BAB II

TINJAUAN PUSTAKA

2. LANDASAN TEORI

Landasan teori adalah sebuah konsep dengan pernyataan yang tertata rapi dan sistematis memiliki variabel dalam penelitian karena landasan teori menjadi landasan yang kuat dalam penelitian yang akan dilakukan. Oleh karena itu dengan menciptakan landasan teori yang baik dalam penelitian akan menjadi salah satu hal terpenting, karena landasan teori menjadi sebuah landasan dalam penelitian itu sendiri.

2.1 Teori Graf

a. Graf G didefinisikan sebagai pasangan himpunan (V,E) yang dalam hal ini, V adalah himpunan tidak kosong dari titik-titik dan E adalah himpunan sisi yang menhgubungkan sepasang titik.

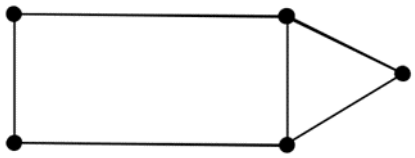
$$V = \{v1, v2, ... vn\} \quad (1)$$

$$E = \{e_1, e_2, ..., e_n\} \quad (2)$$

Atau dapat ditulis singkat

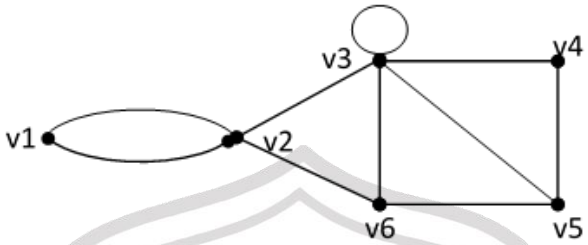
$$G = (V,E) \quad (3)$$

Secara umum graf dapat digambarkan dengan suatu diagram dengan titik ditunjukkan sebagai titik yang dinotasikan dengan v_i, $i = 1, 2, \ldots, n$ dan sisi digambarkan dengan sebuah garis lurus atau garis lengkung yang menghubungkan dua buah titik (v_i, v_j) dan dinotasikan dengan e_k.



Gambar 2.1 Graf 5 titik dan 6 sisi

b. Loop adalah sisi yang berawal dan berakhir pada titik yang sama, sedangkan sisi paralel adalah dua sisi atau lebih berbeda yang menghubungkan dua buah titik v_i dan v_j yang sama.



Gambar 2.2 Graf dengan 6 titik dan 10 sisi

Pada **gambar 2.2** dapat dilihat bahwa e_4 adalah sebuah loop dan e_1 serta e_2 adalah dua buah sisi yang paralel.

c. Graf sederhana adalah graf yang tidak memuat loop dan sisi-sisi paralel. Misalkan $V = (v_1, v_2, v_3, v_4, v_5)$ dan $E=(e_1, e_2, e_3, e_4, e_5, e_6)$, maka $G=(V,E)$ adalah graf sederhana yang dapat dilihat pada Gambar 2.1.

d. Suatu sisi e_k

Dalam suatu graf G dengan titik-titik ujung v_i dan v_j disebut saling incident dengan v_i dan v_j, sedangkan v_i dan v_j ini disebut dua buah titik yang saling adjacent. Jika kedua sisi tersebut incident pada suatu titik persekutuan, maka dua buah sisi e_k dan e_m disebut saling adjacent. Pada Gambar 2.2 dapat dilihat bahwa e_8, e_7, e_9 adalah tiga buah sisi yang incident dengan v_6, sedangkan e_5 dan e_7 adalah adjacent.

e. Derajat dari sebuah titik v_i dalam graf G adalah jumlah sisi yang incident dengan v_i dengan loop dihitung dua kali. derajat dari sebuah titik v_i biasanya dinotasikan dengan $d(v_i)$.

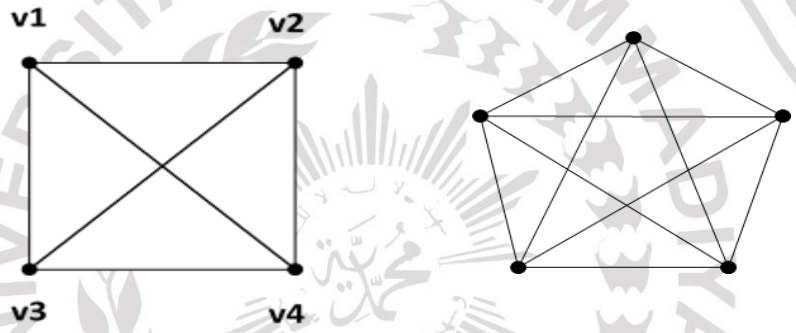
Pada Gambar 2.2 dapat dilihat bahwa $d(v_1) = 2$, $d(v_2) = 4$, $d(v_3) = 5$, $d(v_4) = 2$, $d(v_5) =2$, $d(v_5) = 3$, dan $d(v_6) = 3$.

f. Suatu walk dalam sebuah graf G(V,E) adalah suatu barisan berhingga dari titik dan sisi secara bergantian yang dimulai dan diakhiri dengan titik sehingga setiap sisi incident dengan titik sebelum dan sesudahnya, di mana sebuah sisi hanya dilalui satu kali. Di dalam suatu walk pada sebuah graf dapat terjadi bahwa satu titik dilalui lebih dari satu kali.

g. Suatu Graf Berarah G terdiri dari himpunan titik V(G): {v_1, v_2, …}, himpunan sisi E(G): {e_1, e_2, …}, dan suatu fungsi yang menghubungkan setiap sisi dalam E(G) ke suatu pasangan berurutan titik (v_i,v_j). Jika e_k = (v_i,v_j) adalah suatu sisi dalam G, maka v_i disebut titik awal e_k dan v_j disebut titik akhir e_k. Arah sisi adalah dari v_i ke v_j. (Rasdiana 2015).

2.2 Jenis-jenis Graf

a. Sebuah graf komplit (graf lengkap) dengan n titik, dilambangkan dengan K_n adalah garf sederhana dengan n titik dan setiap dua titik berbeda dihubungkan dengan sebuah sisi. Misalakan graf komplit dengan 4 titik yang disimbolkan dengan K_4 dan graf komplik dengan 5 titik yang disimbolkan dengan K_5



Gambar 2.3 K_4 Graf komplit dengan 4 titik K_5 Graf komplit dengan 5 titik

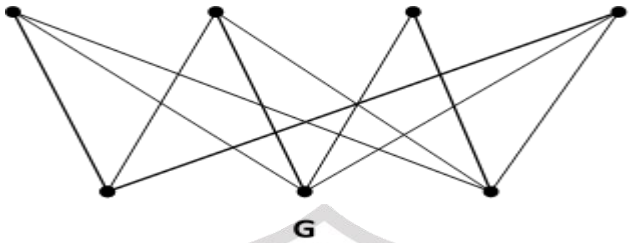
b. Graf yang tidak memili sisi disebut graf kosong atau graf nol. Graf nol dengan titik n dan setiap dua titik berbeda dilambangkan dengan N_n. Misalkan graf kosong dengan 3 titik yang disimbolkan dengn N_3.

N3

Gambar 2.4 graf kosong dengan 3 titik

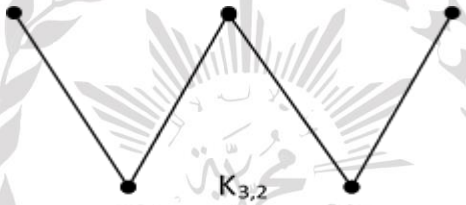
c. Sebuah graf G disebut bipartisi jika himpunan titik G dapat dipartisi menjadi dua himpunan bagian A dan B sedimikan hingga setiap sisi dari G

menghubungkan sebuah titik di A dan sebuah titik di B. Kita sebut (A,B) bipartisi dari G.



Gambar 2.5 Graf G bipartisi

d. Apabila G sederhana dan bipartisi dengan bipartisi (A,B) sedimikan hingga setiap titik di A berhubungan langsung dengan setiap titik di G, maka G disebut graf bipartisi komplit, dilambangkan dengan $K_{m,n}$ dimana |A = m dan |B = n. Sebagai contoh, perhatikan graf pada **gambar 2.6**.

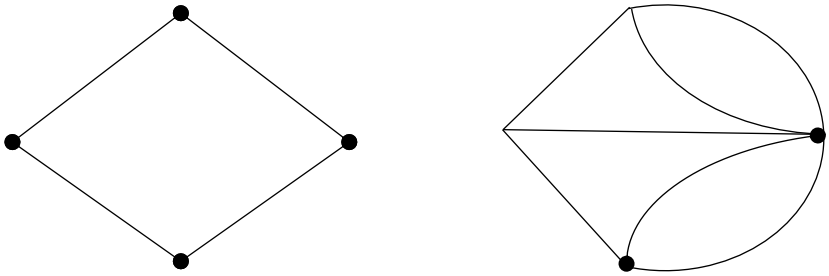


Gambar 2.6 Graf $K_{3,2}$ bipartisi komplit

Perhatikan bahwa, sebagai akibat dari defenisi, graf bipartisi mungkin saja mempunyai sisi rangkap, tetapi tidak mungkin memuat gelung. Begitu juga banyaknya titik graf bipartisi komplit $K_{m,n}$ adalah m+n dan banyaknya sisi adalah mn.[10] (fenny anggraini, sugeng mingparwoto 2015).

2.2.1 Graf tak Berarah (Undireted graf)

Graf yang setiap sisinya tidak mempunyai arah anak panah tetapi memiliki bobot pada setiap sisinya.Urutan pasangan titik yang terhubung oleh sisi tidak diperhatikan. Sehingga (u,v) = (v,u) adalah sisi yang sama. Sehingga graf tak berarah sering dipakai pada jaringan saluran telepon karena sisi pada graf tak berarah menyatakan bahwa saluran telepon dapat beroperasi pada dua arah.



Gambar 2.7 Graf tak berarah

2.2.2 Graf Berarah (directed graf)

Graf berarah merupakan graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah.Konsep graf berarah lebih sering digunakan dibandingkan dengan konsep graf tak berarah.Apabila ruas suatu graf berarah mempunyai suatu bobot, graf berarah tersebut dinamakan suatu jaringan atau *network*.

Gambar 2.8 Graf berarah

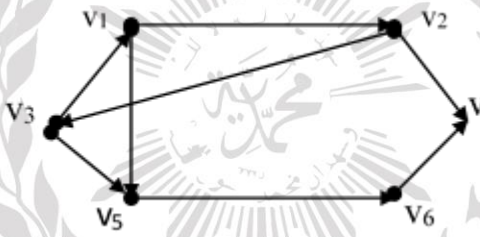
Beberapa Pengertian dalam graf berarah :

1. Derajat ke luar (out degree) suatu titik adalah banyaknya ruas yang mulai / keluar dari titik tersebut.
2. Derajat ke dalam (in degree) suatu titik adalah banyaknya ruas yang berakhir / masuk ke titik tersebut.
3. Titik berderajat ke dalam = 0 disebut sumber (source), sedangkan titik berderajat ke luar = 0 disebut muara (sink).
4. Pengertian Walk, Trail, Path (Jalur) dan Sirkuit (Cycle) berlaku pula pada graf berarah, dimana harus sesuai dengan arah ruas. Kalau tidak sesuai dengan arah ruas-nya, maka disebut sebagai semi walk, semi path atau semi trail.

2.2.3 Path dan Sirkuit (Cycle)

Sebuah Path W dalam sebuah graf berarah G adalah sebuah barisan berganti dari titik-titik dan sisi-sisi berarahnya yang membentuk W = {v_0, e_1, e_2, v_2, . . . , e_n, v_n}, Sedimikan sehingga setiap sisi e_i mulai pada v_{i-1} dan berakhir di v_i. Jika disini tidak ada arti lain, maka kita nyatakan W dengan barisan titiksnya atau barisan sisinya. Ingat bahwa sebuah path dalam sebuah graf berarah bisa juga tertutup, dalam hal dimana titik pertama dan terakhir dalam barisan yang sama. Sebuah path sederhana dalam sebuah graf berarah G adalah sebuah path di G dimana setiap titiknya berbeda.

Sebuah Sebuah cycle dalam graf berarah adalah sebuah path di G dimana semua titiknya berbeda kecuali yang pertama dan terakhir. Path harus memenuhi syarat yang diperlukan pada arah busurnya. Perhatikan graf G pada **gambar 2.9**. Tentukan dua path dari v_1 ke v_6.



Gambar 2.9 Graf G

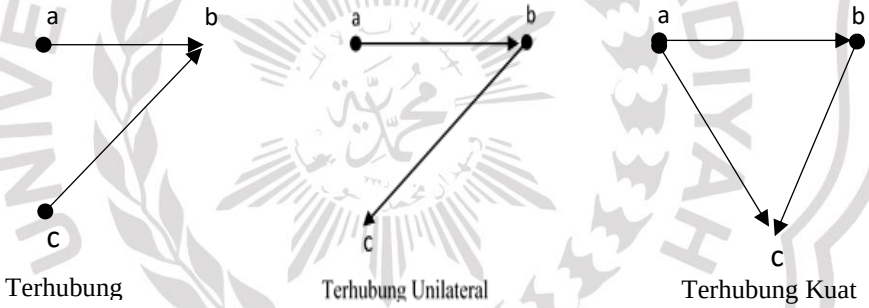
Ada banyak path dari v_1 v_6. Dua diantaranya adalah(v_1, v_5, v_6) dan (v_1, v_2, v_3, v_1, v_5, v_6). Catatan bahwa barisan dari titik-titik v_1, v_2, v_4, v_6 bukan sebuah path yang dari v_4 ke v_6 karena busur yang menghuubungkan v_1 ke v_6 tidak dimula pada v_4 yaitu busur bukan titik dalam arah yang sama seperti path.

2.2.4 Graf Berarah Terhubung

Dua buah titik v_1 dan titik v_2 disebut terhubung jika terdapat lintasan dari v_1 ke v_2. *G* disebut graf terhubung (*connected graf*) jika untuk setiap pasang titik v_i dan v_j dalam himpunan *V* terdapat lintasan dari v_i ke v_j. Jika tidak, maka *G* disebut graf tak terhubung (*disconnected graf*). Graf berarah *G* dikatakan terhubung

jika graf tidak berarahnya terhubung (graf tidak berarah dari *G* diperoleh dengan menghilangkan arahnya).Pada graf berarah terdapat 3 pengertian keterhubungan, yakni :

1. Terhubung lemah, jika terdapat suatu semi path antara setiap 2 titik dari D. Jika *u* dan *v* tidak terhubung kuat tetapi terhubung pada graf tidak berarahnya, maka *u* dan *v* dikatakan terhubung lemah (*weakly connected*).
2. Terhubung unilateral, jika antara setiap 2 titik u dan v dari D, terdapat jalur dari u ke v atau dari v ke u.
3. Terhubung kuat, jika antara setiap 2 titik u dan v dari D, terdapat jalur dari u ke v dan dari v ke u. Dua titik, *u* dan *v*, pada graf berarah *G* disebut terhubung kuat (*strongly connected*) jika terdapat lintasan berarah dari *u* ke *v* dan juga lintasan berarah dari *v* ke *u*. Contoh:



Gambar 2.10 Graf terhubung

2.3 Lintasan Terpendek

Lintasan terpendek adalah lintasan minimum yang diperlukan untuk suatu tempat dari tempat tertentu. Lintasan minimum yang dimaksd dapat dicari dengan menggunakan graf. Graf yang digunakan adalah graf berbobot, Yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Dalam kasus ini yang dimaksud berupa jarak. Dalam hal ini bobot harus bernilai positif, pada lain hal terdapat bobot dengan nilai negatif. Lintasan terpendek dengan titik awal s dan titik tujuan t didefenisikan sebagai lintasan dari s ke t dengan bobot minimum dan berupa lintasan sederhana*(simple path).*

Salah satu aplikasi graf berarah berlabel yang sering dipakai adalah mencari lintasan terpendek diantara 2 titik. Apabila masalahnya adalah mencari lintasan terpendek tetap dapat digunakan dengan cara mengganti nilai sisi.

a. Misalkan G adalah suatu graf, untuk v dan w adalah titik dalam G. suatu Walk dari v ke w adalah barisan titik dan sisi secara berselang-seling, diawali dari titik v dan diakhiri pada titik w. Walk dengan panjang n dari v ke w ditulis : v_0 e_1 v_1 e_2 v_2 … v_{n-2} e_n v_n dengn v_0 = v:v_n = w; v_{i-1} dan v_i adalah titik-titik ujung sisi e_i.Lintasan dengan panjang n dari v ke w adalah walk dari v ke w yang semua sisinya berbeda. Lintasan dari v ke w dituliskan sebagai v = v_0 e_1 v_1 e_2 v_2 … $v_{n-1}e_n$ v_n = w dengan $e_i \neq e_j$untuk i ≠ j. Penulisan berikutnya akan dipergunakan notasi v_1 v_1, A = {v_1 v_2, v_2 v_3, ….. }

b. Lintasan tertutup adalah suatu barisan sisi (e_{i1} ,e_{i2} ,.....,e_{in}) sedemikian rupa sehingga titik terminal $e_{i\,j}$ berimpit dengan titik awal $e_{i(\,j+1)}$ untuk $1 \leq j \leq k - 1$.

Pada Gambar 2.2 terdapat:

1. Pada titik v_1 e_1 v_2 e_3 v_3 e_4 v_3 e_5 v_4 semua sisi berbeda (e_1,e_3,e_4, dan e_5) masing-masing muncul sekali. Ada titik yang berulang (v_3 muncul 2 kali). Titik awal dan titik akhir tidak sama dengan titik awal = $v_1$dan titik akhir = v_4 , Barisan ini merupakan lintasan dari v_1 v_4 dengan panjang 4.
2. Pada titik $v1$ $e1$ $v2$ $e3$ $v3$ $e5$ $v4$ $e5$ $v3$ $e6$ $v5$ ada sisi yang muncul lebih dari sekali, yaitu $e5$ (muncul 2 kali) berarti barisan tersebut merupakan walk dari $v1$ $v5$ dengan panjang lima.

   Terdapat beberapa macam persoalan lintasan terpendek antara lain :

   a. Lintasan terpendek antara dua buah titik tertentyu (a pair shortest path)
   b. Lintasan terpendek antara semua pasangan titik (all pairs shortest path)
   c. Lintasan terpendek dari titik tertentu ke semua titik yang lain (single source shoertest path).

d. Lintasan terpendek antara dua buah titik yang melalui beberapa titik tertentu (intemediae short path). (Gopinda Al Araaf 2014).

2.4 Algoritma Bellman-Ford

Algoritma Bellman-ford adalah algoritma untuk menyelesaikan permasalahan lintasan terpendek dengan dengan sumber tunggal. Jadi algoritma Bellman-ford menghitung jarak terpendek (dari satu sumber) pada sebuah graf berbobot. Maksud dari sumber tunggal ialah bahwa algoritma menghitung semuah jarak terpendek yang berawal dari satu titik. Di samping itu algoritma ini menggunakan d[u] sebagai batas atas dengan jarak d[u,v] dari u ke v. Algoritma ini melakukan inisialisasi jarak titik sumber ke titik nol dan semua titik lainnya (sampai tak hingga). Secara progresif algoritma ini melakukan perbaikan (updating) jarak pada setiap titik sumber ke titik v di dalam V hingga dicapai lintasan dalil Boolean *TRUE* yaitu jika grafik mengandung lingkaran tidak negatif maka titik dapat dicapai dari titik sumber, dan dalam kondisi lain dikatakan Bollean *FALSE*. Algoritma Bellman ford sebagai berikut:

BELLMAN-FORD (G, *w*, *s*)

1. INITIALIZE-SINGLE-SOURCE (G, *s*)
2. for each vertex *i* = 1 to V[G] - 1 do
3. for each edge (u, *v*) in E[G] do
4. RELAX (u, *v*, *w*)
5. For each edge (u, *v*) in E[G] do
6. if d[u] + *w*(u, *v*) < d[*v]* then
7. return FALSE
8. return TRUE

Berikut algoritma Bellman-Ford jika disajikan dalam bentuk notasi matematika:

M [i,v] = min(M [i-1,v] , (M [i-1,n]+ Cvn))

i = iterasi, v = vertex = node, n = node neighbor, C = cost

a. *Langkah-langkah algoritma Bellman-Ford*

Algoritma Bellman-Ford menentukan lintasan terpendek

Secara umum, langkah-langkah algoritmanya adalah sebagai berikut:

a. Menentukan titik asal dan mendaftar semua titik maupun sisi

b. Memberi nilai untuk titik asal sama dengan nol dan titik-titik lainnya dengan tak hingga.

c. Memulai iterasi terhadap semua titik yang ada dimulai dengan titik asal. untuk menentukan jarak dari semua titik yang berhubungan dengan titik asal dengan formula seperti berikut:

U= titik asal

V = titik tujuan

UV = sisi yang menghubungkan U dan V,

Jika jarak V lebih besar dari jarak U+bobot UV maka jarak V diisi dengan jarak U+bobot UV, dilakukan hingga semua titik terjelajahi.

d. Melakukan iterasi untuk semua sisi yang ada untuk mengecek apakah ada siklus negatif dalam graf tersebut, kemudian melakukan pengecekan seperti dibawah ini:

Untuk semua sisi UV, jika jarak V lebih besar dari jarak U+bobob UV maka sudah jelas bahwa graf tersebut memiliki siklus negatif.

b. *Kompsleksitas Waktu algoritma Bellman-Ford*

Secara kompleksitas waktu, algoritma ini hanya memiliki 1 kemungkinan, yaitu dengan notasi *Big-O* diberikan O(V·E). V adalah jumlah titik graf berbobot, lalu E adalah jumlah sisi dalam graf. Oleh sebab itu untuk jumlah sisi ataupun titik yang sangat besar akan menyebabkan algoritma ini akan berjalan lebih lama dibandingkan algoritma Dijkstra.

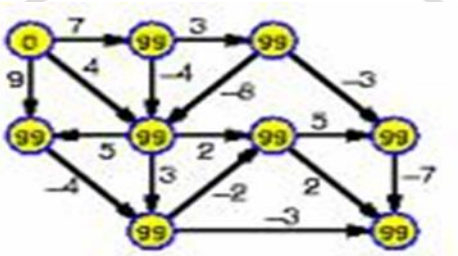
Berikut penjabaran perhitungan kompleksitas waktu.

1. Tahap inisialisasi mempunyai kompleksitas O(V). Maksudnya bahwa, jika n banyaknya titik maka inisialisasi dilakukan sebanyak n dengan titik asal 0 dilakukan satu kali dan titik yang lainnya (n-1).
2. Tahap kedua yaitu melakukan pencarian jalur atau jalan terpendek terhadap suatu titik s mempunyai kompleksitas O(V.E). Hal ini karena

perulangan untuk tahap pencarian ada dua kali, perulangan pertama sebanyak O(E), selanjutnya O(E) diulanga sebanyak O(V) jadi totalnya adalah O(V.E)

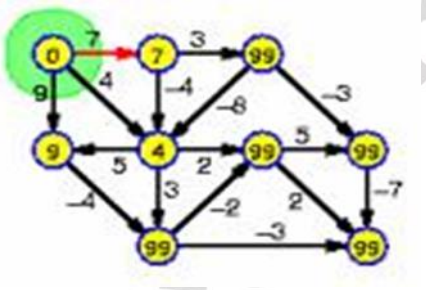
3. Tahap ketiga yaitu pengecekan ada atau tidaknya sisi negative pada jalur, mempunyai kompleksitas O(E).

Dari semua kompleksitas di atas dapat ditentukan kompleksitas waktu algoritma ini adalah O(V.E). Skema Pencarian Lintasan Terpendek dengan algoritma Bellman Ford.



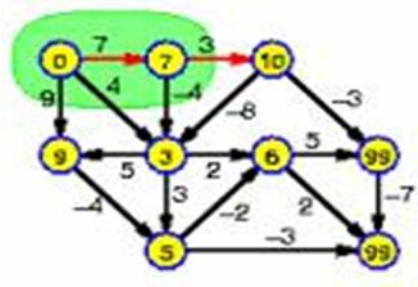
Gambar 2.11 Graf berbobot negative.

Dari **gambar 2.11** sudah di tentukan dari node awal 0 ke node akhir yang nanti di temukan nilai node.



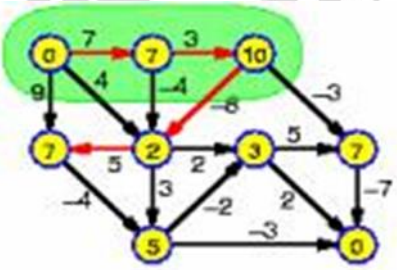
Gambar 2.12 Tahap pertama Algoritma

Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.11. pada node awal diketaui nilai node 0 . Jika node 0 ke node 7 maka 0+7= 7.



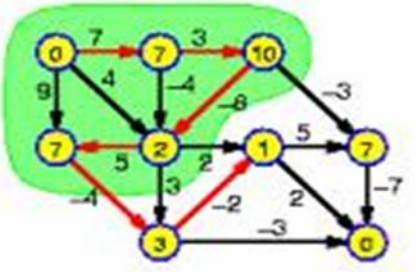
Gambar 2.13 Tahap Kedua Algoritma

Gambar 2.13 Tahap kedua Algoritma Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.11. Selanjutnya nilai node 7 ke node 3 maka 7+3= 10.



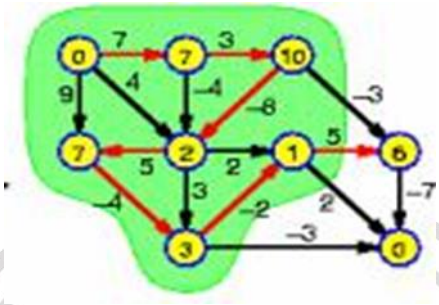
Gambar 2.14 Tahap Ketiga Algoritma

Gambar 2.14 Tahap ketiga Algoritma Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.11. Selanjutnya nilai node 10 ke node (-8) , maka 10+(-8)= 2.



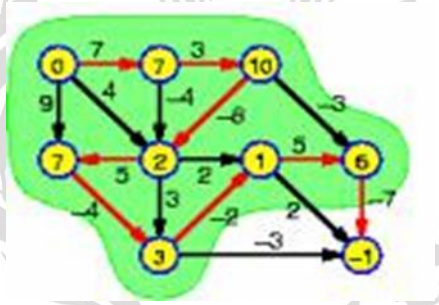
Gambar 2.15 Tahap Keempat Algoritma

Gambar 2.15 Tahap keempat Algoritma Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.11. Selanjutnya nilai node 2 ke node 5, maka 2+5= 7.



Gambar 2.16 Tahap Kelima Algoritma

Gambar 2.16 Tahap Kelima Algoritma Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.11. Selanjutnya nilai node 7 ke node (-4), maka 7+(-4)= 3.



Gambar 2.17 Tahap Keenam Algoritma

Gambar 2.17 Lintasan terpendek untuk penyelesaian contoh graf pada gambar 2.11. Selanjutnya node 3 ke node (-2), maka 3+(-2)= 1, setelah itu node 1 ke node 5, maka 1+5= 6, dan yang terakhir node 6 ke node (-7), maka 6+(-7)= -1.

Maka dapat diketahui penjumlahan dari node awal ke node akhir adalah:

Node 0+7= 7

Node 7+3= 10

Node 10+(-8)= 2

Node 2+5= 7

Node 7+(-4)= 3

Node 3+1= 1

Node 1+5= 6

Node 6+(-7)= -1.

Jadi hasil akhir dari node 0 ke node akhir adalah sebesar -1.

2.5 Sistem Pendukung Keputusan (SPK)

Sistem Pendukung Keputusan (SPK) atau Decision Support System (DSS) adalah sebuah sistem yang mampu memberikan kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah dengan kondisi semi terstruktur dan tak terstruktur. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi semi terstruktur dan situasi yang tidak terstruktur, dimana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat (Turban, 2001).

SPK bertujuan untuk menyediakan informasi, membimbing, memberikan prediksi serta mengarahkan kepada pengguna informasi agar dapat melakukan pengambilan keputusan dengan lebih baik. Implementasi teori-teori pengambilan keputusan yang telah di perkenalkan oleh ilmu-ilmu seperti operation research dan menegement science, hanya bedanya adalah bahwa jika dahulu untuk mencari penyelesaian masalah yang dihadapi harus dilakukan perhitungan iterasi secara manual (biasanya untuk mencari nilai minimum, maksimum, atau optimum), saat ini computer PC telah menawarkan kemampuannya untuk menyelesaikan persoalan yang sama dalam waktu singkat. Sprague dan Watson mendefinisikan Sistem Pendukung Keputusan (SPK) sebagai sistem yang memiliki lima karakteristik yaitu:

1. Sistem yang berbasis komputer.
2. Dipergunakan untuk membantu para pengambil keputusan
3. Untuk memecahkan masalah-masalah rumit yang mustahil dilakukan.
4. Melalui cara simulasi yang interaktif

5. Dimana data dan model analisis sebagai komponen utama.

2.6 Penelitian Sebelumnya

Rasdiana 2015, ”Aplikasi Algoritma Bellman-Ford Dalam Meminimumkan Rute Perjalanan Tukang Bentor Di Kecamatan Biringkanaya”. Didalam penelitian tersebut melakukan penelitian untuk menentukan lintasan minimum yang ditempuh Tukang Bentor yang ada di kecamatan Biringkanaya dengan menggunakan *Algoritma Bellman- Ford.* Pembahasan skripsi ini dibatasi pada aplikasi algoritma Bellman – Ford dan hanya 15 jalan yang diteliti di kecamatan Biringkanaya. Dengan tujuan yang ingin dicapai yaitu untuk mengetahui lintasan minimum yang ditempuh Tukang Bentor yang ada di kecamatan Biringkanaya dengan menggunakan *Algoritma Bellman- Ford.*

Nurul Fadhlia 2015, “Rekomendasi Rute Spbu Terdekat Menggunakan Algoritma Bellman-Ford Berbasis Android”. Didalam penelitian tersebut melakukan penelitian untuk mencari lokasi SPBU yang terdekat dengan jalan tujuan pengguna atau yang terdekat dengan pengguna berada untuk mengisi bahan bakar. Oleh karena itu, diperlukan sebuah aplikasi untuk mendapatkan informasi rekomendasi rute SPBU terdekat menuju lokasi tujuan. Algoritma Bellman-Ford memiliki kinerja yang lebih baik dan efektif dari djkstra karena algoritma *bellman-ford* melakukan proses pengulangan dan perbaikan lintasan untuk setiap arc yang terhubung sehingga dapat menberikan jalur SPBU mana saja yang terdekat untuk menuju lokasi tujuan.

Wahyu Andre Anto, Muslim Alamsyah, Mohammad Zoqi Sarwani, “Pencarian Jalur Tercepat Pariwisata Di Kabupaten Pasuruan Dengan Menggunakan Metode Bellman Ford”. Didalam penelitian tersebut melakukan penelitian terkait dalam kegiatan perjalanan pariwisata sehari-hari, beberapa agen perjalanan wisata membutuhkan pemilihan rute terdekat untuk melintas antar obyek pariwisata dalam Kabupaten Pasuruan sehingga didapatkan efisiensi waktu transit antar obyek dan biaya. Dengan

perbandingan jumlah jalan dan kendaraan yang tidak seimbang maka diperlukan pengetahuan bagi pengendara agen perjalanan untuk memilih jalur alternatif agar mendapatkan jalur tercepat sewaktu melakukan perjalanan menuju wisata di Kabupaten Pasuruan. Algoritma Bellman Ford ini akan memudahkan user dalam mencari jalur tercepat. Pengujian yang akan dilakukan meliputi pengujian dari aspek jarak dan waktu yang dapat ditempuh. Hasil dari sistem pencarian jalur tercepat yang akan dibangun adalah peta pariwisata dengan titik awal menuju titik tujuan dengan jalur tercepat,jalur alternatif, total jarak dan waktu yang akan di tempuh.

UNIVERSITAS MUHAMMADIYAH GRESIK