

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Umum**

Di Indonesia khususnya Surabaya menggunakan acuan Manual Kapasitas Jalan Indonesia. Manual Kapasitas Jalan Indonesia adalah suatu sistem yang disusun sebagai suatu metode efektif yang berfungsi untuk perancangan manajemen lalu lintas yang direncanakan terutama agar pengguna dapat memperkirakan perilaku lalu lintas dari suatu fasilitas pada kondisi lalu lintas geometrik dan keadaan lingkungan tertentu, sehingga diharapkan dapat membantu untuk mengatasi permasalahan kondisi lalu lintas di jalan perkotaan.

Fasilitas – fasilitas jalan yang terdapat pada Manual Kapasitas Jalan Indonesia (MKJI) antara lain Simpang Bersinyal, Simpang Tak Bersinyal, Bagian Jalinan, Jalan Perkotaan, Jalan Luar Kota, dan Jalan Bebas Hambatan. Manual Kapasitas Jalan Indonesia ini dapat digunakan untuk menganalisa rute, segmen, jaringan jalan, termasuk simpang pada suatu kawasan perkotaan yaitu dengan penerapan metode penyelesaian yang sesuai tipe fasilitas lalu lintas. Seperti pada persimpangan dengan sinyal diselesaikan dengan metode simpang bersinyal, persimpangan tanpa sinyal dengan metode simpang tak bersinyal.

## 2.2 Simpang

Simpang adalah suatu area yang kritis pada suatu jalan raya yang merupakan tempat titik konflik dan tempat kemacetan karena bertemunya dua ruas jalan atau lebih. Simpang yang dievaluasi adalah simpang bersinyal dimana simpang bersinyal merupakan suatu fasilitas jalan yang berlengan 3 atau 4 dengan bentuk geometrik serta menggunakan peralatan sinyal pengatur lalu lintas. Persimpangan Jalan Ahmad Yani Giant merupakan jalan arteri primer karena persimpangan ini menghubungkan kota Surabaya dengan kota-kota lain, seperti kota Sidoarjo, kecepatan kendaraan rata-rata tinggi, pergerakan lalu lintas menerus.

### 2.2.1 Simpang Bersinyal

Sebelum dikenal adanya lampu lalu lintas, maka pada simpang-simpang yang sangat sibuk, ditugaskan polisi lalu lintas untuk mengatur arus lalu lintas. Kemudian muncul ide pertama kali pada tahun 1918 di New York untuk menggunakan lampu pengatur lalu lintas dengan warna merah, kuning, hijau.

Persoalan kemudian muncul, karena lampu lalu lintas ternyata tidak dapat mengatur lalu lintas sebaik seorang polisi lalu lintas. Pada pengaturan dengan lampu lalu lintas, maka periode lampu lalu lintas tetap, tidak dapat menyesuaikan dengan keadaan yang ada (kecuali kalau ada polisi yang berjaga di dekatnya). Oleh karena itu, dikembangkan lampu lalu lintas yang dapat menyesuaikan dengan arus lalu lintas yang lewat.

Persoalan berikutnya muncul, yaitu kalau lampu lalu lintas dipasang berjajar pada suatu jalan yang sangat panjang. Para pengemudi mengeluh karena harus berhenti berkali-kali pada setiap lampu lalu lintas. Masalah ini kemudian diatasi dengan koordinasi antar lampu lalu lintas.

#### 2.2.1.1 Kapan dipasang lampu lalu lintas?

Pada saat arus lalu lintas sudah mulai meninggi, maka lampu lalu lintas sudah harus dipasang. Ukuran meningginya arus lalu lintas yaitu dari waktu tunggu rata-rata kendaraan pada saat melintasi simpang. Jika waktu tunggu rata-rata tanpa lampu lalu lintas sudah lebih besar dari waktu tunggu rata-rata dengan lampu lalu lintas, maka perlu dipasang lampu lalu lintas.

Waktu tunggu pada simpang tanpa lampu lalu lintas dipengaruhi oleh :

- Arus lalu lintas pada masing-masing arah,
- Waktu antara kedatangan kendaraan dari masing-masing arah,
- Keberanian pengemudi untuk menerima waktu antara yang tersedia guna menyeberangi jalan.

#### 2.2.1.2 Penentuan Fase

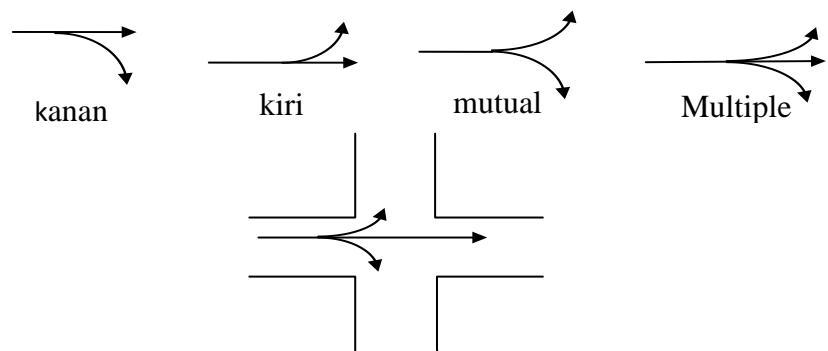
Pada perencanaan lalu lintas, dikenal beberapa istilah :

1. Waktu siklus (*cycle time*) : waktu satu periode lampu lalu lintas, misalnya pada saat suatu arus diruas jalan A mulai hijau, hingga pada ruas jalan tersebut mulai hijau lagi.
2. Fase : suatu rangkaian dari kondisi yang diberlakukan untuk suatu arus atau beberapa arus, yang mendapatkan identifikasi lampu lalu lintas yang sama.
3. Bentuk Alih Gerak (*manuver*)

Dari sifat dan tujuan gerakan di daerah persimpangan, dikenal beberapa bentuk alih gerak (*manuver*) antara lain, *diverging* (memisah), *merging* (menggabung), *crossing* (memotong), dan *weaving* (menyilang).

1) *Diverging* (memisah)

*Diverging* adalah peristiwa memisahkannya kendaraan dari suatu arus yang sama ke jalur yang lain:

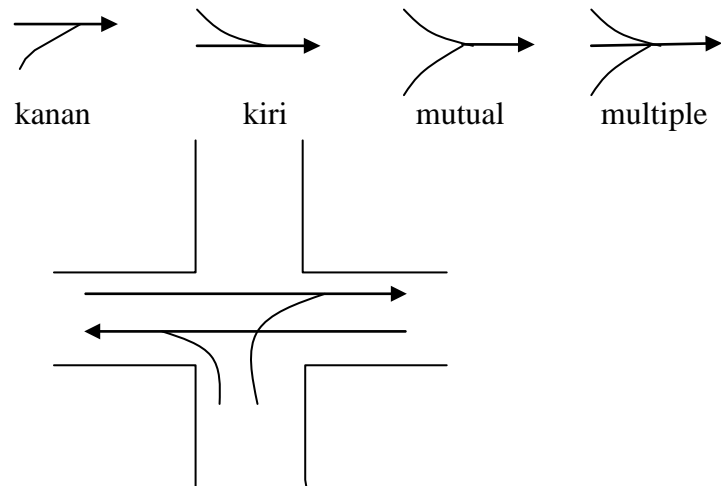


Gambar 2.1 Arus *Diverging* (memisah)

Sumber Munawar : 45 – 47

## 2) *Merging* (menggabung)

*Merging* adalah peristiwa menggabungnya kendaraan dari suatu jalur ke jalur yang lain:

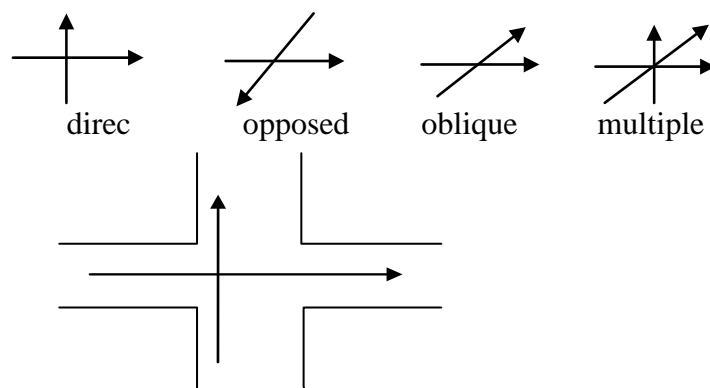


Gambar 2.2 Arus *Merging* (menggabung)

Sumber Munawar : 45 – 47

## 3) *Crossing* (memotong)

*Crossing* adalah peristiwa perpotongan antara arus kendaraan dari satu jalur ke jalur yang lain pada persimpangan di mana keadaan yang demikian akan menimbulkan titik konflik pada persimpangan tersebut.

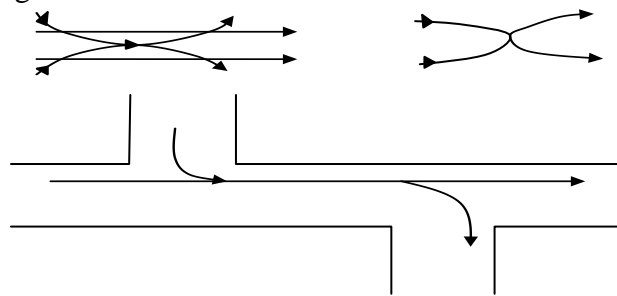


Gambar 2.3 Arus *Crossing* (memotong)

Sumber Munawar : 45 – 47

#### 4) *Weaving* (menyilang)

*Weaving* adalah pertemuan dua arus lalu lintas atau lebih yang berjalan menurut arah yang sama sepanjang suatu lintasan di jalan raya tanpa bantuan rambu lalu lintas. Gerakan ini sering terjadi pada suatu kendaraan yang berpindah dari suatu jalur ke jalur lain misalnya pada saat kendaraan masuk ke suatu jalan raya dari jalan masuk, kemudian bergerak ke jalur lainnya untuk mengambil jalan keluar dari jalan raya tersebut keadaan ini juga akan menimbulkan titik konflik pada persimpangan tersebut.



Gambar 2.4 Arus *weaving* (menyilang)

Sumber Munawar : 45 – 47

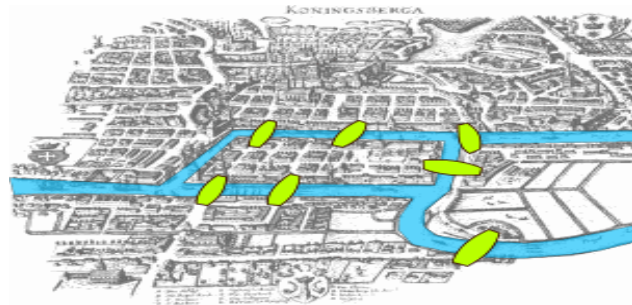
### 2.3 Frontage

Frontage road adalah jalan pendamping jalan utama yang menyediakan akses ke jalan masuk pribadi seperti pusat perbelanjaan, industri, kantor, dll. Frontage road memberikan alternatif rute untuk mengurangi kemacetan di jalan utama sehingga dialihkan ke frontage.

## 2.4 Teori Graf

### 2.4.1 Pengenalan Teori Graf

Teori graf adalah cabang ilmu yang mempelajari sifat-sifat graf, yang pertama kali diperkenalkan pada tahun 1736. Baru pada sekitar tahun 1920 teori graf berkembang pesat terutama salah satunya adalah aplikasinya yang sangat luas dalam kehidupan sehari-hari maupun dalam berbagai bidang ilmu seperti: Ilmu Komputer, Teknik, Sains, bahkan Ilmu Sosial. (Ketut, 2007 : 1).

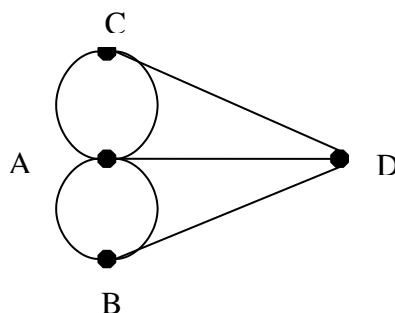


Gambar 2.5 Jembatan utama di *Königsberg*  
Sumber Munir : 355

Menurut catatan sejarah, masalah jembatan *Königsberg* adalah masalah yang pertama kali menggunakan graf tahun 1739 (Munir, 2005 : 354). Di kota *Königsberg* (sebelah timur negara bagian *Prussia*, Jerman), sekarang bernama kota *Kaliningrad*, terdapat sungai *Pregal* yang mengalir mengitari pulau *Kneiphof* lalu bercabang menjadi dua buah anak sungai yang diperlihatkan oleh gambar 2.5. Permasalahannya ialah menemukan perjalanan atau rute dari suatu kota melalui ketujuh buah jembatan masing-masing tepat satu kali, kemudian kembali lagi ke tempat awal. Pulau tersebut tidak dapat dicapai oleh rute apapun selain melalui jembatan-jembatan tersebut.

Tahun 1736, *Leonhard Euler* adalah orang pertama yang berhasil menemukan jawaban masalah tersebut dengan pembuktian yang sederhana (melalui karya tulisannya *Seven Bridges of Königsberg*). Daratan (titik-titik yang dihubungkan oleh jembatan) dinyatakan sebagai titik disebut verteks dan jembatan dinyatakan sebagai *edge*. Dari analisa *Euler* pada jembatan *Königsberg* menghasilkan sebuah model graf, seperti yang diperlihatkan pada gambar 2.6.

Analisis *Euler* mengenai permasalahan jembatan di *Königsberg* tidak menghasilkan solusi. Karena orang tidak mungkin melalui ketujuh jembatan masing-masing tepat satu kali dan kembali lagi ke tempat awal keberangkatan jika derajat (banyaknya garis yang bersisian dengan titik) setiap verteks tidak seluruhnya genap. Penemuan *Euler* adalah kunci yang menandai perkembangan topologi, di mana perbedaan antara *layout* sebenarnya dan graf *schematic* adalah contoh yang bagus untuk gagasan bahwa topologi tidak dibatasi dengan bentuk kaku dari objek-objek tertentu.



Gambar 2.6 Graf yang merepresentasikan jembatan *Königsberg*  
Sumber Munir : 355



## 2.5 Definisi Graf

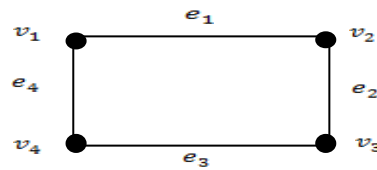
Graf  $G$  didefinisikan sebagai pasangan himpunan  $(V, E)$ , ditulis dengan notasi  $G = (V, E)$  (Munir, 2005 : 356). Sebuah graf  $G$  berisikan dua himpunan yaitu himpunan berhingga tak kosong  $V(G)$  dari objek-objek yang disebut titik dan himpunan berhingga (mungkin kosong)  $E(G)$  yang elemen-elemennya disebut sisi sedemikian hingga setiap elemen  $e$  dalam  $E(G)$  merupakan pasangan tak berurutan dari titik-titik di  $V(G)$  disebut himpunan titik  $G$  dan  $E(G)$  disebut himpunan sisi  $G$  (Ketut, 2007 : 2).

Menurut Munir (2005 : 356), graf  $G$  didefinisikan sebagai pasangan himpunan  $(V, E)$ , ditulis dengan notasi  $G = (V, E)$ , yang dalam hal ini  $V$  adalah himpunan tidak kosong dari titik-titik (*vertices* atau *node*) dan  $E$  adalah himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang titik,  $E$  boleh kosong. Jadi, sebuah graf dimungkinkan tidak mempunyai sisi, tetapi titiknya harus ada minimal satu. Graf yang hanya mempunyai satu buah titik tanpa sisi dinamakan graf trivial.

Misalkan  $G$  adalah sebuah graf. Sebuah jalan  $W$  disebut tertutup jika titik awal dan titik akhir dari  $W$  sama. Jejak tertutup disebut sirkuit. Sirkuit dengan titik awal dan titik akhir internalnya berlainan disebut siklus (*cycle*).

Contoh:

Sebuah graf  $G = (V, E)$  dengan  $V(G) = \{v_1, v_2, v_3, v_4\}$  dan  $E(G) = \{e_1, e_2, e_3, e_4\}$  dimana  $e_1 = (v_1, v_2)$ ,  $e_2 = (v_2, v_3)$ ,  $e_3 = (v_3, v_4)$ ,  $e_4 = (v_4, v_1)$  dapat dipresentasikan dalam bentuk Gambar 2.7 .

Gambar 2.7 Contoh Graf  $G$ 

Sumber Ketut : 2

$$G = \{V, E\}$$

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{e_1, e_2, e_3, e_4\}$$

$$\text{Jalan (walk) : } \{v_1, e_1, v_2, e_2, v_3, e_3, v_4, e_4, v_1\}$$

$$\text{Siklus (cycle) : } \{v_1, e_1, v_2, e_2, v_3, e_3, v_4, e_4, v_1\}$$

$$\text{Jejak tertutup (sirkuit) : } \{v_1, v_2, v_3, v_4, v_1\}$$

$$\text{Lintasan (path) : } \{v_1, v_2, v_3, v_4\}$$

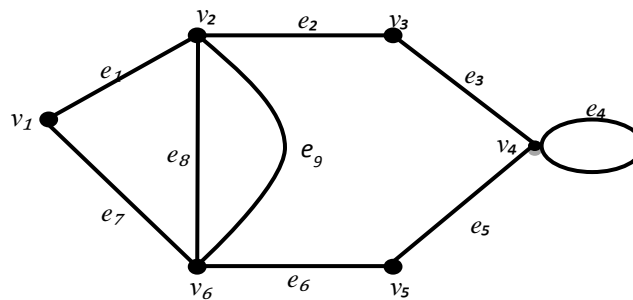
$$\text{Jejak (trail) : } \{v_4, v_1, v_2, v_3\}$$

Sebuah sisi yang hanya menghubungkan sebuah titik dengan dirinya sendiri disebut gelung (*loop*). Jika terdapat lebih dari satu sisi yang menghubungkan dua titik  $u$  dan  $v$  pada suatu graf, maka sisi-sisi tersebut disebut sisi ganda (Ketut, 2007 : 3).

Misalkan  $G$  adalah sebuah graf. Sebuah jalan (*walk*) di  $G$  adalah sebuah barisan berhingga tak kosong  $W = (v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k)$  yang suku-sukunya bergantian titik dan sisi, sedemikian hingga  $v_{i-1}$  dan  $v_i$  adalah titik-titik akhir sisi  $e_i$ , untuk  $1 \leq i \leq k$ . Dapat dinyatakan  $W$  adalah sebuah jalan dari titik  $v_0$  ke titik  $v_k$  atau jalan  $(v_0, v_k)$ . Titik  $v_0$  dan titik  $v_k$  berturut-turut disebut titik awal dan titik akhir  $W$ . Jika semua sisi  $e_1, e_2, e_3, \dots, e_k$  dalam jalan  $W$  yang berbeda, maka  $W$  disebut jejak (*trail*). Jika semua titik  $v_0, v_1, v_2,$

...,  $v_k$  dalam jalan  $W$  juga berbeda, maka  $W$  disebut lintasan (*path*). (Ketut, 2007 : 6).

Contoh:



Gambar 2.8 Graf G

Sumber Munir : 356

$$G = \{V, E\}$$

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$$

$$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$$

$$\text{Jalan (walk) : } \{v_1, e_1, v_2, e_2, v_3, e_3, v_4, e_5, v_5, e_6, v_6, e_7, v_1\}$$

$$\text{Siklus (cycle) : } \{v_1, v_2, v_6, v_1\}$$

$$\text{Jejak tertutup (sirkuit) : } \{v_1, e_1, v_2, e_8, v_6, e_7, v_1\}$$

$$\text{Lintasan (path) : } \{v_1, v_2, v_3, v_4\}$$

$$\text{Jejak (trail) : } \{v_6, v_1, v_2, v_6, v_5\}$$

Pada Gambar 2.8, sisi  $e_4$  adalah *loop*. Sisi  $e_8$  dan  $e_9$  merupakan sisi ganda karena kedua sisi tersebut terkait dengan titik  $v_2$  dan  $v_6$ . Sehingga dapat disimpulkan bahwa graf adalah sekumpulan noktah (simpul) yang dihubungkan dengan sekumpulan garis (sisi).

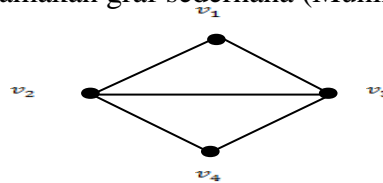
## 2.6 Jenis Graf

Graf dapat dikelompokkan menjadi beberapa kategori (jenis) bergantung pada sudut pandang pengelompokannya. Pengelompokan graf dapat dipandang berdasarkan ada tidaknya sisi ganda, berdasarkan jumlah titik, atau berdasarkan orientasi arah pada sisi (Munir, 2005 : 357).

**2.6.1 Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf maka secara umum graf dapat digolongkan menjadi dua jenis, yaitu:**

a) Graf Sederhana (*Simple Graph*)

Graf yang tidak mengandung gelang (*loop*) maupun sisi ganda dinamakan graf sederhana (Munir, 2005 : 357).



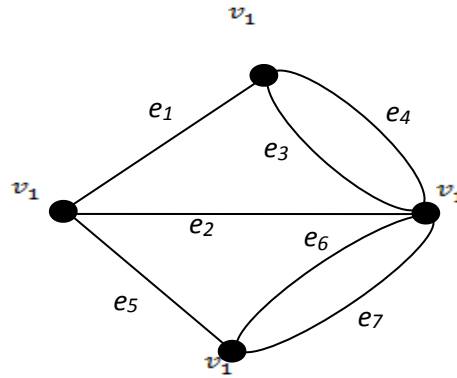
Gambar 2.9 Graf Sederhana  
Sumber Munir : 356

b) Graf Tak Sederhana (*Unsimple Graph*)

Menurut Munir (2005 : 357) graf yang mengandung sisi ganda atau *loop* dinamakan graf tak sederhana (*unsimple graph*).

Ada dua macam graf tak sederhana, yaitu graf ganda (*multigraph*) dan graf semu (*pseudograph*).

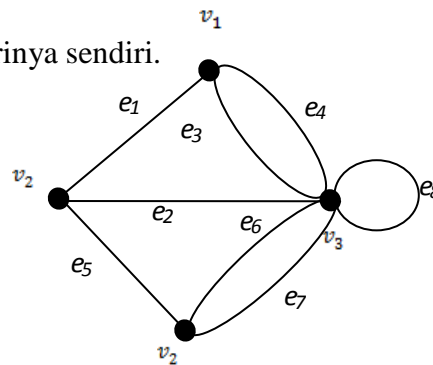
- c) Graf Ganda (*Multigraph*) adalah graf yang mengandung sisi ganda tetapi tidak memiliki *loop* (gelang). Sisi ganda yang menghubungkan sepanjang titik dapat lebih dari dua buah.



Gambar 2.10 Graf Ganda

Sumber Munir : 356

- d) Graf Semu (*Pseudograph*) adalah graf yang mengandung loop (termasuk bila memiliki sisi ganda sekalipun). Graf semu lebih umum dari pada graf ganda, karena sisi pada graf semu dapat terhubung ke dirinya sendiri.

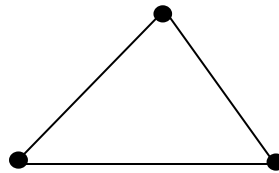


Gambar 2.11 Graf Semu

Sumber Munir : 356

### 2.6.2 Berdasarkan jumlah titik pada suatu graf, yaitu: Graf Berhingga.

Graf berhingga adalah sebuah graf  $G (V, E)$  dengan  $V$  (himpunan titik) dan  $E$  (himpunan sisi) hingga.



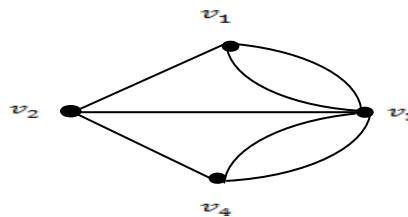
Gambar 2.12 Graf Berhingga

Sumber Munir : 375

### 2.6.3 Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas dua jenis, yaitu:

#### a) Graf Tak Berarah (*Undirected Graph*)

Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak berarah. Pada graf tak berarah, urutan pasangan titik yang dihubungkan oleh sisi tidak diperhatikan. Jadi,  $(v_j, v_k) = (v_k, v_j)$  adalah sisi yang sama (Munir, 2005 : 358). Pada Gambar 2.10 adalah contoh graf tak berarah.



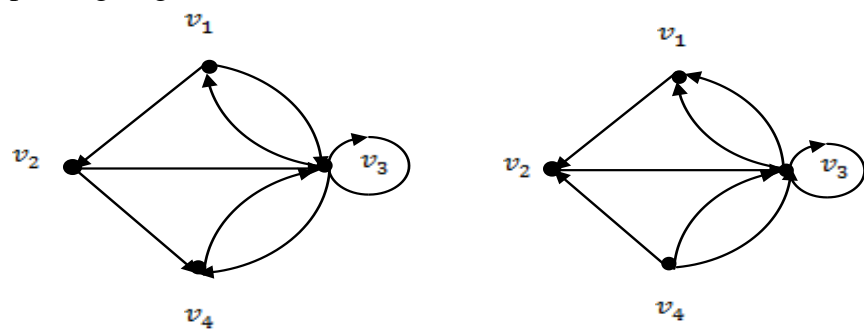
Gambar 2.13 Graf Tak Berarah

Sumber Munir : 356

#### b) Graf Berarah (*Directed Graph* atau *Digraph*)

Graf yang setiap sisinya diberikan orientasi arah disebut

sebagai graf berarah. Sisi berarah disebut juga busur. Pada graf berarah,  $(v_j, v_k)$  dan  $(v_k, v_j)$  menyatakan dua buah busur yang berbeda, dengan kata lain  $(v_j, v_k) \neq (v_k, v_j)$ . Untuk busur  $(v_j, v_k)$ , titik  $v_j$  dinamakan titik asal (*initial vertex*), dan titik  $v_k$  dinamakan titik terminal (*terminal vertex*). Pada graf berarah, *loop* diperbolehkan, tetapi sisi ganda tidak diperbolehkan sedangkan graf ganda berarah *loop* dan sisi ganda diperbolehkan (Munir, 2005 : 358). Pada Gambar 2.14(a) merupakan graf berarah sedangkan Gambar 2.14(b) merupakan graf ganda berarah.



Gambar 2.14 (a) Graf Berarah dan

(b) Graf Ganda Berarah

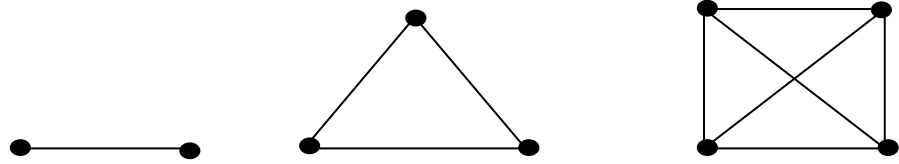
Sumber Munir : 359

#### 2.6.4 Berdasarkan beberapa graf sederhana khusus, maka secara umum graf dibedakan atas dua jenis, yaitu:

##### a) Graf Lengkap (*Complete Graph*)

Graf lengkap ialah graf sederhana yang setiap titiknya mempunyai sisi ke semua titik lainnya. Graf lengkap dengan  $n$  buah titik dilambangkan  $K_n$ .

setiap titik pada  $K_n$  berderajat  $n - 1$  (Munir, 2005 : 377).

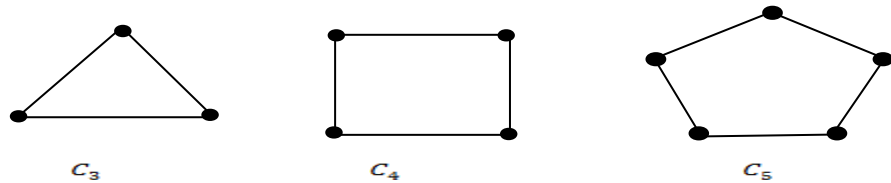


Gambar 2.15 Graf Lengkap

Sumber Munir : 377

b) Graf Lingkaran

Graf lingkaran adalah graf sederhana yang setiap titiknya berderajat dua. Graf lingkaran dengan  $n$  titik dilambangkan dengan  $C_n$  (Munir, 2005 : 377).

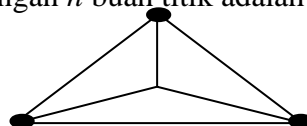


Gambar 2.16 Graf Lingkaran

Sumber Munir : 378

c) Graf Teratur/Graf Reguler

Graf yang setiap titiknya mempunyai derajat yang sama disebut graf teratur. Apabila derajat setiap titik adalah  $r$ , maka graf tersebut disebut sebagai graf teratur derajat  $r$ . Jumlah sisi pada graf teratur derajat  $r$  dengan  $n$  buah titik adalah  $\frac{nr}{2}$  (Munir, 2005 : 378).



Gambar 2.17 Graf Teratur dengan 4 titik dan derajat 3

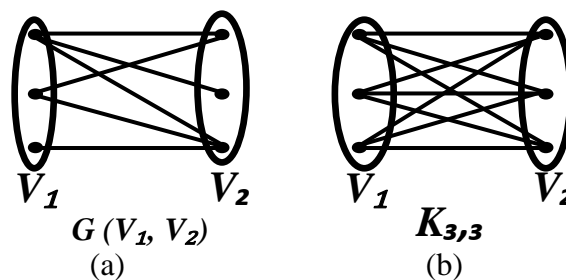
Sumber Munir : 379



d) Graf Bipartit (*Bipartite Graph*)

Graf bipartite adalah graf  $G$  himpunan titiknya dapat dikelompokkan menjadi dua himpunan bagian  $V_1$  dan  $V_2$ , sedemikian sehingga setiap sisi di dalam  $G$  menghubungkan sebuah titik di  $V_1$  ke sebuah titik di  $V_2$  dan dinyatakan sebagai  $G(V_1, V_2)$ . Dengan kata lain, setiap pasang titik di  $C$  (demikian pula dengan titik-titik di  $V_2$ ) tidak bertetangga (Munir, 2005 : 379).

Apabila setiap titik di  $V_1$  bertetangga dengan semua titik di  $V_2$ , maka  $G(V_1, V_2)$  disebut graf bipartit lengkap, dilambangkan dengan  $K_{m,n}$ . Jumlah sisi pada graf bipartit lengkap adalah  $mn$  (Munir, 2005 : 380).



Gambar 2.18 Graf Bipartit

Sumber Munir : 381

Pada Gambar 2.18 (a)  $G(V_1, V_2)$  adalah graf bipartit, sedangkan Gambar 2.18 (b)  $K_{3,3}$  adalah graf bipartit lengkap.

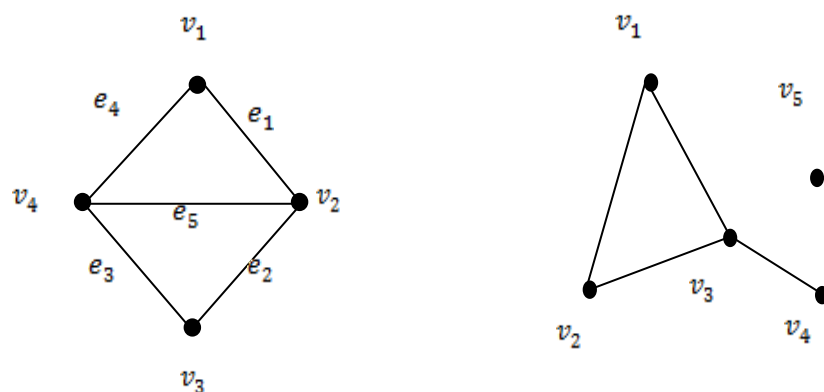
## 2.7 Terminologi Dasar Graf

Dalam pembahasan mengenai graf biasanya sering menggunakan terminologi (istilah) yang berkaitan dengan graf. Berikut ini terminologi

(istilah) yang berkaitan dengan graf yang akan digunakan dalam skripsi ini, yang dirujuk dari Munir (2005).

a) Bertetangga (*Adjacent*)

Dua buah titik pada graf tak berarah  $G$  dikatakan bertetangga bila keduanya terhubung langsung dengan sebuah sisi. Dengan kata lain  $v_j$  bertetangga dengan  $v_k$  jika  $(v_j, v_k)$  adalah sebuah sisi pada graf  $G$  (Munir, 2005 : 365). Pada Gambar 2.19(a), titik  $v_1$  bertetangga dengan titik  $v_2$  dan  $v_4$ , dan titik  $v_1$  tidak bertetangga dengan titik  $v_3$ . Sedangkan pada gambar 2.19(b), titik  $v_1$  bertetangga dengan titik  $v_2$  dan  $v_3$ , titik  $v_1$  tidak bertetangga dengan titik  $v_3$ .



Gambar 2.19 Graf  $G_1$  dan Graf  $G_2$

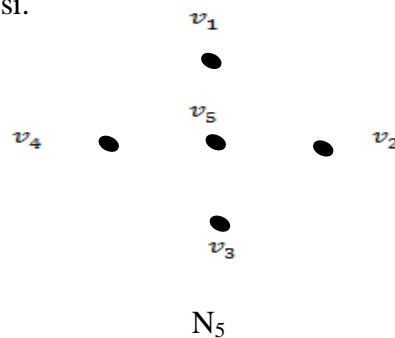
Sumber Munir : 382

b) Bersisian (*Incident*)

Untuk sembarang sisi  $e = (v_j, v_k)$ , sisi  $e$  dikatakan bersisian dengan titik  $v_j$  dan titik  $v_k$  (Munir, 2005 ; 365). Pada Gambar 2.19(a), sisi  $e_1$  bersisian dengan titik  $v_1$  dan titik  $v_2$ , tetapi sisi  $e_1$  tidak bersisian dengan titik  $v_4$ .

c) Graf Kosong (*Null Graph*)

Graf yang himpunan sisinya merupakan himpunan kosong disebut Graf Kosong (*Null Graph*) dan ditulis sebagai  $N_n$ ,  $n$  adalah jumlah titik (Munir, 2005). Graf kosong (graf nol) adalah graf yang tidak memiliki sisi.



Gambar 2.20 Graf Kosong (*Null Graph*)

Sumber Munir : 366

d) Derajat (*Degree*)

Derajat suatu titik pada graf tak berarah adalah jumlah sisi yang bersisian dengan titik tersebut (Munir, 2005 : 366). Pada Gambar 2.19 (a), graf  $G_1$ :  $d(v_1) = d(v_3) = 2$ ,  $d(v_2) = d(v_4) = 3$ .

## 2.8 Pewarnaan pada Graf

### 2.8.1 Algoritma pewarnaan graf (*Welch – Powell*) :

Inputnya dalah suatu graf  $G$

Langkah 1 : Urutkan verteks  $G$  menurut derajat yang mengecil.

Langkah 2 : Biarkan warna pertama  $C_1$  pada verteks pertama dan lalu, secara berurutan, berikan  $C_1$  ke setiap verteks yang tidak

bersebelahan dengan verteks sebelumnya yang telah diberi  $C_1$ .

Langkah 3 : Ulangi langkah 2 dengan warna kedua  $C_2$  dan verteks berikutnya yang belum diwarnai.

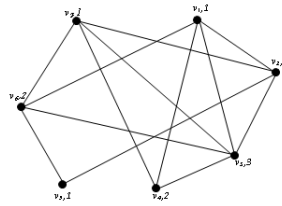
Langkah 4: Ulangi Langkah 3 dengan warna ketiga  $C_3$ , lalu warna keempat  $C_4$ , dan demikian seterusnya sampai semua verteks telah diwarnai.

Langkah 5 : Keluar

Pewarnaan titik maupun pewarnaan sisi pada graf merupakan salah satu topik dalam teori graf yang kaya dengan aplikasi. Konsep pewarnaan graf dibagi menjadi 2. Yang pertama konsep pewarnaan titik pada graf, kemudian dilanjutkan dengan permasalahan menentukan minimum banyak warna yang diperlukan untuk mewarnai titik graf yang selanjutnya disebut bilangan khromatik graf, dan akhirnya disajikan aplikasi pewarnaan titik pada graf. Yang kedua adalah tentang pewarnaan sisi graf, indeks khromatik, beserta aplikasinya.

2.8.2 Misal  $G$  sebuah graf. Sebuah pewarnaan- $k$  dari  $G$  adalah pewarnaan semua titik  $G$  dengan menggunakan  $k$  warna sedemikian hingga dua titik  $G$  yang berhubungan langsung mendapat warna yang berbeda. Jika  $G$  memiliki sebuah pewarnaan- $k$  maka  $G$  dikatakan dapat diwarnai sebuah  $k$  warna. Sebuah pewarnaan- $k$  dari graf  $G$  biasanya ditunjukkan dengan melabel titik-titik  $G$  dengan warna  $1,2,3,\dots,k$ .

Misalnya, sebuah pewarnaan-5 graf  $G$  diperlihatkan pada gambar berikut.



Gambar 2.21 Sebuah pewarnaan-5 dari  $G$

Sumber Ketut :152

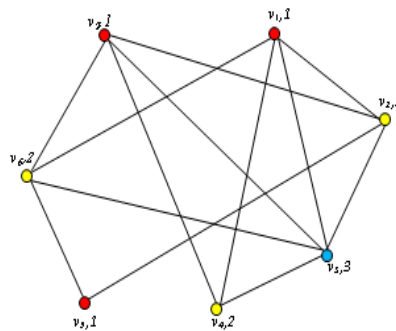
Karena titik  $v_1$  dan titik  $v_2$  berhubungan langsung pada graf  $G$ , maka titik  $v_1$  dan titik  $v_2$  tidak boleh mendapat warna yang sama. Dalam hal ini, titik  $v_1$  diwarnai dengan warna 1 dan titik  $v_2$  diwarnai dengan warna 2. Sedangkan, titik  $v_1$  dan  $v_5$  tidak berhubungan langsung, sehingga  $v_5$  dapat diwarnai dengan warna 1. Karena titik  $v_3$  harus berbeda dengan warna  $v_1$  dan  $v_2$ , jadi  $v_3$  diberi warna yang berbeda dengan warna 1 dan warna 2, misalnya diberi warna 3, dan seterusnya.

Perhatikan bahwa, graf  $G$  mempunyai gelung (loop), misalnya pada titik  $v_1$  maka  $v$  berhubungan langsung dengan dirinya sendiri, sehingga tidak ada pewarnaan titik yang memungkinkan untuk graf  $G$ . jika dua titik berbeda di graf  $G$  dihubungkan oleh satu sisi atau lebih dari satu sisi, maka kedua titik tersebut tetap harus mendapat warna yang berbeda. Sehingga, berkaitan dengan pewarnaan titik pada graf, cukup dibatasi pada graf-graf yang berbeda saja.

#### 1) Bilangan Khromatik pada Graf

Misalkan  $G$  sebuah graf. Bilangan khromatik (Chromatic Number) dari graf  $G$ , dilambangkan dengan  $\chi(G)$ , didefinisikan sebagai berikut :  $\chi(G) = \min \{k / \text{ada pewarnaan-}k \text{ pada } G\}$ .

Dengan kata lain, bilangan khromatik graf  $G$  adalah minimum banyaknya warna yang diperlukan untuk mewarnai sebuah titik  $G$ , sedemikian hingga setiap dua titik yang berhubungan langsung mendapat warna yang berbeda. Jika  $\chi(G) = k$  maka ada sebuah pewarnaan- $k$  pada graf  $G$ , tetapi sebaliknya tidak berlaku. Misalnya seperti diperlihatkan pada gambar 2.1, terdapat sebuah pewarnaan-5 pada graf  $G$ , tetapi  $\chi(G) \neq 5$ . Karena graf  $G$  dapat diwarnai dengan kurang dari 5 warna, misalnya  $G$  dapat diwarnai dengan 3 warna, seperti terlihat pada gambar 2.22, dan arena graf  $G$  tidak dapat diwarnai dengan menggunakan kurang dari 3 warna, maka bilangan khromatik  $G$  adalah 3, atau  $\chi(G) = 3$ .



Gambar 2.22 Sebuah pewarnaan-3 dari  $G$

Sumber Ketut :153

## 2.9 Visual Basic 6.0

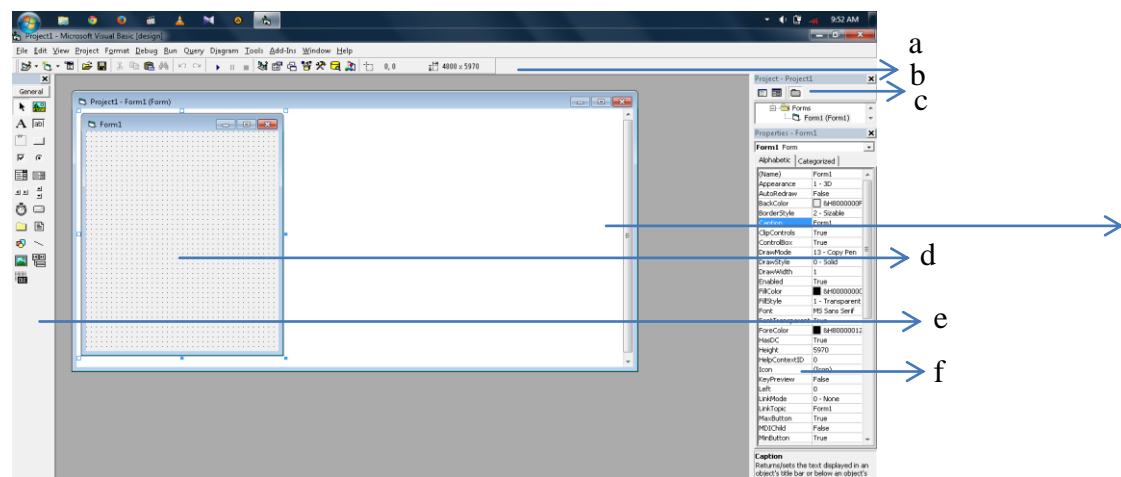
Microsoft *Visual Basic 6.0* merupakan bahasa pemrograman yang cukup populer dan mudah untuk dipelajari serta dapat membuat program dengan aplikasi GUI (*Graphical User Interface*) atau program yang memungkinkan pengguna komputer berkomunikasi dengan komputer tersebut dengan menggunakan modus grafik atau gambar (Hardianti, 2013: 35).

*Visual Basic* adalah salah satu *development tool* untuk membangun aplikasi dalam lingkungan windows. Dalam pengembangan aplikasi, *visual basic* menggunakan pendekatan visual untuk merancang *user interface* dalam bentuk form, sedangkan untuk kodingnya menggunakan pendekatan bahasa *basic*. *Basic* telah menjadi tools yang terkenal bagi para pemula maupun para developer dalam pengembangan aplikasi skala kecil sampai ke skala besar. Dalam lingkungan *Window's User-interface* sangat memegang peranan penting, karena dalam pemakaian aplikasi yang kita buat, pengguna senantiasa berinteraksi dengan *user interface* tanpa menyadari bahwa dibelakangnya berjalan instruksi-instruksi program yang mendukung tampilan dan proses yang dilakukan.

Pada pemrograman *Visual Basic*, pengembangan aplikasi dimulai dengan pembentukan *user interface*, kemudian mengatur properti dari objek-objek yang digunakan dalam *user interface*, dan baru dilakukan penulisan kode program untuk menangani kejadian-kejadian (*event*). Tahap pengembangan aplikasi demikian dikenal dengan istilah pengembangan aplikasi dengan pendekatan *Bottom Up*.

### 2.9.1 Komponen Utama *Microsoft Visual Basic*

Mengenal komponen-komponen *Visual Basic* merupakan hal yang sangat penting. Komponen-komponen ini akan membantu kita dalam pembuatan program (Hardianti, 2013: 35). Pertama kali menjalankan *Visual Basic* akan tampil beberapa komponen, yaitu Baris Menu, *Toolbar*, *Form*, Jendela Projek, Jendela *Properties*, dan Jendela *Form Layout*, seperti terlihat pada Gambar 2.23.



Gambar 2.23 Tampilan Project Visual Basic

#### a. Baris Menu

Menyimpan seluruh perintah yang terdapat pada *Visual Basic*.

#### b. *Toolbox*

Merupakan kumpulan ikon-ikon objek untuk membuat tampilan program atau *form*.

#### c. *Toolbar*



Merupakan kumpulan ikon-ikon perintah yang sering dipakai pada *Visual Basic*.

d. *Form*

Tempat untuk meletakkan objek-objek sebagai tampilan program.

e. *Jendela Properties*

Adalah jendela berisi *properties* (karakteristik) form dan objek-objek yang ada dalam form tersebut.

f. *Jendela Project*

Adalah jendela berisi project, form-form, modul-modul dan lainnya yang berhubungan dengan project yang kita buat.