satu program
- Dalam menggunakan, pemrograman perlu mengetahui di mana diperoleh, namanya, bagaimana mengirim data padanya, dan jabawan baliknya.

PENDEKATAN PEMROGRAMAN MODULAR

MODEL TOP-DOWN

- modul utama tunggal
- sub modul dari suatu modul minimum dua. Bila hanya satu ditijau kembali seperti contoh berikut ;



Gambar 2. Contoh Struktur Pemrograman Modular

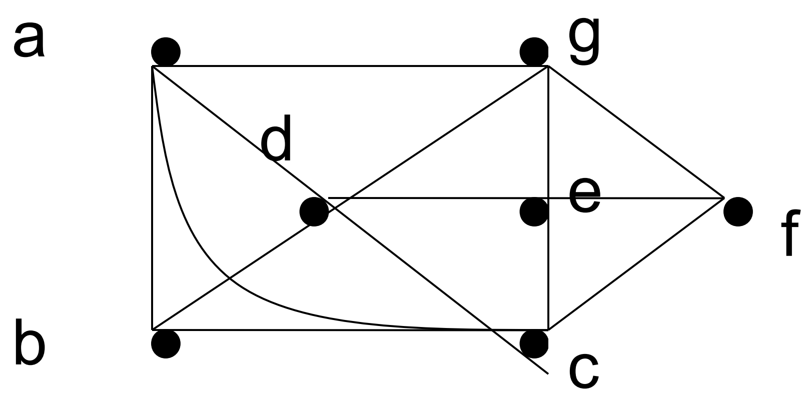
Masalah path minimum (*Shortest path problem*)	**mencari route dengan jarak terpendek dalam suatu jaringan transportasi.**
Masalah aliran maksimum (*maximum flow problem*)	**menghitung volume aliran BBM dari suatu reservoir ke suatu titik tujuan melalui jaringan pipa.**
Masalah pencariah dalam graph (*graph searching problem*)	**mencari langkah-langkah terbaik dalam program permainan catur komputer.**
Masalah pengurutan topologis (*topological ordering problem*)	**menentukan urutan pengambilan mata-mata kuliah yang saling berkaitan dalam hubungan prasyarat (prerequisite).**
Masalah jaringan tugas (*Task Network Problem*)	**membuat penjadwalan pengerjaan suatu proyek yang memungkinkan waktu penyelesaian tersingkat.**
Masalah pencarian pohon rentang minimum (*Minimum Spanning Tree Problem*)	**mencari rentangan kabel listrik yang totalnya adalah minimal untuk menghubungkan sejumlah kota.**
Travelling Salesperson Problem	**tukang pos mencari lintasan terpendek melalui semua alamat penerima pos tanpa harus mendatangi suatu tempat lebih dari satu kali.**
Four-color problem	**dalam menggambar peta, memberikan warna yang berbeda pada setiap propinsi yang saling bersebelahan.**

Teori Graf

Siklus Hamilton (Hamiltonian cycle) adalah siklus dalam graf G yang mengandung setiap verteks di G tepat satu kali.

Contoh 10.1 :

Untuk graf G = (V,E) dalam Gambar 10.1, carilah sebuah siklus Hamilton.



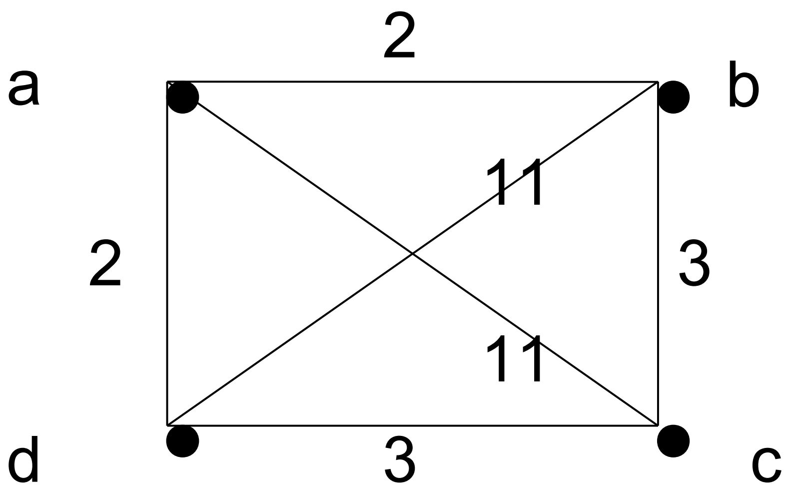
Penyelesaian :

Sebuah siklus Hamilton untuk graf G adalah siklus (a, b, c, d, e, f, g, a).

Masalah Perjalanan Wiraniaga

- Soal perjalanan wiraniaga berkaitan dengan masalah pencarian siklus Hamilton dengan panjang minimum dalam sebuah graf berbobot G. Jika kita menganggap verteks-verteks dalam graf berbobot sebagai kota dan bobot rusuk sebagai jarak, masalah perjalanan wiraniaga adalah mencari sebuah rute terpendek sehingga wiraniaga tersebut dapat mengunjungi setiap kota satu kali, berawal dan berakhir pada kota yang sama.
- **Contoh 10.2 :**

Selesaikanlah masalah perjalanan wiraniaga untuk graf berikut.



Kita lihat bahwa siklus Hamilton (a, b, c, d, a) dan (a, d, c, b, a) merupakan siklus Hamilton dengan panjang minimum untuk G.

Penyelesaian :

Siklus Hamilton	Panjang
(a, b, c, d, a)	2+3+3+2 = 10
(a, b, d, c, a)	2+11+3+11 = 27
(a, c, b, d, a)	11+3+11+2 = 27
(a, c, d, b, a)	11+3+11+2 = 27
(a, d, b, c, a)	2+11+3+11 = 27
(a, d, c, b, a)	2+3+3+2 = 10

PENDEKATAN PEMROGRAMAN MODULAR

GRAPH

Graph terdiri dari simpul-simpul dan lintasan-lintasan.

- Simpul

  VERTEX : dapat mewakili peristiwa kata, sasaran, dan sebagainya.

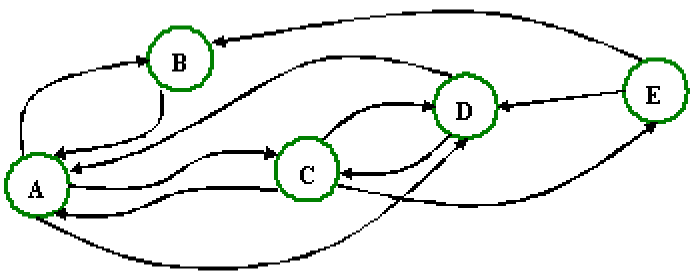
- Lintasan
  - Lintasan i j : menghubungkan simpul i dan j
  - Lintasan bearah ij : menunjujkkan lintasan yang dilewati 9ditunjukan dengan anak panah) dari simpul i menuju j
  - Lintasan langsung i j: lintasan dari i ke j tanpabmelewati simpul yang lain.
  - Lintasan tak langsung ij : lintasan dari I ke j dengan melewati simpul lain.

PENDEKATAN PEMROGRAMAN MODULAR

GRAPH

3. Derajat simpul (vertex) :

- k : menunujukkan cacah lintasan yang terhubung dengan simpul yang bersangkutan.
- Derajat simpul masuk k1 : menunjukkan cacah lintasan yang masuk menuju simpul
- Derajat simpul keluar k2 : cacah lintasan yang keluar



PENDEKATAN PEMROGRAMAN MODULAR

GRAPH

Graph dapat mewakili :

- jaringan transportasi
- jaringan kegiatan suatu proyek dan lain-lain

Lintasan dapat mewakili : biaya operasi, jarak antara simpul (kota), profit, dan sebagainya. Besaran dalam dimensinya masing-masing

PENDEKATAN PEMROGRAMAN MODULAR

GRAPH

Graph dapat direpresentasikan dalam bentuk matrik lintasan

ke / dari	A	B	C	D	E
A	0	1	1	1	0
B	1	0	0	0	0
C	1	0	0	1	1
D	1	0	1	0	0
E	0	1	0	1	0

1 = ada lintasan
0 = tidak ada

simpul y Graph dapat direpresentasikan dalam bentuk matrik lintasan ang bersangkutan

→ 1 = TRUE
0 = FALSE

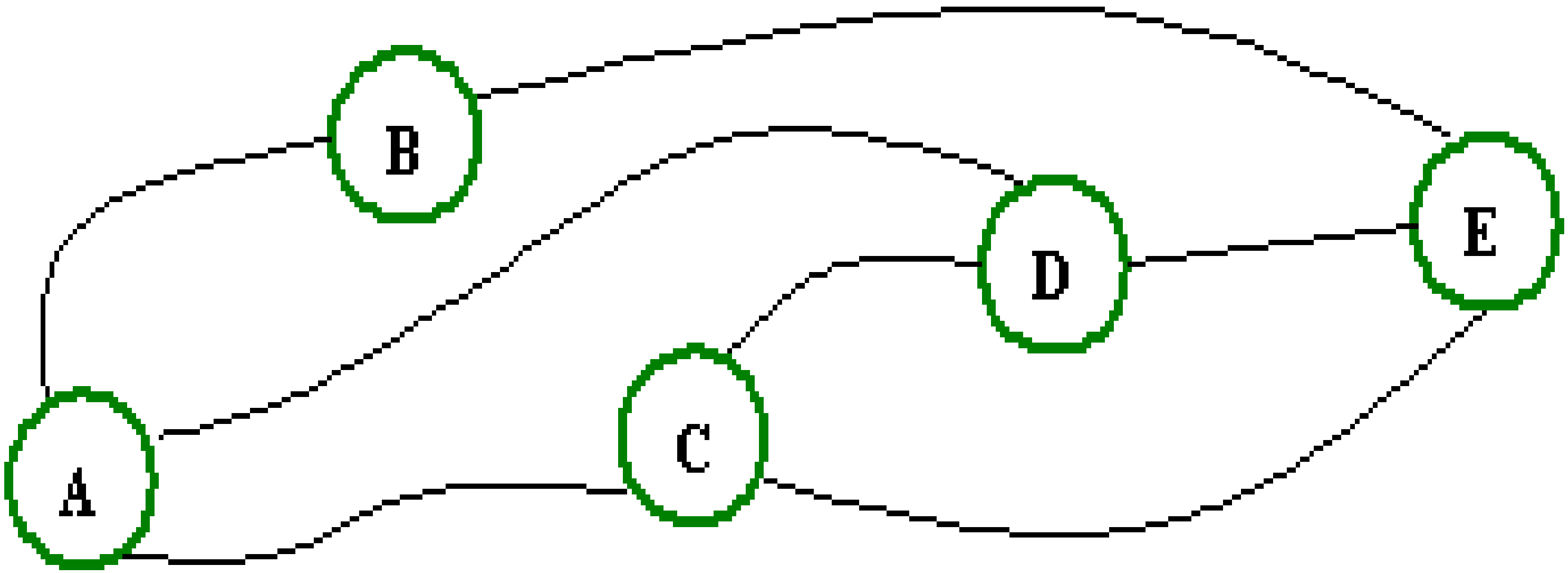
PENDEKATAN PEMROGRAMAN MODULAR

GRAPH

Graph dengan lintasan berarah = directed graph = diagraph

Graph dengan lintasan tak berarah = undirect graph = undiagraph

Disini hanya muncul ada atau tidak hubungan antara dua vertex (dalam undirect graph (undiagraph) lintasan tak berarah, matriks lintasannya simetris)



PENDEKATAN PEMROGRAMAN MODULAR

GRAPH

ke / dari	A	B	C	D	E
A	0	1	1	1	0
B	1	0	0	0	0
C	1	0	0	1	1
D	1	0	1	0	0
E	0	1	0	1	0

Matrik lintasan tak berarah

1 = ada hubungan

0 = tidak ada hubungan

atau vertex itu sendir

PENDEKATAN PEMROGRAMAN MODULAR

GRAPH

bobot lintasan menunjukkan harga besaran yang diwakilinya. Dalam banyak kasus nilai dalam lintasan menunjukkan harga besaran yang diwakilikinya.

Misalnya untuk masalah transportasi ada unsur-unsur matriks lintasan menunjukkan biaya (dalam satuan mata uang). Untuk vertex *i* ke *j* bila tidak ada lintasannya, biaya = ∞ dan bila *i* biaya = 0

dari \ ke	A	B	C	D	E
A		100	39	200	
B	100				150
C	39			300	250
D	200		300		350
E		150	250	350	

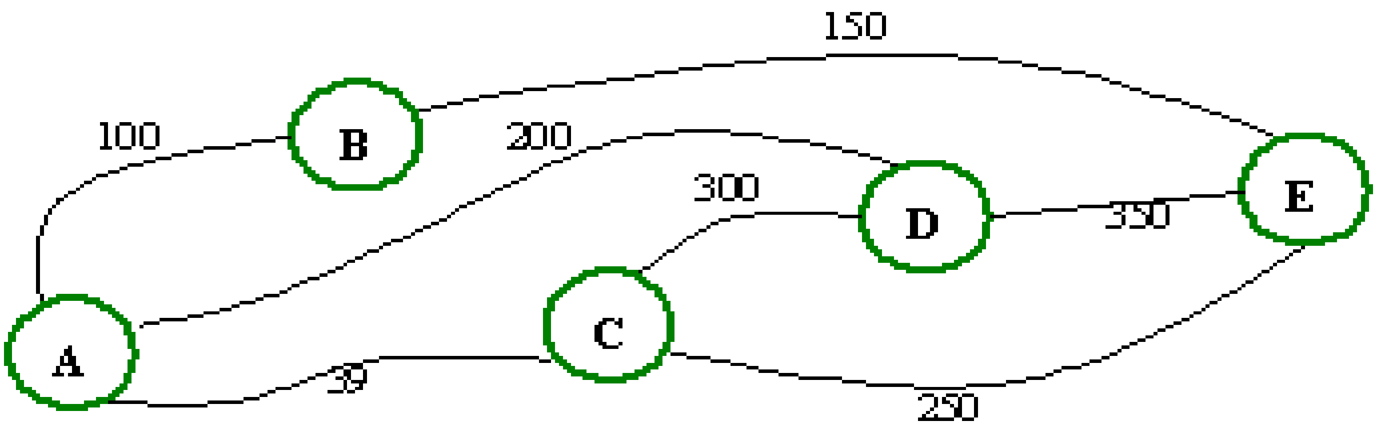
Nilai 0 dan ~
Tak ditulis

PENDEKATAN PEMROGRAMAN MODULAR

GRAPH

Dalam lintasan berarah, bobot lintasan yang dicantumkan dalam matrik lintasan hanya bila ada lintasan (matrik lintasan mungkin tidak simetris).

→ matrik lintasan mungkin tidak simetris

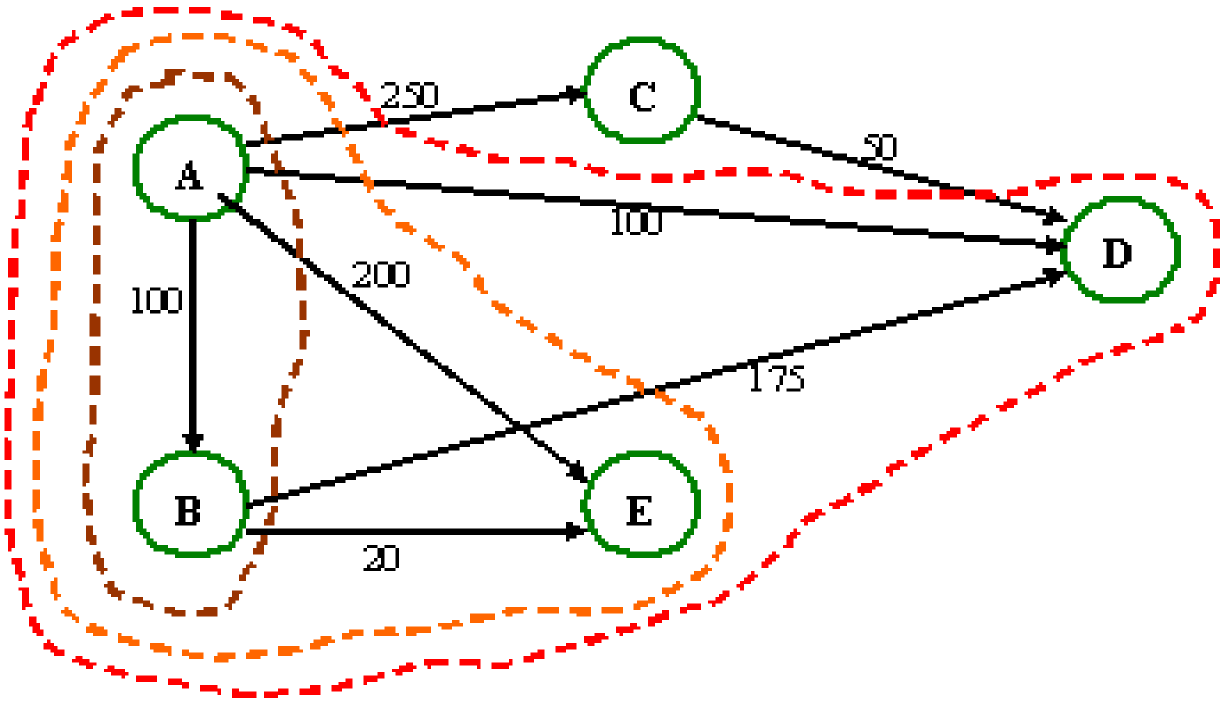


PENDEKATAN PEMROGRAMAN MODULAR

GRAPH

Contoh kasus :

- mencari lintasan terpendek atau biaya termurah dari suatu perjalanan
- perjalanan diawali dari salah satu vertex, menuju kepada semua vertex lain



PENDEKATAN PEMROGRAMAN MODULAR

GRAPH

Algoritma untuk menemukan lintasan terpendek dikenalkan oleh **E.Dijkstra**

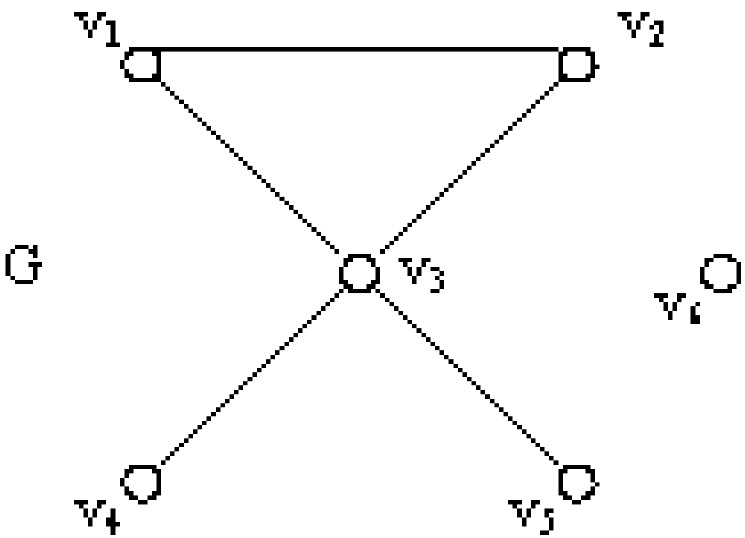
3. Awali pada lintasan (i) pada simpul (vo),

   lintasan terpendek dari vo-v1. lingkup S→Vo.

5. Pilih vertex (v-s) sedemikian sehingga lintasan (v) minimu dari semua lintasan dalam (v-s)
6. Untuk setiap u dalam 1-s, diperbarui lintasan (u) = min, lintasan u, lintasan w, lintasan (v,w)
7. Ulangi langkah 2

PENDEKATAN PEMROGRAMAN MODULAR

GRAPH



$$
\begin{array}{c|cccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 0 & 1 & 1 & 0 & 0 & 0 \\
2 & 1 & 0 & 1 & 0 & 0 & 0 \\
3 & 1 & 1 & 0 & 1 & 1 & 0 \\
4 & 0 & 0 & 1 & 0 & 0 & 0 \\
5 & 0 & 0 & 1 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

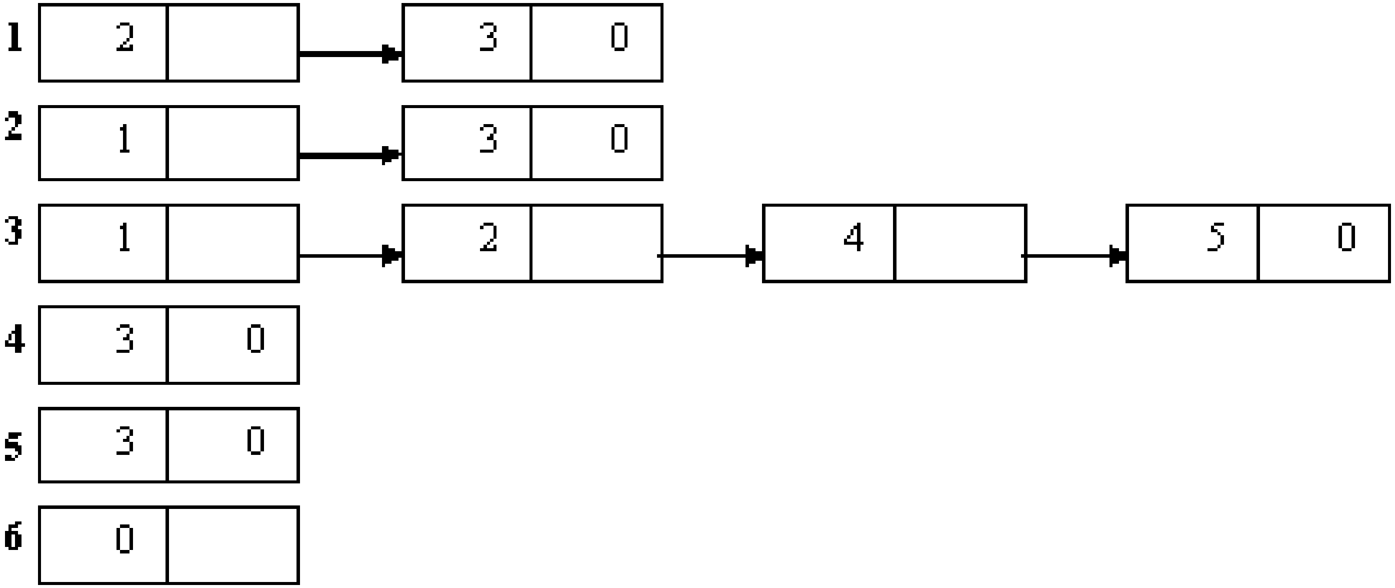
Matrix lintang

A

PENDEKATAN PEMROGRAMAN MODULAR

GRAPH

Representasi dalam linklist



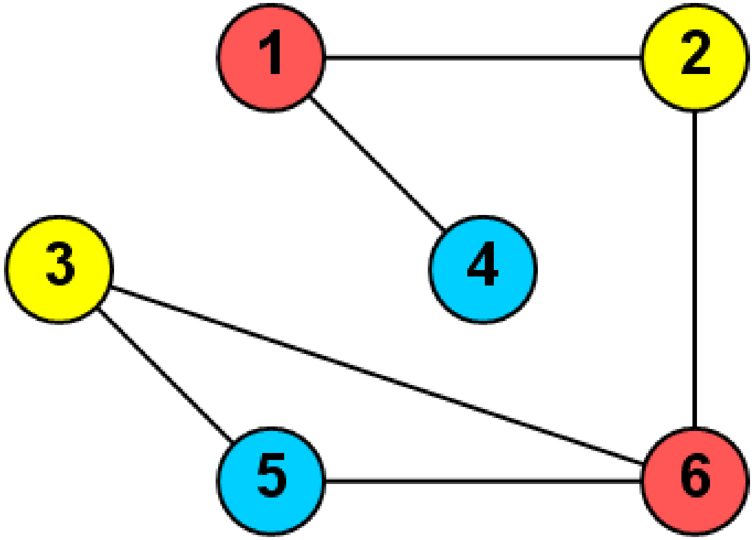
GRAPH COLORING

- Graph coloring adalah problem klasik algoritma tentang bagaimana caranya mewarnai graph dengan warna berbeda untuk setiap node yang ”berdekatan”
  - Berdekatan berarti ada edge yang menghubungkan kedua node tersebut
  - Nilai adjacency-nya lebih besar dari 0
- Tantangan dari problem ini adalah bagaimana caranya mengusahakan agar jumlah warna yang diperlukan seminimal mungkin.

VARIASI PROBLEM

- Edge coloring
  - Bukan node yang diwarnai melainkan edge-nya. Sejumlah edge yang bertemu di node tertentu tidak boleh diberi warna yang sama.
- Region coloring
  - Mewarnai sebidang wilayah yang terbagi atas sub-wilayah kecil. Setiap sub-wilayah yang memiliki perbatasan tidak boleh diberi warna yang sama. Problem ini lebih terkenal dengan sebutan map coloring.
  - Map coloring dapat diselesaikan dengan mengubahnya menjadi graph, mewarnai graph, kemudian dipetakan kembali.

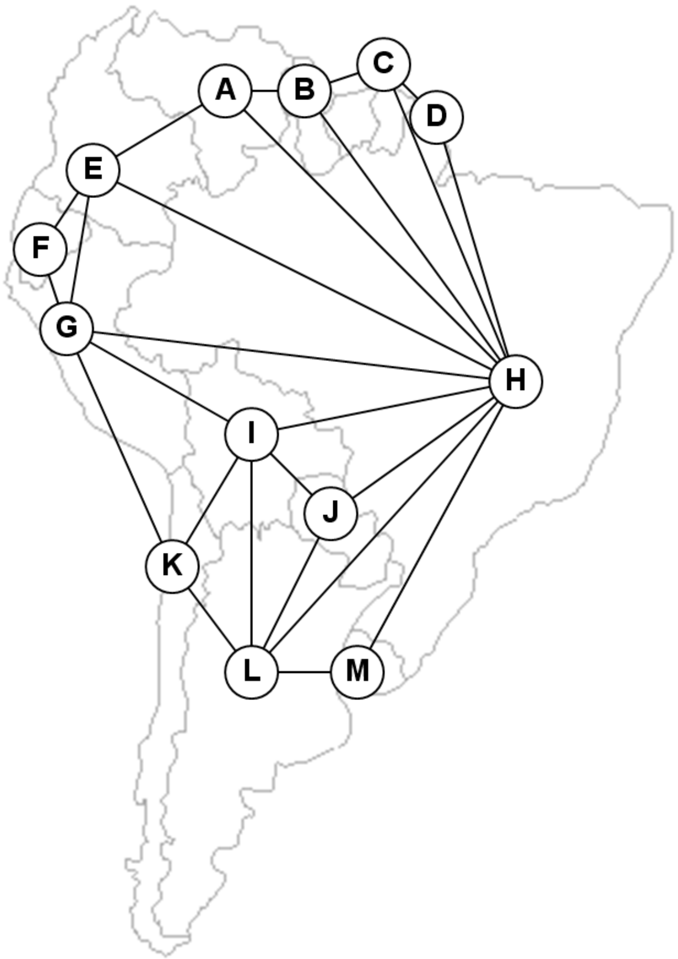
CHROMATIC NUMBER



- Diperlukan 3 warna untuk mewarnai graph di atas
- Jumlah warna minimal yang harus dipakai untuk mewarnai sebuah graph disebut chromatic number.

MAP COLORING

KONVERSI MAP KE GRAPH

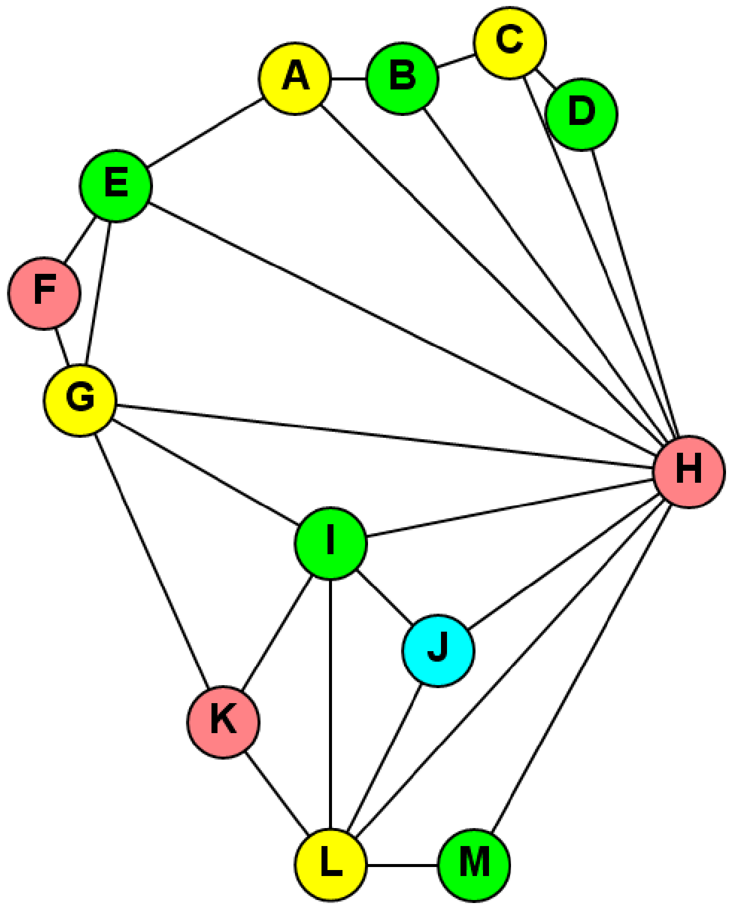


ALGORITMA WELSH & POWELL

1. Urutkan node dalam sebuah graph berdasarkan jumlah edge yang terhubung padanya, secara descending (dari besar ke kecil)
2. Berdasarkan urutan di atas, warnai semua node dalam graph dengan warna 1 jika sebuah node tidak berbatasan dengan node yang sudah berwarna 1
3. Ulangi proses ini untuk warna 2, warna 3, dst hingga semua node telah diberi warna

- Merupakan salah satu contoh algoritma Metode Greedy
  - Hasilnya belum tentu optimal
  - Penyelesaian secara optimal adalah NP-Complete problem
- Baca bab 6.6 di buku untuk penjelasan detil langkah per langkah jalannya algoritma Welsh & Powell

HASIL GRAPH COLORING



MAP COLORING

Algoritma Prim

Sebuah algoritma dalam teori graf untuk mencari pohon rentang minimum untuk sebuah graf berbobot yang saling terhubung. Ini berarti bahwa sebuah himpunan bagian dari edge yang membentuk suatu pohon yang mengandung node, di mana bobot keseluruhan dari semua edge dalam pohon diminimalisasikan. Bila graf tersebut tidak terhubung, maka graf itu hanya memiliki satu pohon rentang minimum untuk satu dari komponen yang terhubung. Algoritma ini ditemukan pada 1930 oleh matematikawan V.Jarnik dan kemudian secara terpisah oleh computer scientist Robert C.Prim pada 1957 dan ditemukan kembali oleh Dijkstra pada 1959. Karena itu algoritma ini sering dinamai algoritma DJP atau algoritma Jarnik.

PENDEKATAN PEMROGRAMAN MODULAR

Algoritma Prim

Untuk menentukan minimum spanning tree

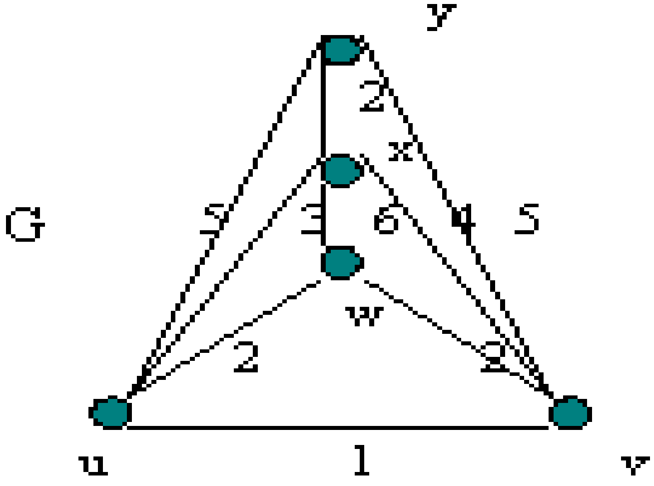
Algoritmanya:

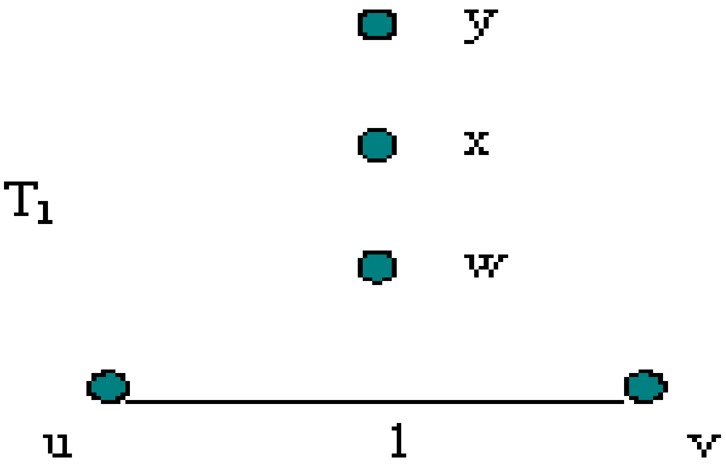
- Inisialisasi tree T
- T diperbaharui bila lintasan adalah minimum yang menghubungkan vertex, dan vertex tak di T, maka T←T+e
- Bila E(T) = p – 1 (p= cacah vertex maka output adalah E(T)). Bila tidak, ke langkah 2.

PENDEKATAN PEMROGRAMAN MODULAR

<u>Algoritma Prim</u>.

Contoh :

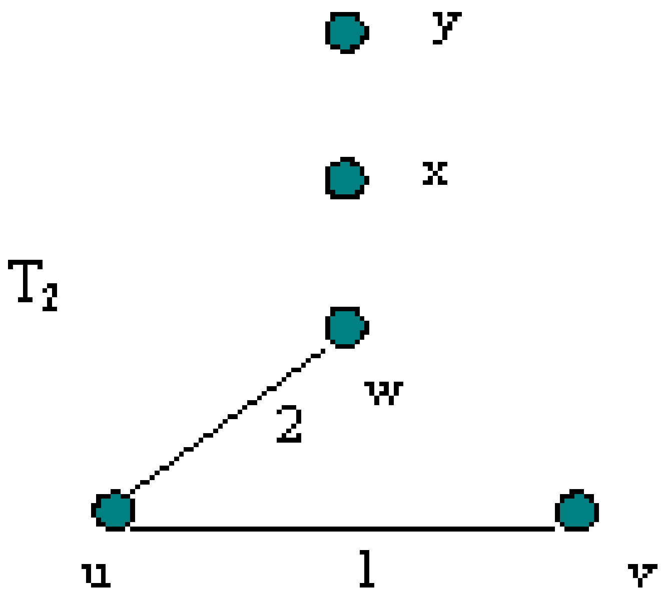


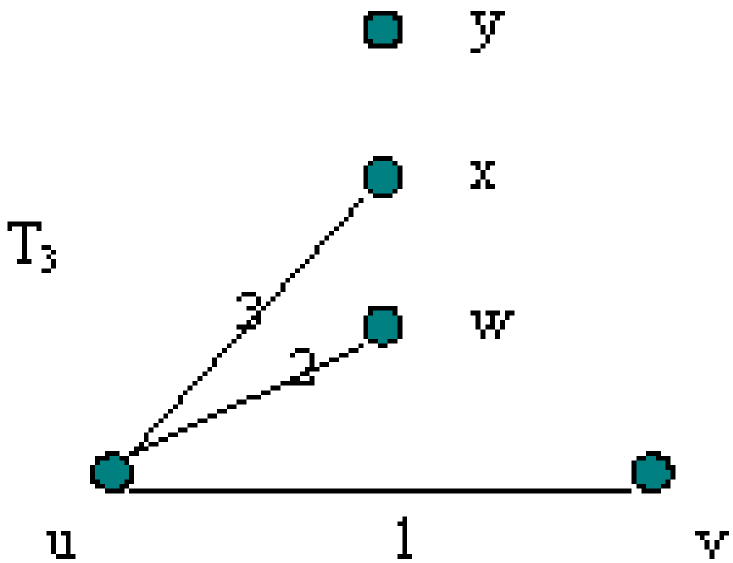


PENDEKATAN PEMROGRAMAN MODULAR

Algoritma Prim.

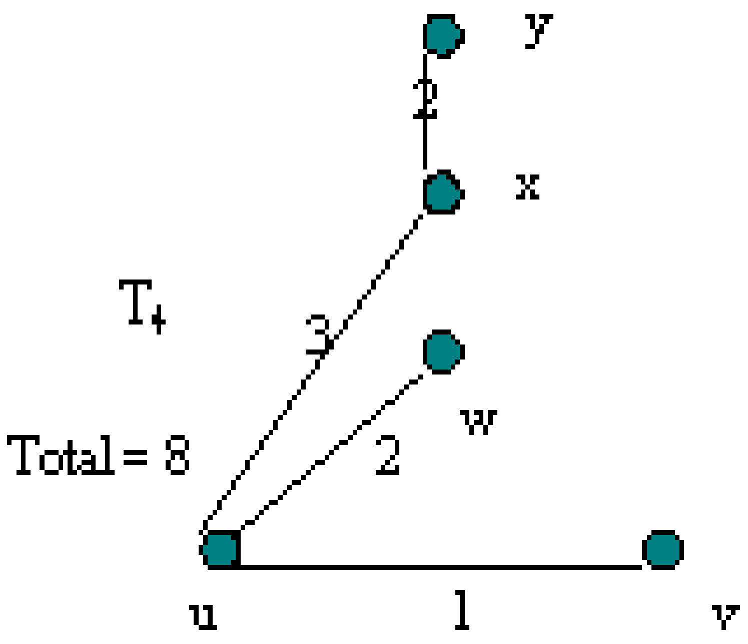
Contoh :

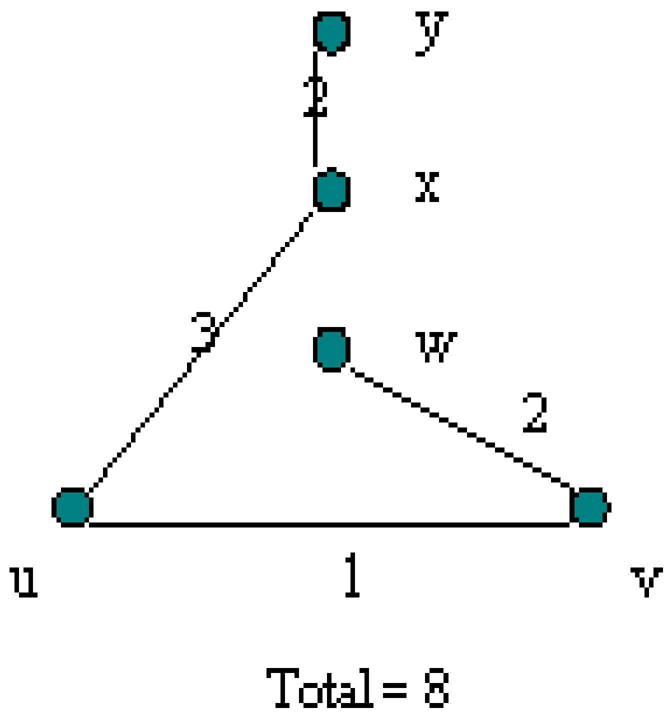




PENDEKATAN PEMROGRAMAN MODULAR

Algoritma Prim.





PENDEKATAN PEMROGRAMAN MODULAR

Algoritma *Bouruvka*

Untuk menentukan minimum spanning tree

Algoritmanya:

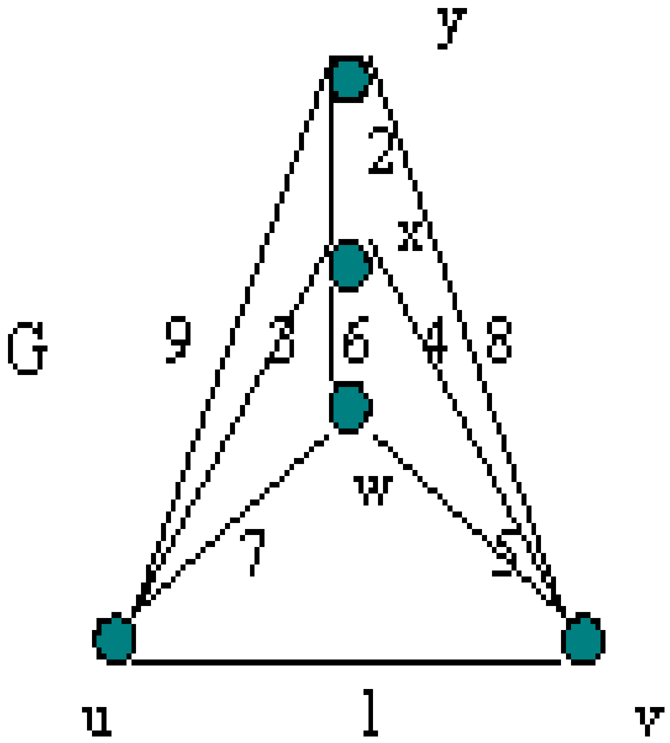
4. **Inisialisasi spanning Forest . F→Fp**
5. **Forest F diperbaharui. Untuk tiap komponen F' dari hubungan vertex F' ke vertex lain dari F dengan lintasan minimum.**

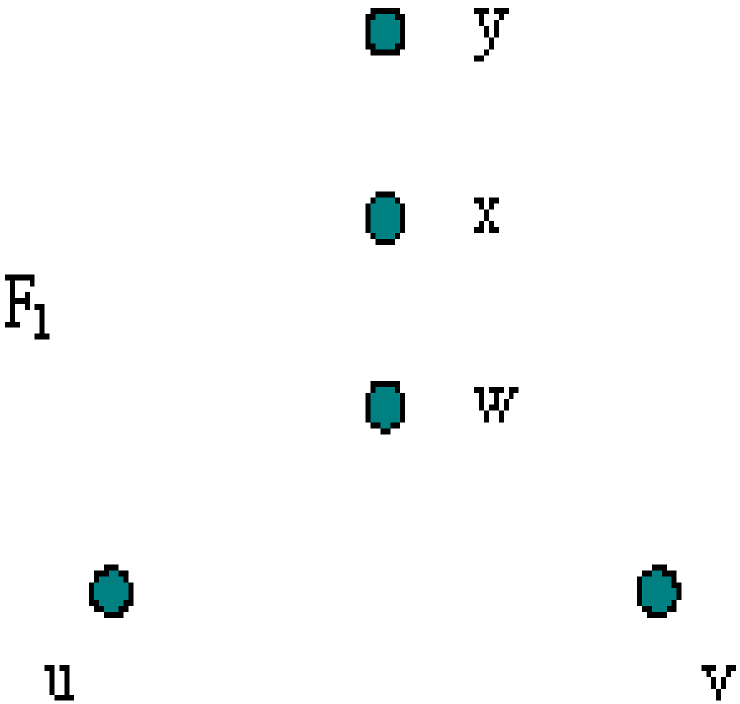
   Set lintasan adalah S, sehingga F → F+ S

7. **Lintasan |E(F)|=p-1 [p cacah vertex maka output E(F), sebaliknya ke langkah 2.**

PENDEKATAN PEMROGRAMAN MODULAR

<u>Algoritma *Bouruvka*</u>





PENDEKATAN PEMROGRAMAN MODULAR

Algoritma *Bouruvka*

C