BAB 2

TINJAUAN PUSTAKA

Bab ini terdiri dari dua subbab, yaitu subbab mengenai tinjauan pustaka dan subbab mengenai landasan teori. Dalam tinjauan pustaka, dikemukakan beberapa pustaka terdahulu yang relevan dengan permasalahan yang dikaji dalam penelitian ini. Dalam landasan teori, dikemukakan konsep dasar teori yang digunakan dalam penelitian ini; terutama berkaitan dengan penerapan algoritma Welch-Powell untuk melakukan pewarnaan *vertex* pada *graph*.

2.1 Tinjauan Pustaka

Menurut Buckley & Lewinter (2003) sebuah *graph* mempunyai peran yang besar dalam kehidupan manusia. *Vertex* sebuah *graph* dapat merepresentasikan orang, komputer, kota. Dan *edge* menunjukkan *vertex-vertex* yang memiliki hubungan. Sebagai contoh, dua orang saling mengenal, dua komputer yang dapat berkomunikasi atau dua kota yang dihubungkan oleh jalur penerbangan. Sehingga sebuah *graph* dapat diartikan sebagai representasi matematis dari sebuah hubungan. Teori *graph* digunakan untk menyederhanakan masalah-masalah yang ada didunia nyata sehingga hubungan antar komponen dalam suatu masalah dapat dipahami dengan mudah.

Menurut Hutabarat (2009), pewarnaan simpul (*vertex coloring*) suatu *graph* adalah pemberian warna terdapat *vertex* sedemikian hingga dua *vertex* yang berdampingan mempunyai warna yang berlainan. Sebuah *vertex* dapat diberikan sembarang warna asalkan warna yang diberikan berbeda dengan *vertex* yang berdekatan dengannya. Dikatakan G berwarna n, bila terdapat pewarnaan dengan menggunakan warna.

Sendra (2009) menggunakan pewarnaan *graph* khususnya pewarnaan *vertex* dalam menemukan solusi untuk membantu penjadwalan matakuliah. Hasil penelitian yang diperoleh Sendra (2009), model relasi *graph* matakuliah yang dirancang pada sistem dan pemberian warna pada *graph* menggunakan algoritma Welch-Powell

dapat membentuk paket-paket matakuliah yang bebas dari tabrakan. Sedangkan Bidhi (2010) menggunakan algoritma Welch-Powell dalam melakukan pewarnaan *graph* untuk membuat program bantu pengambilan matakuliah. Pada program bantu pengambilan matakuliah yang dirancang Bidhi (2010), diperoleh kesimpulan algoritma Welch-Powell dapat diterapkan pada aplikasi progaram bantu pengambilan matakuliah tersebut, karena jadwal yang dihasilkan akurat.

2.2 Landasan Teori

2.2.1 Definisi *Graph*

Sebuah *graph*, disimbolkan dengan G(V,E), merupakan struktur matematis yang terdiri dari dua himpunan V dan E. Elemen-elemen dari V disebut dengan *vertex* atau *node*, dan elemen dari E disebut dengan *edge*. Setiap *edge* mempunyai satu atau dua *vertex* yang berasosiasi dengannya yang disebut dengan *endpoints* (Gross & Yellen, 1999).

Menurut Ardiansyah, dkk (2010), sebuah *graph* itu sendiri adalah sebagai pasangan himpunan (V, E) yang dalam hal ini:

V = himpunan tidak kosong dari *vertex-vertex* (*vertices* atau node) dan

E = himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang *vertex*.

Dalam notasi matematika, *graph* dapat ditulis dengan **G(V, E).**

2.2.2 Jenis-jenis *Graph* yang umum

Adapun beberapa jenis *graph* yang sudah dikenal secara umum yang diambil dari http://mathworld.wolfram.com adalah sebagai berikut :

- *Graph* Sederhana

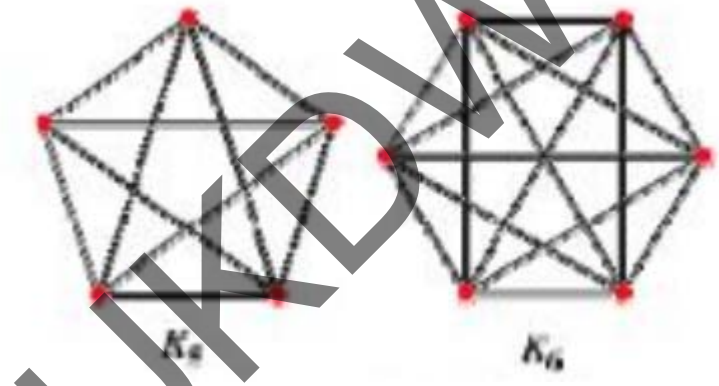
Adalah *graph* yang tidak berbobot dan tidak berarah yang tidak mempunyai *loop* dan *multiple edge*.

Gambar 2.1 Contoh *Graph* Seederhana

Diambil dari: http://mathworld.wolfram.com/Simple*Graph*.html

- *Graph* Komplit

  Adalah *graph* dimana setiap pasang *vertex* dari *graph* dihubungkan edge. Komplit *graph* dengan n *vertex* disimbolkan dengan Kn.



Gambar 2.2 Contoh *Graph* Komplit K5 & K6

Diambil dari: http://mathworld.wolfram.com/CompleteGraph.html

- *Graph* kosong

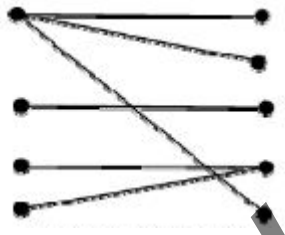
  Adalah *graph* yang terdiri dari *vertex* atau *node* saja tanpa *edge*.

Gambar 2.3 Contoh *Graph* Kosong

Diambil dari: http://mathworld.wolfram.com/EmptyGraph.html

- *Graph* Bipartite

Disebut juga dengan *bigraph*, adalah suatu kumpulan vertex dari sebuah *graph* yang dibagi menjadi dua himpunan sehingga tidak ada dua vertex dalam himpunan yang sama yang ber-*adjacent*.

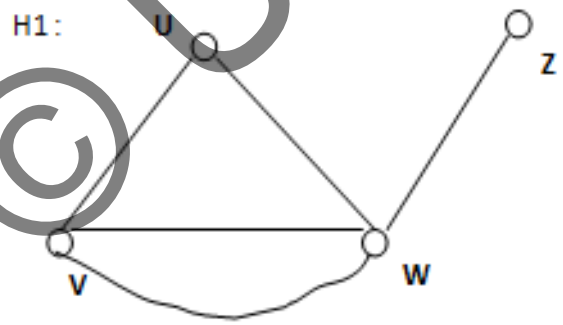


Gambar 2.4 Contoh *Graph* Bipartite

Diambil dari: http://mathworld.wolfram.com/BipartiteGraph.html

2.2.3 Derajat Titik

Misal G adalah *graph* yang tanpa *loop*, maka derajat suatu titik v di G, dinyatakan dengan deg(v), dimana v adalah suatu *vertex* di G. Derajat *degree* dari v adalah *edge* yang bertemu di v. Untuk *loop*, apabila suatu *vertex* tunggal dengan *loop* maka deg(v)=2



Gambar 2.5 *Graph* yang tidak memiliki *Loop*

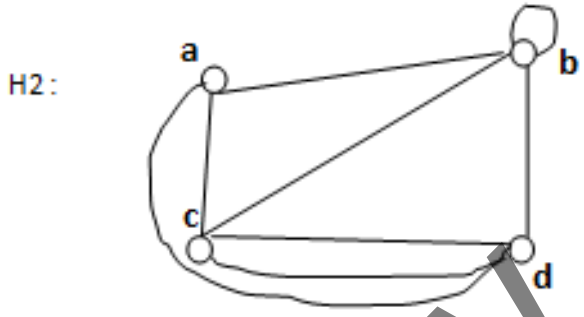
Pada gambar 2.5, dapat dihitung masing masing derajat dari *vertex*nya dengan melihat *edge* yang dimiliki dari *vertex* tersebut.

Deg (U) = 2

Deg (V) = 3

Deg (W) = 4

Deg (Z) = 1



Gambar 2.6 *Graph* yang memiliki *Loop*

Pada gambar 2.7, pada v(b) terdapat *loop*, maka derajat dari *vertex* pada *graph* H2 adalah:

Deg (a) = 3

Deg (b) = 5

Deg (c) = 4

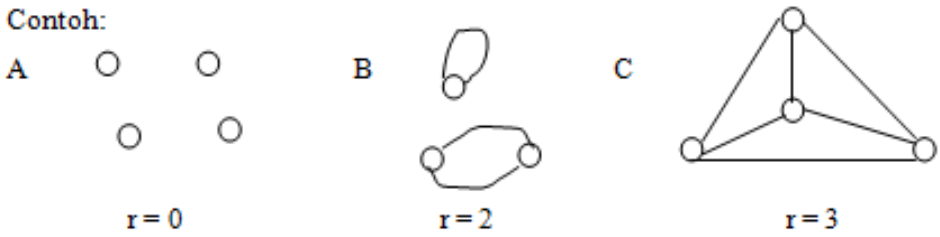
Deg (d) = 4

Pendaftaran derajat-derajat suatu *graph* disebut *degree sequence* atau barisan derajat, yang merupakan deret atau barisan bilangan bulat positif.

Pada *graph* H1 memiliki barisan derajat : (1, 2, 3, 4)

Pada *graph* H2 memiliki barisan derajat : (3, 4, 4, 5)

Jika semua *vertex* berderajat sama maka *graph* tersebut dinamakan *graph* reguler, atau sering disebut dengan *graph* reguler derajat r.



Gambar 2.7 *Graph* Reguler derajat r

Pada gambar 2.8, *graph* A memiliki *graph* reguler derajat = 0 , sedangkan *graph* B = 2, dan C = 3.

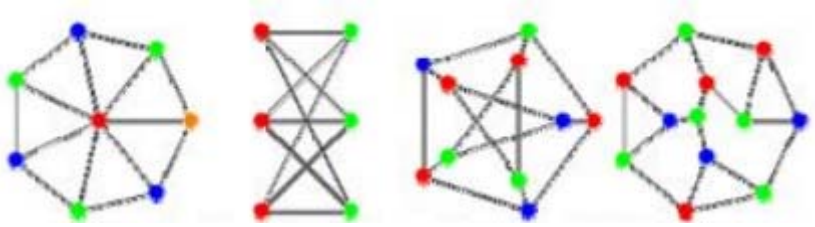
2.2.4 Pewarnaan *Graph*

Pada jurnal informatika yang ditulis oleh Ardiansyah dkk (2010) dijelaskan bahwa masalah pewarnaan *graph* diyakini pertama kali muncul sebagai masalah pewarnaan peta, dimana warna setiap daerah pada peta yang berbatasan dibuat berlainan sehingga mudah untuk dibedakan. Hal ini kemudian mengembangkan teorema-teorema menarik dan berujung pada teorema 4 warna, yang menyatakan: **" bilangan kromatik *graph* planar tidak lebih dari 4."**. Teorema ini pertama kali muncul sebagai suatu perkiraan oleh Francis Guthrie, seorang mantan murid dari Augustus De Morgan, pada tahun 1852 dan akhirnya dibuktikan oleh Kenneth Appel dan Wolfgang Haken. Pembuktian teorema ini menggunakan komputer dengan waktu yang melebihi 1000 jam.

Pewarnaan *graph* adalah suatu proses pemberian label atau warna pada *edge* atau *vertex* dari sebuah *graph*. Ada tiga macam pewarnaan *graph*, yaitu pewarnaan *vertex*, pewarnaan *edge*, dan pewarnaan wilayah (*region*).

Pewarnaan *vertex* adalah suatu proses pemberian label atau warna pada *vertex* dari suatu *graph* sehingga *vertex* yang berdampingan (*adjacent*) memiliki label atau warna yang berbeda (Buckley & Lewinter, 2003, hlm 165). Sebuah pewarnaan *graph* G adalah sebuah pemetaan warna-warna ke *vertex* G sehingga *vertex adjacency* atau

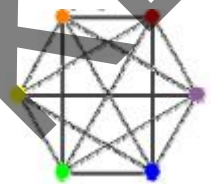
vertex yang berdampingan mempunyai warna yang berbeda. *Graph* planar G dikatakan berwarna n jika terdapat sebuah pewarnaan dari G yang menggunakan n warna. Berikut ini adalah gambar *graph* dengan pewarnaan *vertex* :



Gambar 2.8 *Graph* dengan pewarnaan *vertex*

Diambil dari: http://mathworld.wolfram.com/*Vertex* Coloring.html

Jumlah warna minimum yang diperlukan untuk mewarnai G disebut bilangan *chromatic* (kromatik) dari G atau dapat disimbolkan dengan K(G). Sebuah *graph* komplit (Kn) memiliki bilangan kromatik K sebanyak n, artinya warna minimal yang dibutuhkan *graph* komplit adalah sebanyak n warna.



Gambar 2.9 *Graph* komplit (K6)

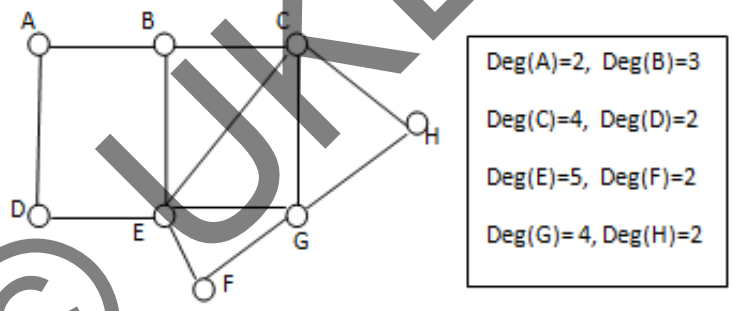
Diambil dari: http://mathworld.wolfram.com/ChromaticNumber.html

Pemberian warna minimal pada suatu *graph* merupakan pekerjaan yang sulit. Sehingga dibutuhkan suatu algoritma tertentu untuk pemberian warna dari suatu *graph*. A. Schaerf melalui penelitiannya yang berjudul *A Survey of Automated Timetabling* melakukan survei terhadap berbagai macam algoritma yang digunakan dalam proses penjadwalan. Salah satu algoritma pewarnaa *graph* yang dijelaskan dalam penelitiannya adalah algoritma welch-powell. Schaerf (1999) menyatakan bahwa pada algoritma welch-powell, *vertex* dengan derajat terbesar diwarnai terlebih

dahulu, sebab *vertex* dengan derajat yang besar merupakan *vertex* yang paling sulit diwarnai. Berikut adalah langkah-langkah dari algoritma welch-powell:

1. Urutkan semua *vertex* berdasarkan derajatnya, dari derajat besar ke derajat kecil.
2. Ambil warna pertama (misalnya merah), warnai *vertex* pertama yang sudah kita urutkan berdasarkan derajatnya pada langkah pertama. Kemudian warnai *vertex* yang tidak berdampingan dengan *vertex* pertama tadi dengan warna yang masih sama (merah).
3. Mulai lagi dengan urutan derajat yang paling tinggi berikutnya yang belum diwarnai
4. Kemudian ulangi langkah kedua dengan warna kedua, dan seterusnya, sampai semua *vertex* telah diberi warna.

Contoh :



Gambar 2.10a *Graph* yang akan diwarnai

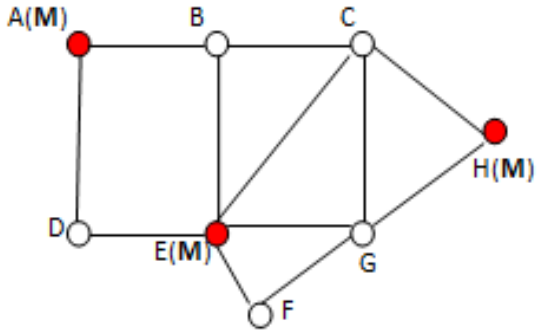
Langkah1:

Urutkan *vertex* dari derajat terbesar kederajat yang lebih kecil, maka didapat urutan *vertex* adalah: **E,C,G,B,A,D,F,H**

Langkah 2:

Vertex **E** diberi warna pertama, misal warna merah (**M**) , kemudian dicari *vertex* yang telah diurutkan yang tidak berdampingan atau berhubungan dengan *vertex* **E** maka:

Vertex A dan H tidak berdampingan dengan *vertex* E, maka diwarnai merah (**M**) juga.

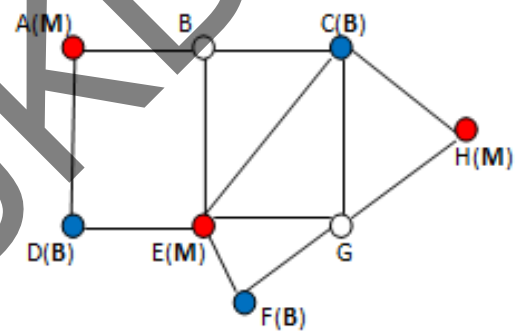


Gambar 2.10b *Graph* diwarnai dengan warna ke-1

Langkah 3:

Sisa urutan *vertex* adalah **C,G,B,D,F.** Maka *vertex* **C** akan diberi warna kedua, misalnya warna biru (**B**), dan semua *vertex* yang tidak berdampingan dengan *vertex* C juga diberi warna biru.

Vertex D dan F tidak berdampingan dengan *vertex* C, maka diwarnai biru (**B**) juga.

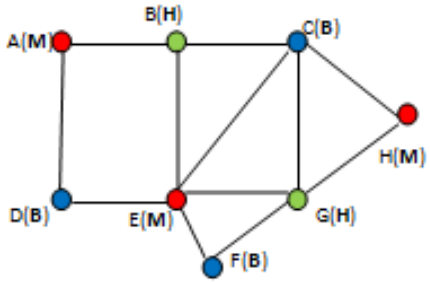


Gambar 2.10c *Graph* diwarnai dengan warna ke-2

Langkah 4:

Sisa *vertex* adalah **B,G**. *Vertex* **B** diberi warna ketiga, misalnya warna hijau, semua *vertex* yang tidak berdampingan dengan *vertex* **B** diberi warna hijau juga.

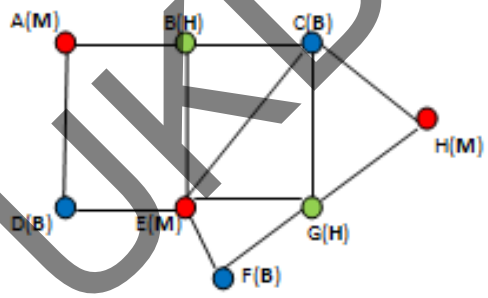
Vertex G tidak berdampingan dengan *vertex* B, maka diwarnai hijau (**H**) juga.



Gambar 2.10d *Graph* diwarnai dengan warna ke-3

Langkah 5:

Pewarnaan *Graph* menggunakan algoritma Welch-Powell telah selesai, hasilnya adalah:



Gambar 2.10e *Graph* yang telah selesai diwarnai

Hasil akhir pewarnaan yang diperoleh adalah:

Vertex **A**, **E** dan **H** diberi warna merah **(M)**

Vertex **C**, **D** dan **F** diberi warna biru **(B)**

Vertex **B** dan **G** diberi warna hijau **(H)**

Bilangan Kromatik (**K(G)**) atau jumlah warna minimum yang dibutuhkan *graph* pada gambar 2.11 e adalah **3** warna**.**

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

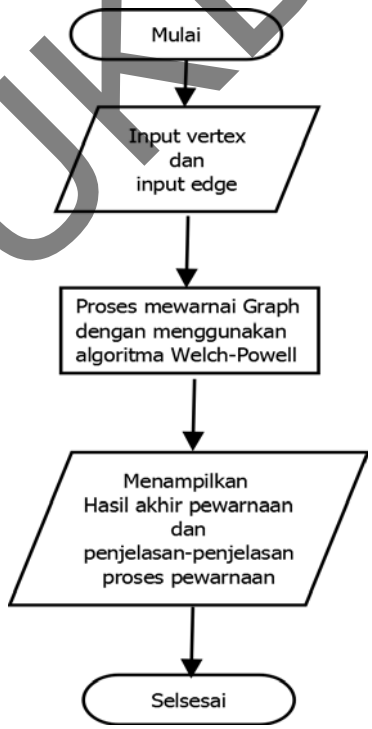
Analisis dan perancangan sistem terdiri dari empat subbab yaitu pemilihan bahasa pemrograman, perancangan proses, perancangan tampilan, dan spesifikasi sistem. Secara keseluruhan bab ini membahas tentang rancangan dari sistem yang akan dibuat.

3.1 Pemilihan Bahasa Pemrograman

Dalam perancangan aplikasi simulasi pewarnaan *graph*, digunakan bahasa pemrograman *Microsoft Visual Basic 2005 Express Edition.*

3.2 Gambaran Kerja Sistem

Gambaran kerja sistem secara umum dapat terlihat pada gambar 3.1.



Gambar 3.1 Flowchart program secara umum

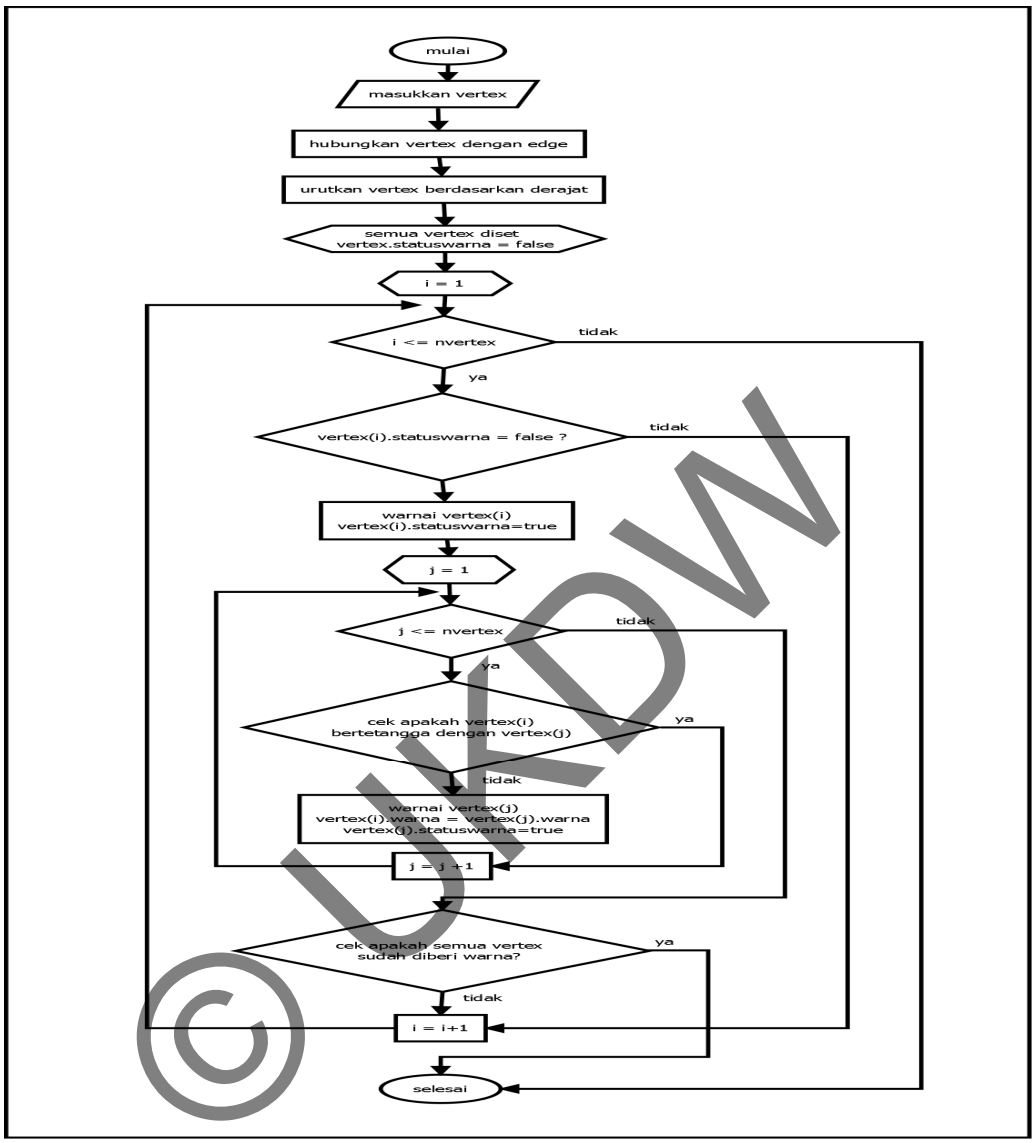
Program akan dimulai dengan *user* meng-*input*-kan *vertex*, sedangkan *edge* bisa di-*input*-kan ketika *vertex* telah diinputkan oleh *user*, jika *vertex* belum diinputkan maka *edge* tidak dapat diinputkan.

3.2.1 Proses Pewarnaan *Graph* menggunakan Algoritma Welch-Powell

Proses kerja algoritma Welch-Powell dalam sistem ini adalah sebagai berikut :

1. Mulai
2. Masukkan *vertex* sebanyak yang diinginkan, minimal 1 maximal 100
3. Hubungkan *vertex-vertex* dengan *edge*
4. Urutkan *vertex-vertex* berdasarkan banyaknya *edge* atau derajat yang dimiliki setiap *vertex*
5. Lakukan pewarnaan pertama terhadap *vertex* yang telah diurutkan pada langkah
6. Cek apakah *vertex* diurutan berikutnya merupakan tetangga atau terhubung dengan *vertex* yang telah diwarnai pada langkah 6
7. Jika tetangga maka ulangi langkah 7, jika tidak maka warnai dengan warna yang sama dengan langkah 6
8. Setelah semua *vertex* yang memungkinkan diwarnai sama dengan warna pada langkah 6 maka pewarnaan dengan warna pertama selesai
9. Cek apakah *vertex* ada yang belum diwarnai, jika tidak pewarnaan selsesai, jika iya maka ulangi langkah ke 4 dan lakukan pewarnaan dengan warna kedua
10. Lakukan pewarnaan sampai semua *vertex* diwarnai
11. selesai

Flowchart dari proses kerja algoritma Welch-Powell di atas dapat dilihat dalam gambar 3.2 berikut :



Gambar 3.2 Flowchart Algoritma Welch-Powell

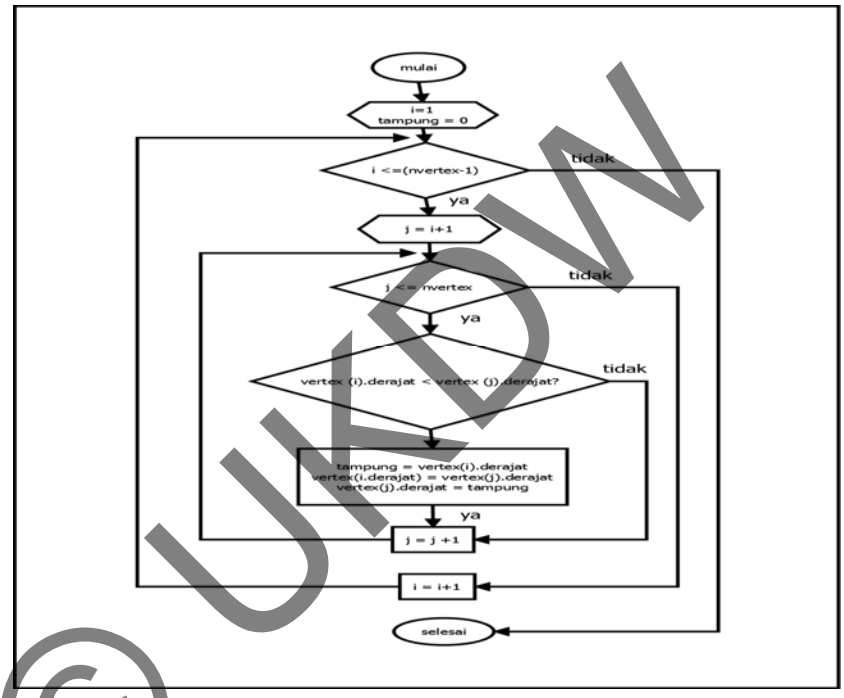
3.2.2 Proses Pengurutan *Vertex*

Proses kerja mengurutkan vertex dalam sistem ini menggunakan metode Exchange Sort, adapun proses pengurutannya adalah sebagai berikut :

1. Mulai
2. Bandingkan banyaknya *edge* dari *vertex* indeks awal dengan indeks berikutnya

3. Cek apakah *edge* dari *vertex* indeks awal lebih kecil dari index berikutnya, jika iya maka lakukan pertukaran indeks *vertex*, jika tidak maka lakukan lagi langkah 2 sampai n *vertex*
4. Selesai

Flowchart dari proses pengurutan *verex* di atas dapat dilihat dalam gambar 3.3 berikut :



Gambar 3.3 Flowchart Pengurutan *Vertex* berdasarkan banyak derajat

3. 3 Perancangan Struktur Data

3.3.1 Perancangan Struktur Data untuk *Vertex*

Untuk merepresantikan *vertex*, dibuat sebuah *struct* seperti yang ditunjukkan gambar 3.4. selain *struct*, untuk merepresantikan *vertex* juga dibuatkan sebuah *array* yang menampung banyaknya jumlah *vertex* yaitu *varvertex (100)* yang dideklarasikan bertipe *struct* sehingga variabel-variabel dari *struct* untuk *vertex* dapat diakses oleh *array* banyaknya jumlah vertex yaitu *varvertex (100).*

```
Public Structure tipevertex
    Dim x As Single
    Dim y As Single
    Dim derajat As Integer
    Dim r As Byte
    Dim g As Byte
    Dim b As Byte
End Structure
```

Gambar 3.4 *Struct* untuk merepresantikan *vertex*

3.3.2 Perancangan Struktur Data untuk *Edge*

Untuk merepresantikan *edge juga* dibuat sebuah *struct* seperti yang ditunjukkan gambar 3.5. selain memiliki *struct*, untuk merepresantikan *edgex* juga dibuatkan sebuah *array* yang menampung banyaknya jumlah *edge* yang digunakan untuk menghubungkan *vertex* yaitu *varedge (1000)* yang dideklarasikan bertipe *struct* sehingga variabel-variabel dari *struct* untuk *edge* dapat diakses oleh *array* banyaknya *edge*.

```
Public Structure tipeedge
    Dim id1 As Integer
    Dim id2 As Integer
    Dim x1 As Single
    Dim y1 As Single
    Dim x2 As Single
    Dim y2 As Single
End Structure
```

Gambar 3.5 *Struct* untuk merepresantikan *edge*

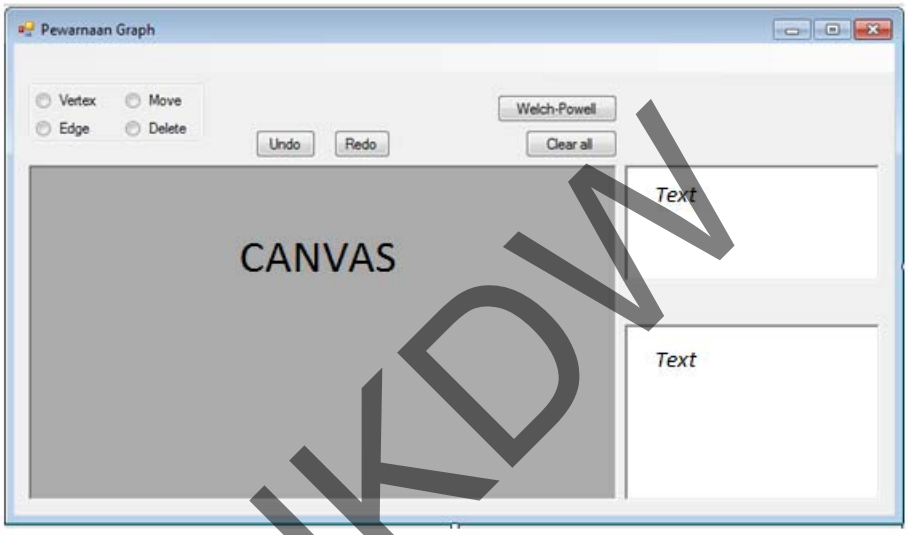
3. 4 Perancangan Antarmuka Sistem

Perancangan antarmuka menggambarkan bagaimana tampilan dari sistem yang akan dibuat.

3.4.1 Perancangan *Form Input*

Gambar 3.5 menunjukan tampilan *form* untuk menggambarkan *graph*. *User* akan meng-*input vertex* dengan cara memilih *radio button vertex* di atas *canvas* kemudian menghubungkan dengan *edge* dengan cara memilih *radio button edge*. Jika

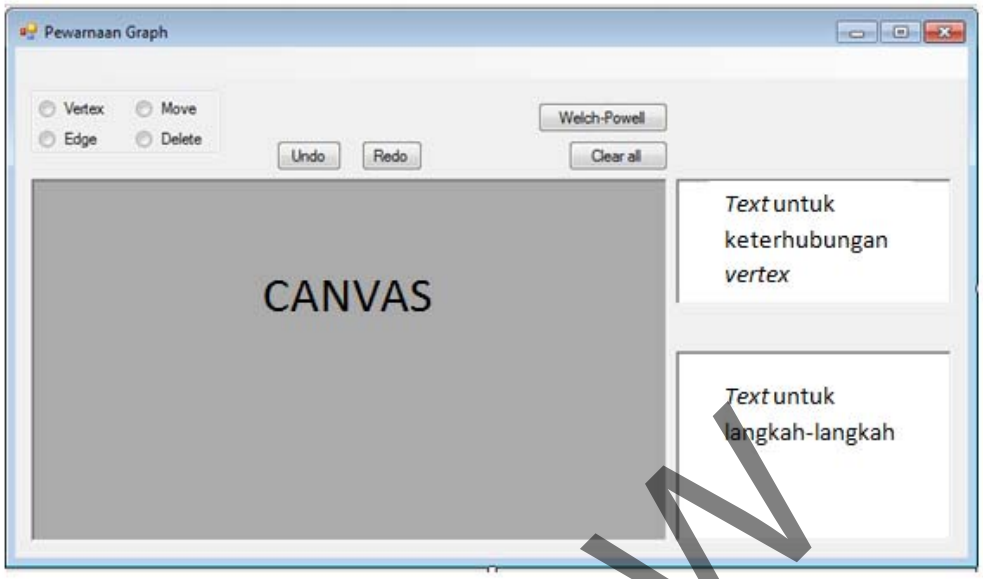
user ingin mengubah posisi *vertex*, maka *user* dapat memilih *radio button move*. *User* dapat melakukan *undo* dan *redo* pada saat menggambar dengan mengklik tombol *undo* atau *redo*. Untuk menghapus dan membersihkan *canvas* dari gambar *graph, user* dapat mengklik *button clear all*, untuk melihat keterhubungan antar *vertex user* dapat mengklik *button* keterhubungan. Setelah selesai menggambar graph, *user* dapat mengklik *button* Welch-Powell untuk menjalankan proses pewarnaan *graph.*



Gambar 3.5 *Form Input*

3.4.2 Perancangan *Form output*

Gambar 3.6 menunjukan tampilan *form output* ataupun hasil keluaran dari proses yang dijalankan dari *input*-an *user* setelah menekan *button Welch-Powell*. *Form output* dibuat satu *form* dengan *form input. Form output* menampilkan langkah-langkah dan proses pewarnaan *graph* dengan menggunakan algoritma Welch-Powell. Langkah-langkah pewarnaan dapat dilihat di *rich text box summary*. Sedangkan ketika *user* ingin melihat keterhubungan antar *vertex* maka dapat dilihat di *rich text box* dibawah *button keterhubungan*.

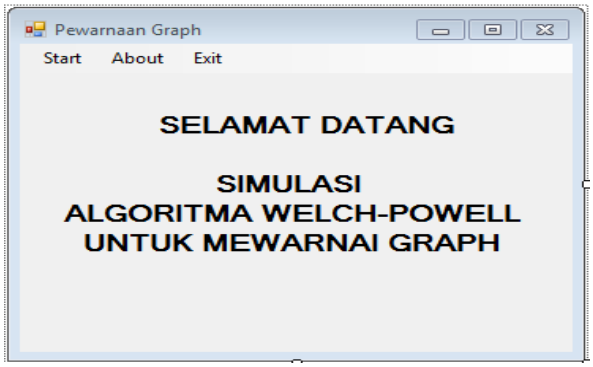


Gambar 3.6 *Form output*

3.5 Perancangan *Form* Tambahan

3.5.1 Perancangan *Form* Utama

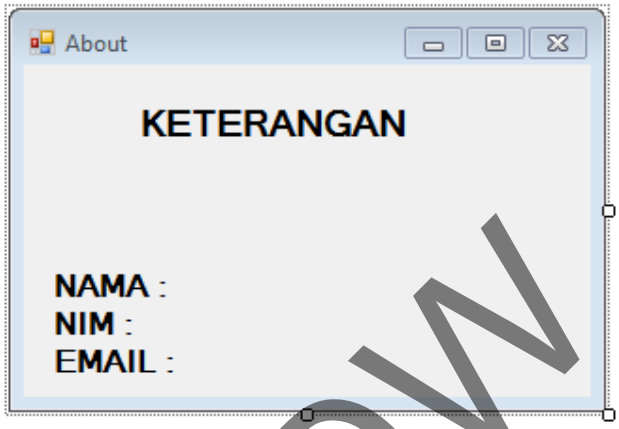
Pada tampilan *form* utama ini, *user* diberi tiga pilihan, yaitu *start*, *About*, dan *exit*. Melalui tiga pilihan itu, *user* dapat memilih *button Start* apabila ingin memulai. *User* juga dapat memilih *button About* apabila ingin mendapatkan informasi yang disediakan oleh program. Selain itu, *user* dapat memilih *Keluar* apabila menginginkan keluar dari aplikasi. Gambar 3.7 berikut ini menunjukkan tampilan *form* Utama.



Gambar 3.7 *Form* Utama

3.5.2 Perancangan *Form About*

Form about berisi keterangan, nama, NIM dan *email* penyusun atau penulis. Gambar 3.8 berikut ini merupakan perancangan tampilan *form about.*



Gambar 3.8 *Form About*

3.6 Perancangan Pengujian algoritma Welch-Powell

1. **Tujuan pengujian**

   Tujuan pengujian pewarnaan adalah untuk mendapatkan hasil optimal kemungkinan warna yang dibutuhkan untuk melakukan pewarnaan pada *graph* yang diperoleh dengan menggunakan algoritma Welch-Powell dan pengaruh *graph* komplit terhadap jumlah warna.

2. **Bahan dan alat**

   Bahan yang dibutuhkan untuk melaksanakan penelitian ini adalah teori-teori pendukung tentang graf dan pewarnaan graf dalam mengimplementasikan ke dalam bahasa pemrograman Visual Basic.

   Adapun alat yang digunakan adalah perangkat keras (*hardware*) danperangkat lunak (*software*).

   a. Perangkat Keras

   Spesifikasi perangkat keras yang digunakan dalam membangun aplikasi simulasi algoritma Welch-Powell untuk melakukan pewarnaan *Graph* adalah sebagai berikut.

1. *Processor* : *Intel*(R) *Core*(TM)*2Duo* @2, 2*0GHz* (2CPU)
2. *Memory* : 2 GB
3. *Harddisk* : 110 GB

b. Perangkat Lunak

Perangkat lunak yang digunakan dalam membangun aplikasi aplikasi simulasi algoritma Welch-Powell untuk melakukan pewarnaan *Graph* adalah *Microsoft Visual Basic 2005 Express edition.* Di samping itu, untuk menggunakan aplikasi aplikasi simulasi algoritma Welch-Powell untk melakukan pewarnaan *Graph* tidak perlu menggunakan atau meng-*install* aplikasi apapun.

3. Mekanisme pengujian

a. Diambil sembarang *graph,* dari sembarang *graph* tersebut dicari *graph* komplit yang ada.
b. Jalankan algoritma Welch-Powell, sehingga memperoleh jumlah warna yang paling minimal
c. Setelah memperoleh hasil warna yang minimal dari algoritma Welch-Powell, analisis apakah hasil pewarnaan minimal sama dengan jumlah n pada bagian *graph* komplit (Kn)

4. Indikator pengujian

Pada algoritma Welch-Powell, *vertex* yang diwarnai terlebih dulu adalah *vertex* dengan derajat yang paling besar terlebih dahulu, *vertex* yang saling berhubungan atau bertetangga tidak boleh diberi warna yang sama.

BAB 4

IMPLEMENTASI DAN ANALISIS SISTEM

Implementasi dan analisis sistem menggambarkan implementasi dari rancangan yang telah dibuat yang dikemukakan pada bab 3, yaitu proses kerja sistem dan analisis penerapan algoritma Welch-Powell untuk mencari jumlah warna terkecil dalam melakukan pewarnaan *graph*. Berikut ini akan dikemukakan hal-hal yang terkait dengan implementasi dan analisis sistem tersebut.

4.1 Implementasi Sistem

Implementasi sistem berfungsi untuk menjelaskan implementasi dari rancangan tampilan dan rancangan proses kerja sistem. Subbab ini menjelaskan tentang *form* yang terdapat dalam aplikasi simulasi algoritma Welch-Powell

4.1.1 Implementasi Rancangan Tampilan

Aplikasi simulasi algoritma Welch-Powell untuk melakukan pewarnaan *graph* ada 3 *form*, yaitu satu *form* utama, satu *form input* yang digabung *form output*, dan satu *form* tambahan. Berikut akan dibahas tentang implementasi rancangan tampilan *form input*, proses, *ouput*, dan *form* tambahan.

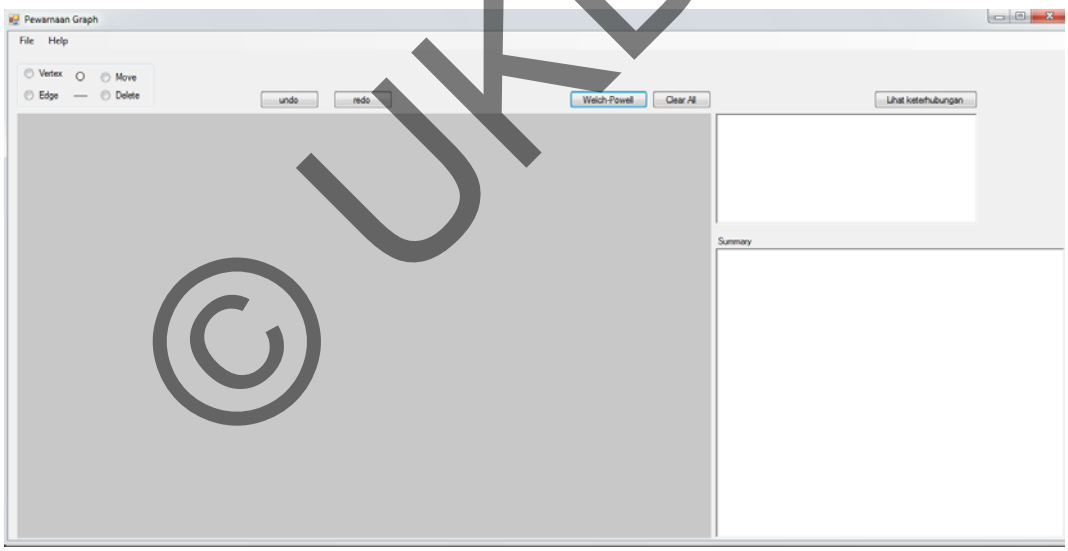
4.1.1.1 *Form Input*

4.1.1.1.1 *Form* Utama

Form utama mempunyai satu *menu strip* yang berisi *Start*, *About*, dan *exit*, seperti yang terlihat pada gambar 4.1 berikut ini.

Gambar 4.1 Tampilan Utama Menu

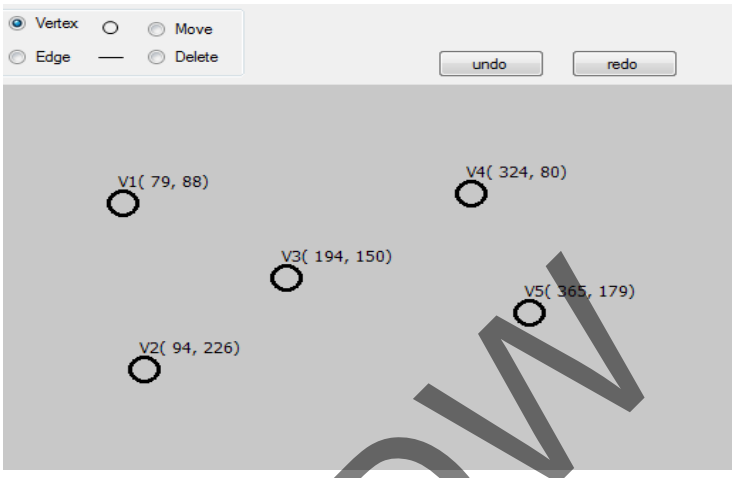
4.1.1.1.2 *Form input*



Gambar 4.2 Tampilan *Form input*

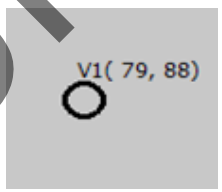
Gambar 4.3 menunjukkan tampilan *form input* awal dari program. Di atas *canvas* ada empat *radio button*, yaitu masing-masing *radio button vertex, radio button edge, radio button move* dan *radio button delete*

Pada saat *user* memilih *radio button vertex*, program akan diset untuk *user* dapat melakukan penggambaran *vertex* pada *canvas*. Hal ini ditunjukkan pada gambar 4.4.



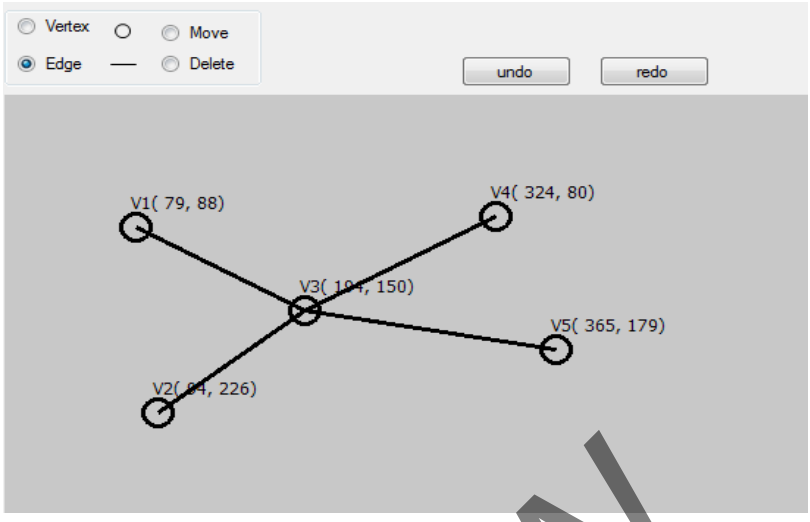
Gambar 4.3 Tampilan *Form input vertex*

Tampilan *vertex*, menunjukkan *vertex* inputan keberapa dan berada dikoordinat mana, hal ini dapat dilihat digambar 4.4, v1 sebagai contoh.



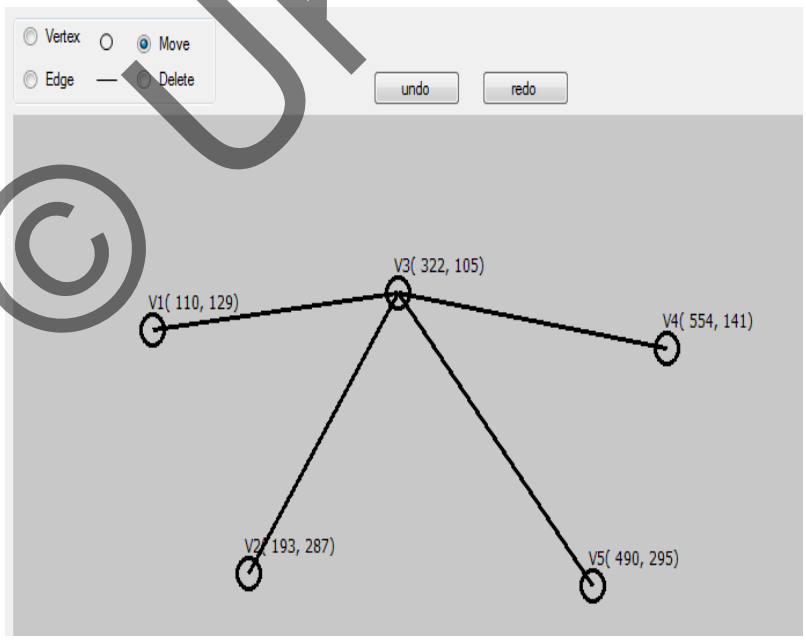
Gambar 4.4 Gambar *vertex* V1

Pada saat *user* memilih *radio button edge*, program akan diset untuk *user* dapat melakukan penggambaran *edge* pada *canvas* untuk menghubungkan *vertex*. Hal ini ditunjukkan pada gambar 4.5.



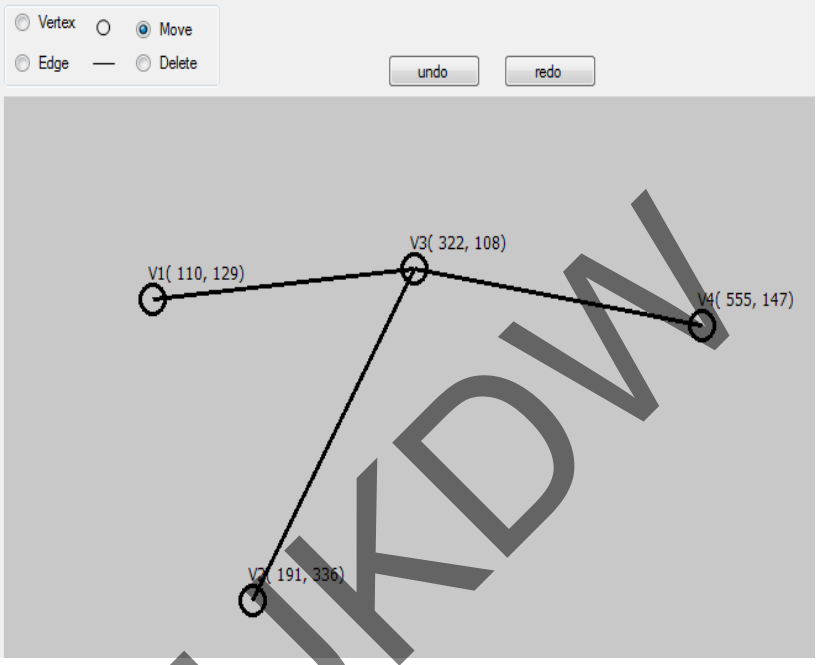
Gambar 4.5 Tampilan *Form input edge*

Pada saat *user* memilih *radio button move*, program akan diset untuk *user* dapat melakukan pemindahan lokasi *vertex* yang akan digeser. Contoh, akan digeser *vertex-vertex* yang ada pada gambar 4.5. maka hasil penggeseran *vertex* dapat dilihat pada gambar 4.6.



Gambar 4.6 Tampilan *Form input move*

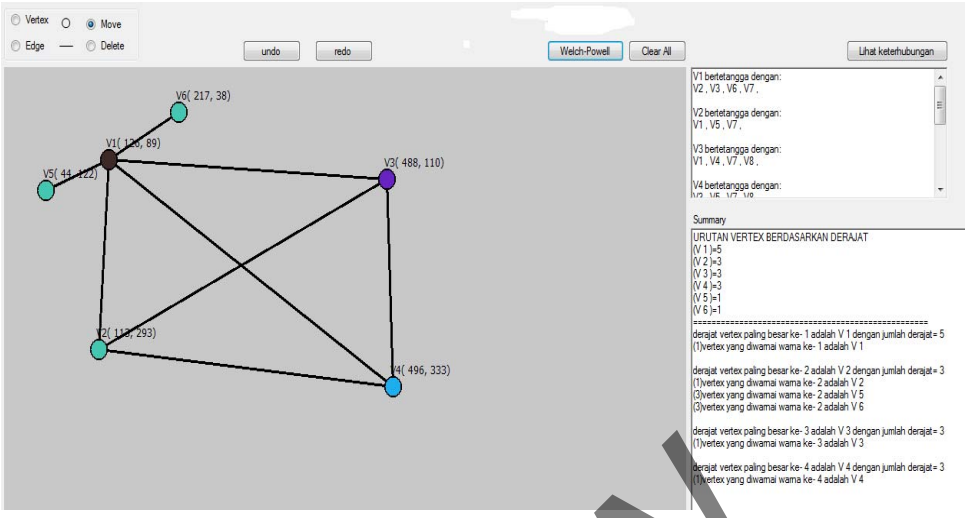
Pada gambar 4.6 dapat dilihat terjadi perubahan koordinat *vertex-vertex* yang ada pada gambar 4.5. sedangkan ketika *user* memilih *radio button delete*, maka *vertex* yang akan diilih untuk duhapus akan dihapus oleh sistem. Hal ini dapat dilihat pada gambar 4.7, sebagai contoh akan dihapus *vertex* v5.



Gambar 4.7 Tampilan *Form input delete*

4.1.1.1.3 *Form Ouput*

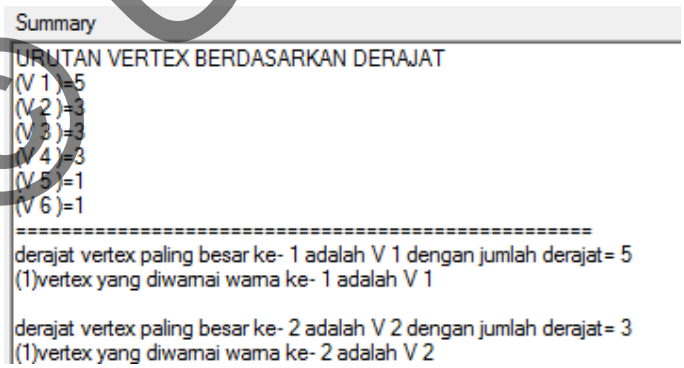
Algoritma dijalankan setelah *user* menekan *button* Welch-Powell. Hasil output yang akan ditampilkan berupa hasil vertex yang telah diwarnai pada *canvas*, seperti yang terlihat pada gambar 4.8.



Gambar 4.8

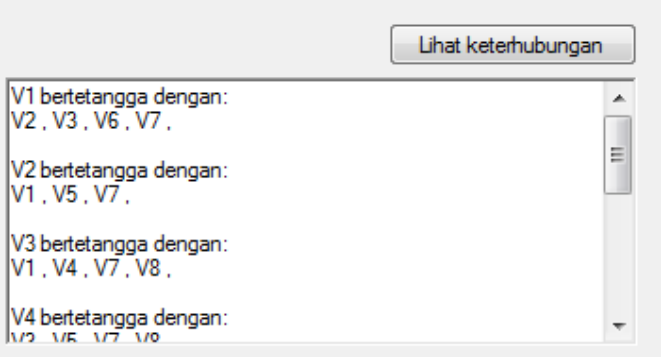
Tampilan *Form* saat melakukan pewarnaan menggunakan algoritma Welch-Powell

Sedangkan penjelasan langkah-langkah pewarnaan *graph* menggunakan algoritma Welch-Powell akan muncul pada *text box summary* seperti pada gambar 4.9, sehingga memudahkan *user* untuk mempelajari langkah demi langkah proses pewarnaan menggunakan algoritma Welch-Powell.



Gambar 4.9 Tampilan penjelasan langkah-langkah pewarnaan

Ketika *user* menekan *button* keterhubungan maka akan diperlihatkan hubungan antar *vertex-vertex*, seperti ditunjukkan pada gambar 4.10.

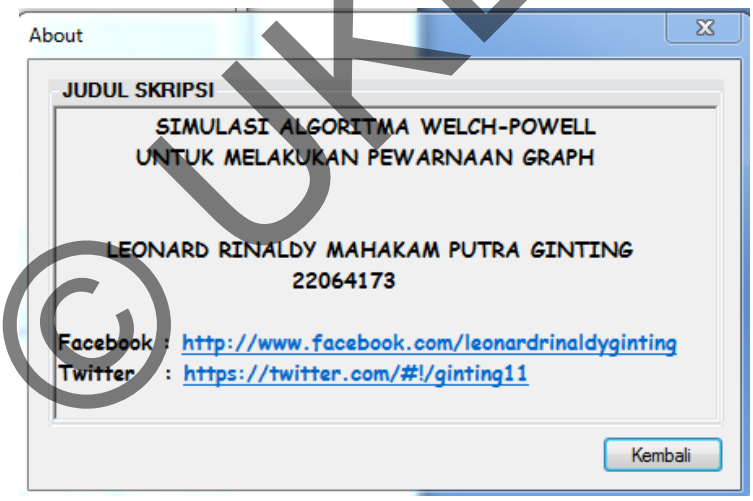


Gambar 4.10 Tampilan hubungan *vertex-vertex*

4.1.1.4 *Form* Tambahan

4.1.1.4.1 *Form About*

Gambar 4.11 menunjukkan tampilan *form about*. *Form* ini hanya berisi keterangan tentang aplikasi dan penulis. Selain itu hanya terdapat tombol *Kembali* untuk kembali ke *form* menu utama.



Gambar 4.11 Tampilan *Form About*

4.1.2. Implementasi Rancangan Proses

Pada subbab implementasi rancangan proses ini akan dibahas implementasi algoritma Welch-Powell untuk menentukan langkah terbaik dalam mencari jumlah warna minimum dalam melakukan pewarnaan *vertex* pada *graph*.

Dalam hal ini solusi dicari berdasarkan pengecekan warna disetiap *vertex* yang berhubungan sehingga didapatkan jumlah warna minimum dari sebuah *graph*. Proses pewarnaan ini dilakukan sampai semua *vertex* mendapatkan warna dan maing-masing *vertex* yang bertetangga atau berhubungan memiliki warna yang berbeda. Berikut ini adalah potongan *pseudocode* untuk melakukan pewarnaan pada *vertex.*

```
While (selesai = false)
        for i = 1 to banyaknya list vertex yang ada
        Set Max = 0
        Set Idmax t= 0
          If vertex(i).derajat = max And varvertex(i).derajat > max then
                    Set Max = vertex.derajat
                    Set Idmax=i
        End if
        If idmax=0 then
          Set selesai = true
        End if
        ' Warnai vertex idmax
        Set Vertex(idmax).statuswarna = true
        'cari tetangga
        For i = 1 to banyaknya list vertex
              If i <> idmax then
                    Set adaedge = false
                    For j = 1 to banyaknya list edge

                    'cek dari titik awal garis id1
                              If varedge(j).id1 = idmax And varedge(j).id2 = i then
                                        Set adaegde = true
                                        Set ntetangga = ntetangga+1
                                        Set tetangga(ntetangga) = i
                              End if

                    'cek dari titik akhir garis id2
                              If varedge(j).id2 = idmax And varedge(j).id2 = i then
                                        Set adaegde = true
                                        Set ntetangga = ntetangga+1
                                        Set tetangga(ntetangga) = i
                              End if
                    Next
              End if
        next
```

```
  ' cari yang bukan tetangganya
For i = 1 To nvertex
    If i <> idmax And varvertex(i).statuswarna = False Then
        Set adaedge = False
            For j = 1 To nedge
            ' cek dari id1
                If varedge(j).id1 = idmax And varedge(j).id2 = i Then
                    Set adaedge = True
                End If
            ' cek dari id2
                 If varedge(j).id2 = idmax And varedge(j).id1 = i Then
                     Set adaedge = True
                 End If
            Next
            If adaedge = false And i<= max then
              'warnai yang bukan tetangga
               Set Varvertex(i).statuswarna = true
            Exit for
            End if
    End if
  Next

  'cari yang bukan tetangga dari bukan tetangganya
  Set selesai2 = false
  Set Idmax=i
   While selesai2 = false
        For i=1 to banyaknya vertex dilist
            If i <> idmax then
              Set adaedge = false
              For j = 1 to banyanknya edge di list
                    'cek dari id1
                    If varedge(j) .id1 = idmax And varedge(j).id2 = i then
                            Set adaegde = true
                            Set ntetangga = ntetangga+1
                    End if
                    'cek dari id2
                    If varedge(j) .id2 = idmax And varedge(j).id1= i then
                            Set adaegde = true
                            Set ntetangga = ntetangga+1
                    End if
                Next
            End if
          Next
```

```
        set jumlahcari = 0
        for i=1 to nvertex    'banyaknya vertex dilist
          set adatetangga = false
            for j=1 to ntetangga  'banyaknya tetangga dilist
                    if i = vertextetangga(j) then
                              set adatetangga = true
                    exit for
                    end if
            next
             if i <> idmax And vertex(i).statuswarna = false and adatetangga =false then
                    set adaedge = false
                    For j = 1 To nedge
                    ' cek dari id1
                       If varedge(j).id1 = idmax And varedge(j).id2 = i Then
                            Set adaedge = True
                       End If
                    ' cek dari id2
                        If varedge(j).id2 = idmax And varedge(j).id1 = i Then
                             Set adaedge = True
                         End If
                    Next
                    If adaedge = false And i<= max then
                      'warnai yang bukan tetangga
                       Set Varvertex(i).statuswarna = true
                    Exit for
                    End if
             End if
            Next
                    idmax = i
                    jumlahcari += 1
                    Exit For
                    End If
             End If
             Next
            If jumlahcari = 0 Then
                     selesai2 = True
          End If
         End While
        End If
End While
```

Gambar 4.12 ***Pseudocode*** **Pewarnaan** ***graph***

Pewarnaan selesai diwarnai ketika kondisi semua vertex(i).statuswarna bernilai *true*, max bernilai 0 dan selesai bernilai *true*. Jadi proses pewarnaan dilakukan sebanyak tiga kali proses, yaitu :

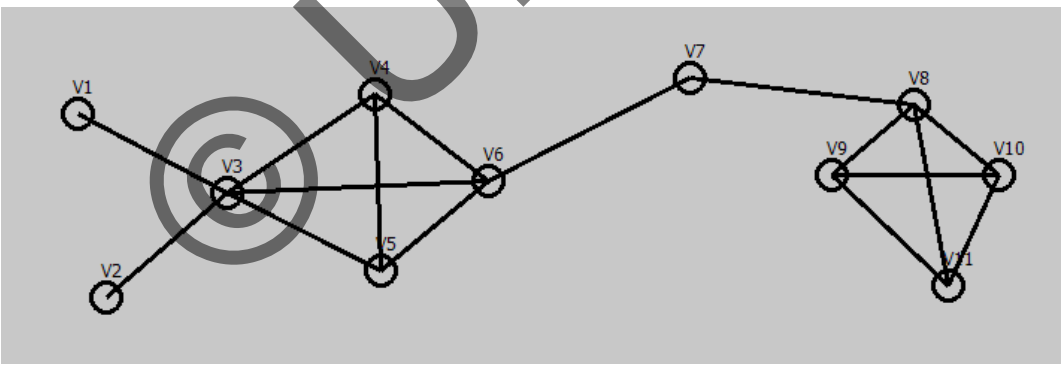
1. *vertex* berderajat besar (*max*) diwarnai terlebih dahulu
2. cari yang tidak berhubungan dengan *vertex max*, kemudian warnai sama dengan *vertex max*
3. kemudian cari yang bukan tetangga dari *vertex max* dan *vertex* yang tidak terhubung atau bukan tetangga dari *vertex* pada proses 2.

4.2 Analisis Sistem

Pada subbab ini akan dijelaskan analisis perbandingan jumlah warna *vertex* yang diwarnai menggunakan algoritma Welch-Powell dan pewarnaan manual yang *vertex* dengan derajat besar sebagai rootnya.

4.2.1. Analisis Pengujian Algoritma Welch-Powell

Gambar 4.13 menunjukkan kondisi awal *graph* sembarang yang telah digambarkan *user*.

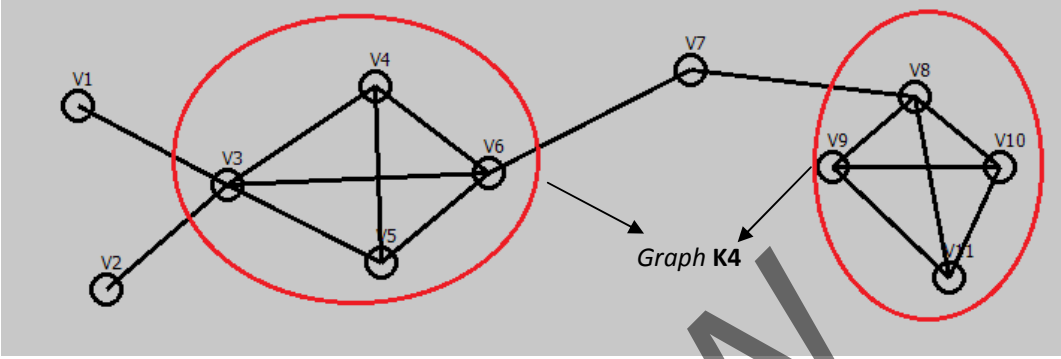


Gambar 4.13 Kondisi Awal *Graph*

Dari gambar 4.13 maka diperoleh derajat *vertex* sebagai berikut :

V1 = 1	V5 = 3	V9 = 3
V2 = 1	V6 = 4	V10 = 3
V3 = 5	V7 = 2	V11 = 3
V4 = 3	V8 = 4	

Gambar 4.13 menunjukkan bahwa *graph* memiliki *graph* komplit K4 yang ditunjukkan pada gambar 4.14. sehingga pada bagian *graph* komplit minimal warna yang dibutuhkan adalah 4.



Gambar 4.14 Bagian *Graph* yang membentuk *Graph* komplit

Akan dicari warna minimal *graph* sembarang seperti gambar 4.13 yang memiliki *graph* komplit K4. Apakah hasil bilangan kromatik K(G) atau jumlah warna minimal yang dibutuhkan untuk pewarnaan *graph* sembarang pada gambar 4.13

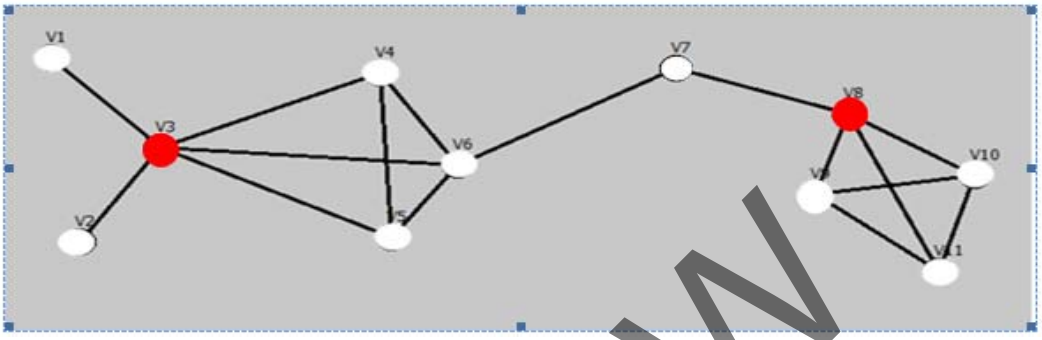
Langkah 1:

Urutkan vertex berdasarkan derajat, dimulai dari derajat terbesar. Diperoleh hasil pengurutan adalah sebagai berikut:

V3 = 5
V6 = 4
V8 = 4
V4 = 3
V5 = 3
V9 = 3
V10 = 3
V11 = 3
V7 = 2
V1 = 1
V2 = 1

Langkah 2:

V3 akan diwarnai terlebih dahulu dengan warna pertama yaitu diambil warna merah. Kemudian warnai *vertex* yang tidak berhubungan dengan V3 dengan warna pertama, seperti pada gambar 4.15

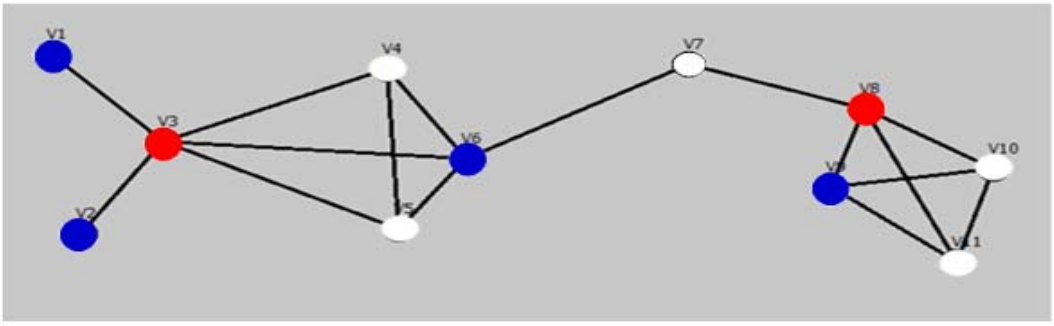


Gambar 4.15 Pewarnaan Welch-Powell dengan warna ke-1

Diperoleh V3 dan V8 *vertex* yang diwarnai dengan warna pertama. Cek apakah masih ada *vertex* yang belum diwarnai. Ternyata masih ada sehingga pewarnaan dilanjutkan dengan warna kedua.

Langkah 3:

Urutan *vertex* yang diwarnai berikutnya adalah V6 yang akan diwarnai dengan warna kedua yaitu diambil warna biru. Kemudian warnai *vertex* yang tidak berhubungan dengan V6, diberi warna yang sama yaitu warna kedua seperti pada gambar 4.16

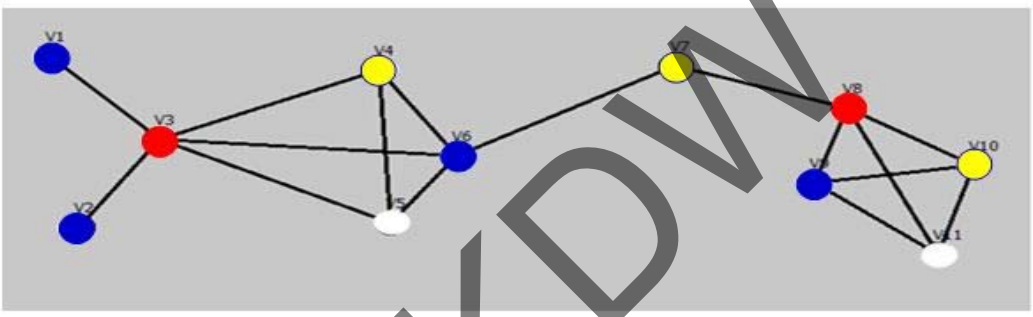


Gambar 4.16 Pewarnaan Welch-Powell dengan warna ke-2

Diperoleh V1, V2, V6 dan V9 adalah *vertex* yang diwarnai dengan warna kedua. Cek apakah masih ada *vertex* yang belum diwarnai. Ternyata masih ada sehingga pewarnaan dilanjutkan dengan warna ketiga.

Langkah 4:

Urutan *vertex* yang diwarnai berikutnya adalah V4 yang akan diwarnai dengan warna ketiga yaitu diambil warna kuning. Kemudian dicek apakah urutan *vertex* berikutnya bertetangga dengan V4 dan belum diberi warna. Kemudian beri warna sama dengan V4 yaitu dengan warna ketiga seperti pada gambar 4.17

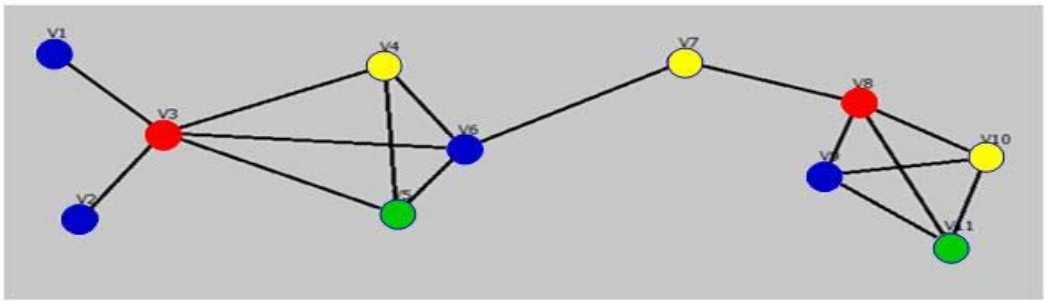


Gambar 4.17 Pewarnaan Welch-Powell dengan warna ke-3

Diperoleh V4, V7 dan V10 adalah *vertex* yang diwarnai dengan warna ketiga. Cek apakah masih ada *vertex* yang belum diwarnai. Ternyata masih ada sehingga pewarnaan dilanjutkan dengan warna keempat.

Langkah 5:

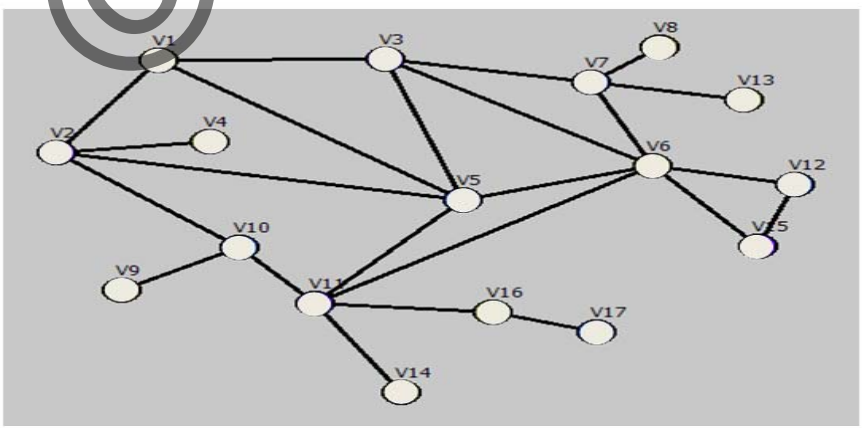
Urutan *vertex* yang diwarnai berikutnya adalah V5 yang akan diwarnai dengan warna keempat yaitu diambil warna hijau. Kemudian dicek apakah urutan *vertex* berikutnya berhubungan dengan V5 dan belum diberi warna. Kemudian beri warna sama dengan V5 yaitu dengan warna ketiga seperti pada gambar 4.18



Gambar 4.18 Pewarnaan Welch-Powell dengan warna ke-4

Diperoleh V5 dan V11 adalah *vertex* yang diwarnai dengan warna keempat. Cek apakah masih ada *vertex* yang belum diwarnai. Ternyata sudah tidak ada sehingga pewarnaan selesai.

Hasil akhir pewarnaan yang diperoleh dengan pewarnaan secara manual adalah:
Vertex V3 dan V8 diberi warna **merah**
Vertex V1, V2, V6 dan V9 diberi warna **biru**
Vertex V4, V7 dan V10 diberi warna **kuning**
Vertex V5 dan V11 diberi warna **hijau**
Hasil warna minimum atau bilangan Kromatik (K(G)) adalah **4.**
Akan diambil satu sampel contoh *graph* sembarang lagi untuk membandingkan pewarnaan manual dengan pewarnaan menggunakan algoritma Welch-Powell. *Graph* yang diambil sebagai sampel seperti pada gambar 4.19

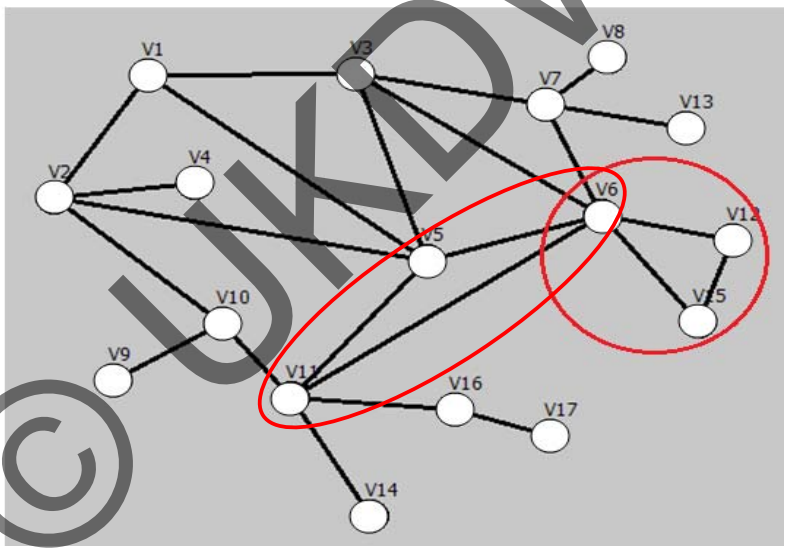


Gambar 4.19 *Graph* Sembarang untuk sampel

Derajat *vertex* pada *graph* gambar 4.18 adalah sebagai berikut:

V1 = 3	V7 = 4	V12 = 2	V17 = 1
V2 = 4	V8 = 1	V13 = 1	
V3 = 4	V9 = 1	V14 = 1	
V5 = 5	V10 = 3	V15 = 2	
V6 = 6	V11 = 5	V16 = 2	

Gambar 4.19 menunjukkan bahwa *graph* memiliki *graph* komplit K3 yang ditunjukkan pada gambar 4.20. sehingga pada bagian *graph* komplit minimal warna yang dibutuhkan adalah 3. Akan dicoba apakah hasil pewarnaan *graph* sembarang pada gambar 4.20 menghasilkan bilangan kromatik K(G) atau jumlah warna terkecil yang dibutuhkan = **3**



Gambar 4.20 Bagian *Graph* yang membentuk *Graph* komplit

Langkah 1:

Urutkan vertex berdasarkan derajat, dimulai dari derajat terbesar. Diperoleh hasil pengurutan adalah sebagai berikut:

(V 6)=6

(V 5)=5

(V 11)=5

(V 2)=4

(V 3)=4

(V 7)=4

(V 1)=3

(V 10)=3

(V 12)=2

(V 15)=2

(V 16)=2

(V 4)=1

(V 8)=1

(V 9)=1

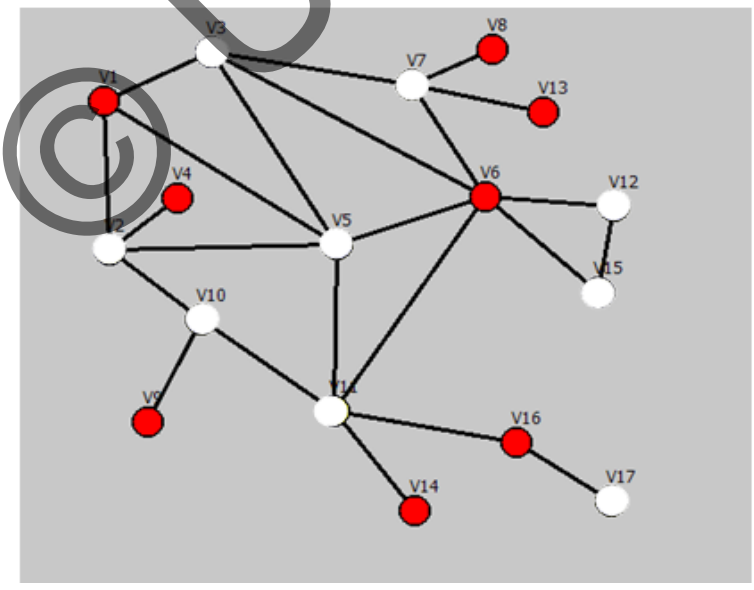
(V 13)=1

(V 14)=1

(V 17)=1

Langkah 2:

Vertex dengan derajat terbesar yaitu V6 diwarnai terlebih dahulu, diberi warna merah sebagai warna pertama. Kemudian warnai *vertex* yang tidak berhubungan dengan V3 dengan warna pertama, maka diperoleh hasil seperti pada gambar 4.21.

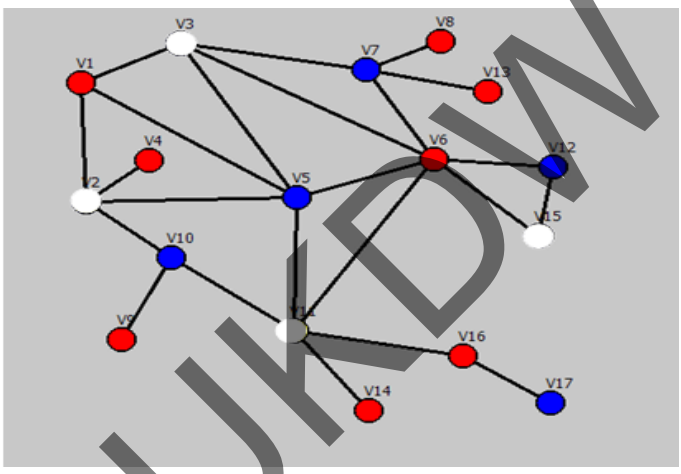


Gambar 4.21 Pewarnaan Algoritma Welch-Powell Warna ke-1

Diperoleh V1, V4, V6 , V8, V9, V13, V14 dan V16 adalah *vertex* yang diwarnai dengan warna pertama. Cek apakah masih ada *vertex* yang belum diwarnai. Ternyata masih ada sehingga pewarnaan dilanjutkan dengan warna ketiga.

Langkah 3:

Vertex dengan derajat terbesar berikut sesuai dengan urutan yang belum diwarnai yaitu V5. Dipilih warna biru sebagai warna kedua. Kemudian warnai dengan warna yang sama *vertex* yang tidak berhubungan dengan V5. Maka diperoleh hasil seperti pada gambar 4.22.

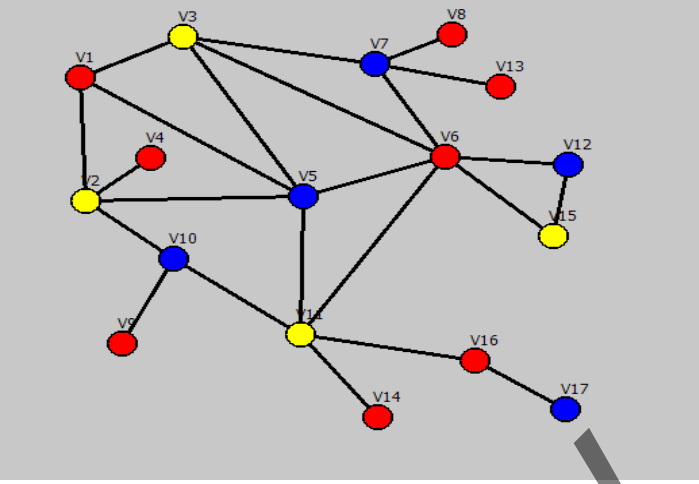


Gambar 4.22 Pewarnaan Algoritma Welch-Powell Warna ke-2

Vertex-vertex yang berwarna kedua adalah V5, V7, V10, V12 dan V17. Setelah cek apakah masih ada vertex yang belum diwarnai, lanjutkan pewarnaan terhadap *vertex* yang belum diberi warna.

Langkah 4:

Vertex dengan derajat terbesar berikut sesuai dengan urutan yang belum diwarnai yaitu V11. Dipilih warna kuning sebagai warna ketiga. Kemudian sesuai warnai *vertex* yang tidak berhubungan dengan V11 dengan warna ketiga. Maka diperoleh hasil seperti pada gambar 4.23.



Gambar 4.23 Pewarnaan Algoritma Welch-Powell Warna ke-3

Vertex-vertex yang berwarna kedua adalah V2, V3, V12 dan V15. Setelah cek apakah masih ada vertex yang belum diwarnai, ternyata semua *vertex* telah diwarnai, maka pewarnaan *graph* dengan algoritmaWelch-Powell telah selesai. Hasil akhir pewarnaan yang diperoleh dengan algoritma Welch-Powell adalah:

Vertex V1, V4, V6 , V8, V9, V13, V14 dan V16 diberi warna **merah**
Vertex V5, V7, V10, V12 dan V17 diberi warna **biru**
Vertex V2, V3, V12 dan V15 diberi warna **kuning**
Hasil warna minimum atau bilangan Kromatik (K(G)) adalah **3.**

Berikut adalah beberapa hasil penelitian yang sudah dilakukan dan dirangkum pada tabel 4.1 dimana hasil percobaan akan dilampirkan diakhir laporan.

Tabel 4.1 Rangkuman Hasil Penelitian

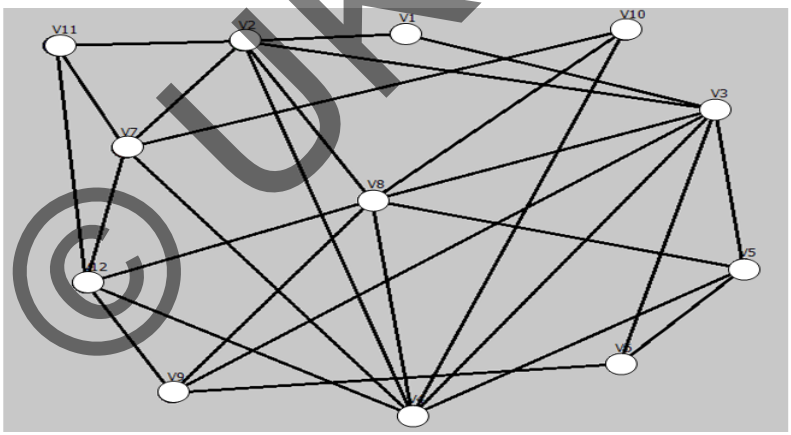
no	Jumlah *Vertex*	Jumlah *edge*	Graph Komplit yang ada dan warna minimal yang dibutuhkan	Jumlah pewarnaan welch-Powell	Welch-powell menjamin warna minimal
1	10	21	5	5	Ya
2	11	28	5	5	Ya
3	12	30	5	5	Ya
4	13	30	5	5	Ya
5	12	19	4	4	Ya
6	13	24	4	4	Ya
7	15	29	4	4	Ya
8	13	29	4	4	Ya
9	15	31	4	4	Ya
10	9	21	4	4	Ya

11	15	38	5	5	Ya
12	11	23	6	6	Ya
13	11	26	6	6	Ya
14	11	30	6	6	Ya
15	16	41	6	6	Ya
16	10	21	4	4	Ya
17	12	28	5	5	Ya
18	12	35	4	5	Tidak
19	12	37	4	4	Ya
20	11	25	5	5	Ya

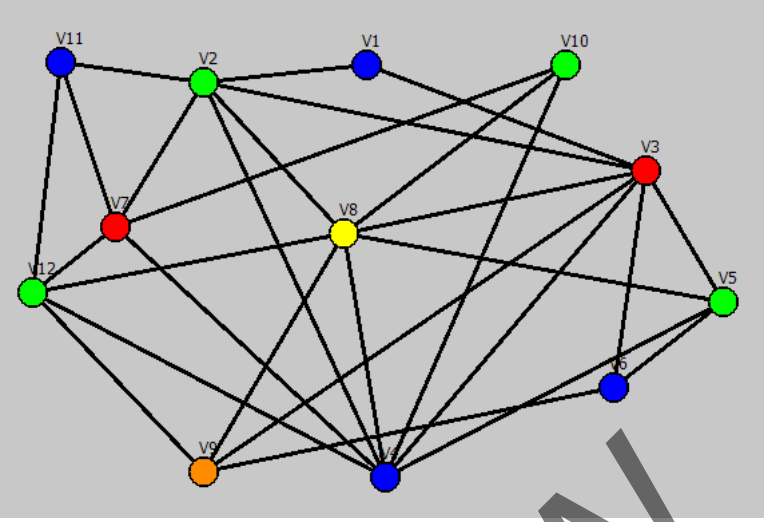
Dari tabel rangkuman hasil akan dicek persentase keakuratannya:

Persentase = ((20 – 1)/20)*100% = 95%.

Dari 20 kali percobaan, pada percobaan ke-18 dengan menggunakan algoritma Welch-Powel menghasilkan jumlah warna 5, sedangkan warna minimal yang dimungkinkan pada percobaan ke-18 adalah 4 warna. *Graph* sembarang dan hasil pewarnaan dengan algoritma Welch-Powell pada percobaan ke-18 dapat dilihat pada gambar 4.24 dan 4.25

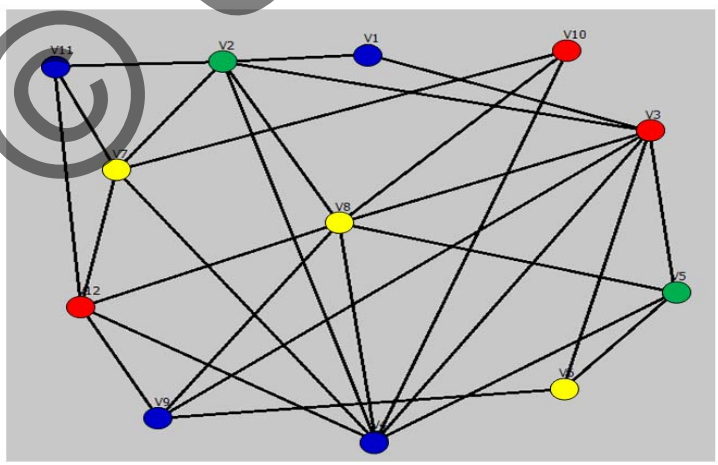


Gambar 2.24 *Graph* sembarang percobaan ke-18



Gambar 2.25 *Graph* sembarang percobaan ke-18 yang telah diwarnai dengan algoritma Welch-Powell

Graph sembarang yang ditunjukkan gambar 4.24 memiliki bagian *graph* yang membentuk *graph* komplit yaitu V3, V4, V5 dan V8 yang membentuk *graph* komplit K4. Gambar 2.25 menunjukkan hasil pewarnaan *graph* sembarang pada gambar 2.24 dengan menggunakan algoritma Welch-Powell. Hasil warna minimal yang diperoleh dengan menggunakan algoritma Welch-Powell adalah **5 warna**.



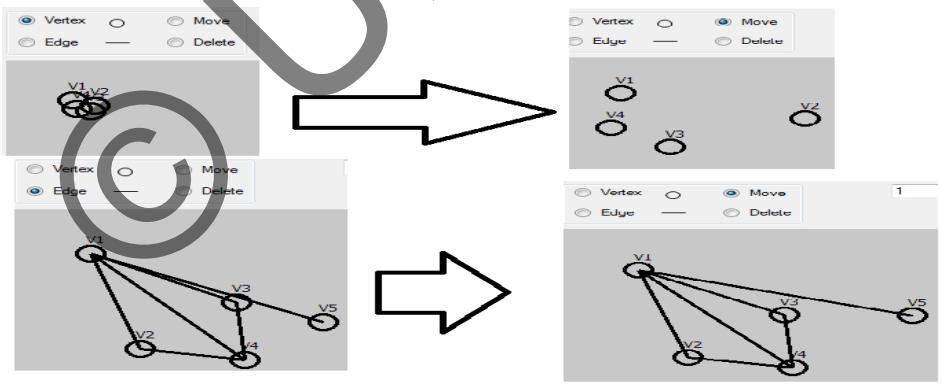
Gambar 2.26 *Graph* sembarang percobaan ke-18 yang telah diwarnai secara manual untuk mencari warna minimal

Gambar 2.26 menunjukkan bahwa pada *graph* sembarang percobaan ke-18 yang digambarkan pada gambar 2.24, warna minimal yang memungkinkan dan diperoleh pada gambar 2.26 adalah **4 warna**. Hal ini dikarenakan algoritma Welch-Powell tidak memperhitungkan dan mempertimbangkan langkah urutan pewarnaan berikutnya. Pada gambar 2.24, V7 dan V12 memiliki derajat yang sama yaitu 5. gambar 2.25 algoritma Welch-Powell memilih V7 yang diwarnai terlebih dahulu sedangkan pada gambar 2.26 *vertex* yang diwarnai terlebih dahulu adalah V12.

4.3 Evaluasi Program Simulasi Pewarnaan *Graph*

Kelebihan aplikasi ini adalah mengoptimalkan pewarnaan minimum pada *graph.* *Iput*-an *vertex* bisa digeser atau di-*drag* oleh *user* dan *edge* yang menghubungkan *vertex* yang digeser dengan *vertex* lainnya menyusuaikan kemana *vertex* tersebut digeser , sehingga *user* dengan mudah mampu mengatasi penggambaran *vertex* yang menumpuk seperti pada gambar 4.24.

Kelemahan aplikasi ini ialah tidak adanya animasi yang memadai dalam melakukan simulasi pewarnaan dan animasi untuk penjelasan langkah-langkahnya sehingga *interface*-nya kelihatan monoton.



Gambar 4.24 Proses Penggeseran *Vertex*

LAMPIRAN A

LISTING PROGRAM

```
Module Module1
    Public statusgambar As Boolean
    Public statusvertex As Boolean
    Public tabuvertex(1000) As Integer
    Public ntabu As Integer
    Public matrikadjency(1000, 1000) As Integer
    Public matriktetangga(1000, 1000) As Integer
    Public jmltetangga(1000) As Integer
    Public Structure tipevertex
        Dim x As Single
        Dim y As Single
        Dim derajat As Integer
        Dim r As Byte
        Dim g As Byte
        Dim b As Byte
        Dim statuswarna As Boolean
        Dim id As Integer
    End Structure
    Public Structure tipeedge
        Dim id1 As Integer
        Dim id2 As Integer
        Dim x1 As Single
        Dim y1 As Single
        Dim x2 As Single
        Dim y2 As Single
    End Structure
    Public Structure tipeundo
        Dim nvertex As Integer
        Dim varvertex() As tipevertex
        Dim nedge As Integer
        Dim varedge() As tipeedge
    End Structure
    Public varvertex(1000) As tipevertex
    Public nvertex As Integer
    Public varedge(1000) As tipeedge
    Public nedge As Integer
    Public varundo(1000) As tipeundo
    Public nundo As Integer
    Public cursorundo As Integer
End Module

================================================================================
Imports System
Imports System.Drawing.Graphics
Imports System.Drawing.Imaging
Imports System.Windows.Forms
Public Class Form1
    Dim g As System.Drawing.Graphics
    Dim pen1 As New System.Drawing.Pen(Color.Black, 3)
    Dim brush As SolidBrush = New SolidBrush(Color.Blue)
    Dim f As Font = New Font("Tahoma", 10)
    Dim f2 As Font = New Font("Tahoma", 15, FontStyle.Bold)
    Dim m_x1 As Integer
    Dim m_y1 As Integer
    Dim m_x2 As Integer
    Dim m_y2 As Integer
    Dim id1 As Integer
```

```
    Dim id2 As Integer
    Dim cek As Integer = 0
    Dim s As String
    Dim tampi1 As Integer ' untuk vertex mouse down
    Dim tampi2 As Integer 'untuk vertex mouse up

    Private Sub Vertex_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Vertex.CheckedChanged
        statusvertex = True
        statusgambar = True
    End Sub
    Private Sub Edge_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Edge.CheckedChanged
        statusvertex = False
        statusgambar = True
    End Sub

    Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
        formUTAMA.Show()
        Me.PictureBox1.Refresh()
        nvertex = 0
        nedge = 0
    End Sub
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Me.Refresh()
        cek = 0
        Label2.Text = "JUMLAH WARNA : " & cek
        g = PictureBox1.CreateGraphics
        nundo = 1
        Dim i As Integer

        ReDim varundo(nundo).varvertex(1000)
        ReDim varundo(nundo).varedge(1000)
        For i = 1 To nvertex
            varundo(nundo).varvertex(i) = varvertex(i)
        Next i
        varundo(nundo).nvertex = nvertex
        For i = 1 To nedge
            varundo(nundo).varedge(i) = varedge(i)
        Next i
        varundo(nundo).nedge = nedge
        'TextBox1.Text = "1"
        'TextBox2.Text = nundo.ToString
        cursorundo = nundo
        'TextBox3.Text = cursorundo.ToString

    End Sub
    Private Sub PictureBox1_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles PictureBox1.MouseDown
        m_x1 = e.X
        m_y1 = e.Y
        Dim ada As Boolean = False
        Dim i As Integer
        ' cari vertex yang dipilih
        For i = 1 To nvertex
```

```
            If m_x1 >= varvertex(i).x - 20 And m_x1 <= varvertex(i).x + 20
And m_y1 >= varvertex(i).y - 20 And m_y1 <= varvertex(i).y + 20 Then
                ada = True
                id1 = i
                m_x1 = varvertex(i).x + 10
                m_y1 = varvertex(i).y + 10
                tampi1 = i
                Exit For
            End If
        Next
        If ada = False And statusvertex = False Then
            m_x1 = 0
            m_y1 = 0
        End If
    End Sub
    Private Sub PictureBox1_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles PictureBox1.MouseUp
        If m_x1 = 0 And m_y1 = 0 And statusvertex = False Then
            Exit Sub
        End If
        Dim initial As String 'untuk nama vertex
        Dim initial2 As String 'untuk nama edge
        Dim i As Integer
        m_x2 = e.X
        m_y2 = e.Y
        Dim ada As Boolean = False

        ' untuk drag
        If RadioButton1.Checked And statusgambar = False Then

            ' update vertex bersesuaian
            varvertex(id1).x = e.X
            varvertex(id1).y = e.Y
            ' update edge bersesuaian
            For i = 1 To nedge
                If varedge(i).id1 = id1 Then
                    varedge(i).x1 = varvertex(id1).x + 10
                    varedge(i).y1 = varvertex(id1).y + 10
                End If
                If varedge(i).id2 = id1 Then
                    varedge(i).x2 = varvertex(id1).x + 10
                    varedge(i).y2 = varvertex(id1).y + 10
                End If
            Next
            ' reset picturebox1
            PictureBox1.Refresh()
            ' gambar ulang vertex
            For i = 1 To nvertex
                g.DrawEllipse(pen1, varvertex(i).x, varvertex(i).y, 20, 20)
                initial = ""
                initial += "V" + i.ToString
                g.DrawString(initial, f, Brushes.Black, varvertex(i).x + 5,
varvertex(i).y - 15)
            Next
            ' gambar ulang edge
            For i = 1 To nedge
```

```
                g.DrawLine(pen1, varedge(i).x1, varedge(i).y1, varedge(i).x2,
varedge(i).y2)
                initial2 = ""
                initial2 += "e" + i.ToString
                g.DrawString(initial2, f2, Brushes.Black, ((varedge(i).x1 +
varedge(i).x2) / 2), ((varedge(i).y1 + varedge(i).y2) / 2))
            Next

            nundo += 1

            ReDim varundo(nundo).varvertex(1000)
            ReDim varundo(nundo).varedge(1000)
            For i = 1 To nvertex
                varundo(nundo).varvertex(i) = varvertex(i)
            Next i

            varundo(nundo).nvertex = nvertex
            For i = 1 To nedge
                varundo(nundo).varedge(i) = varedge(i)
            Next i
            varundo(nundo).nedge = nedge
            ' TextBox1.Text = "1"
            'TextBox2.Text = nundo.ToString
            cursorundo = nundo
            'TextBox3.Text = cursorundo.ToString

            Exit Sub
        End If
        Dim j As Integer

        'untuk hapus vertex
        If Rdelete.Checked And statusgambar = False Then
            Dim tamp_vertex(100) As tipevertex
            Dim tamp_edge(100) As tipeedge
            Dim tamp As Integer

            ' cek jika yang di hapus id pertama
            If id1 = 1 Then
                ' update edge sekalian ubah vertexnya
                varvertex(id1).id = 0
                For i = 2 To nvertex
                    varvertex(i).id = i - 1
                    For j = 1 To nedge
                        If varedge(j).id1 = id1 Then
                            varedge(j).id1 = 0
                            varedge(j).id2 = 0
                        ElseIf varedge(j).id2 = id1 Then
                            varedge(j).id1 = 0
                            varedge(j).id2 = 0
                        Else ' update id1 dan id2 yang bukan dihapus
                            varedge(j).id1 = varedge(j).id1 - 1
                            varedge(j).id2 = varedge(j).id2 - 1
                        End If
                    Next
                Next
            ElseIf id1 = nvertex Then
                ' update edge sekalian ubah vertexnya
```

```
        varvertex(id1).id = 0
        For i = 1 To nvertex - 1
            For j = 1 To nedge
                If varedge(j).id1 = id1 Then
                    varedge(j).id1 = 0
                    varedge(j).id2 = 0
                ElseIf varedge(j).id2 = id1 Then
                    varedge(j).id1 = 0
                    varedge(j).id2 = 0
                End If
            Next
        Next
    Else ' jika posisi yang dihapus ditengah2
        ' dari nomor 1 sampe id1-1
        varvertex(id1).id = 0
        For i = 1 To id1 - 1
        Next
        ' untuk posisi id1 sampai nvertex '-1
        For i = id1 + 1 To nvertex
            varvertex(i).id = i - 1
        Next
        ' update edgenya
        For j = 1 To nedge
            If varedge(j).id1 = id1 Then
                varedge(j).id1 = 0
                varedge(j).id2 = 0
            ElseIf varedge(j).id2 = id1 Then
                varedge(j).id1 = 0
                varedge(j).id2 = 0
            Else
                If varedge(j).id1 > id1 Then
                    varedge(j).id1 = varedge(j).id1 - 1
                End If
                If varedge(j).id2 > id1 Then
                    varedge(j).id2 = varedge(j).id2 - 1
                End If

            End If
        Next
    End If
    initial = ""
    tamp = 0
    'hapus vertex
    For i = 1 To nvertex
        If varvertex(i).id <> 0 Then
            tamp += 1
            tamp_vertex(tamp).x = varvertex(i).x
            tamp_vertex(tamp).y = varvertex(i).y
            tamp_vertex(tamp).id = varvertex(i).id
        End If
    Next
    nvertex = tamp
    Dim tamp1 As Integer
    tamp1 = 0
    'hapus edge dari vertex
    For i = 1 To nedge
        If varedge(i).id1 <> 0 And varedge(i).id2 <> 0 Then
```

```
                    tamp1 += 1
                    tamp_edge(tamp1).id1 = varedge(i).id1
                    tamp_edge(tamp1).id2 = varedge(i).id2
                    tamp_edge(tamp1).x1 = varedge(i).x1
                    tamp_edge(tamp1).y1 = varedge(i).y1
                    tamp_edge(tamp1).x2 = varedge(i).x2
                    tamp_edge(tamp1).y2 = varedge(i).y2
                End If
            Next
            rtb.Text = tamp1.ToString
            nedge = tamp1
            tamp1 = 0
            For i = 1 To nedge
                varedge(i).id1 = tamp_edge(i).id1
                varedge(i).id2 = tamp_edge(i).id2

                varedge(i).x1 = tamp_edge(i).x1
                varedge(i).y1 = tamp_edge(i).y1
                varedge(i).x2 = tamp_edge(i).x2
                varedge(i).y2 = tamp_edge(i).y2
            Next
            ' UPDATE POSISI YANG SUDAH DIHAPUS
            rtb.Text = nedge.ToString
            PictureBox1.Refresh()
            'gambar ulang vertex
            For i = 1 To nvertex
                varvertex(i).id = tamp_vertex(i).id
                varvertex(i).x = tamp_vertex(i).x
                varvertex(i).y = tamp_vertex(i).y
                g.DrawEllipse(pen1, varvertex(i).x, varvertex(i).y, 20, 20)
                initial = ""
                initial += "V" + i.ToString
                g.DrawString(initial, f, Brushes.Black, varvertex(i).x + 5,
varvertex(i).y - 15)
            Next
            'gambar ulang edge
            For i = 1 To nedge
                g.DrawLine(pen1, varedge(i).x1, varedge(i).y1, varedge(i).x2,
varedge(i).y2)
                initial2 = ""
                initial2 += "e" + i.ToString
                g.DrawString(initial2, f2, Brushes.Black, ((varedge(i).x1 +
varedge(i).x2) / 2), ((varedge(i).y1 + varedge(i).y2) / 2))
            Next

            nundo += 1

            ReDim varundo(nundo).varvertex(1000)
            ReDim varundo(nundo).varedge(1000)
            For i = 1 To nvertex
                varundo(nundo).varvertex(i) = varvertex(i)
            Next i

            varundo(nundo).nvertex = nvertex
            For i = 1 To nedge
                varundo(nundo).varedge(i) = varedge(i)
            Next i
```

```
            varundo(nundo).nedge = nedge
            ' TextBox1.Text = "1"
            ' TextBox2.Text = nundo.ToString
            cursorundo = nundo
            ' TextBox3.Text = cursorundo.ToString

            Exit Sub
        End If

'=================================================================================
==================

        If statusvertex Then 'untuk gambar vertex
            nvertex += 1
            varvertex(nvertex).id = nvertex
            varvertex(nvertex).x = e.X
            varvertex(nvertex).y = e.Y
            g.DrawEllipse(pen1, e.X, e.Y, 20, 20)
            initial = ""
            initial += "V" + nvertex.ToString
            g.DrawString(initial, f, Brushes.Black, e.X + 5, e.Y - 15)
            Label4.Text = "JUMLAH VERTEX : " & nvertex.ToString

            nundo += 1

            ReDim varundo(nundo).varvertex(1000)
            ReDim varundo(nundo).varedge(1000)
            For i = 1 To nvertex
                varundo(nundo).varvertex(i) = varvertex(i)
            Next i

            varundo(nundo).nvertex = nvertex
            For i = 1 To nedge
                varundo(nundo).varedge(i) = varedge(i)
            Next i
            varundo(nundo).nedge = nedge
            'TextBox1.Text = "1"
            'TextBox2.Text = nundo.ToString
            cursorundo = nundo
            'TextBox3.Text = cursorundo.ToString
        Else
            ' cari vertex yang dipilih untuk gambar edge
            ada = False
            For i = 1 To nvertex
                If m_x2 >= varvertex(i).x - 20 And m_x2 <= varvertex(i).x +
20 And m_y2 >= varvertex(i).y - 20 And m_y2 <= varvertex(i).y + 20 Then
                    ada = True
                    id2 = i
                    If id1 = id2 Or id2 = id1 Then 'untuk ga bisa loop /
menunjuk dirinya sendiri
                        Exit Sub
                    End If
                    m_x2 = varvertex(i).x + 10
                    m_y2 = varvertex(i).y + 10
                    tampi2 = i
                    Exit For
                End If
```

```
            Next

            If ada = True Then
                ' cek
                ada = False
                For i = 1 To nedge
                    If (varedge(i).id1 = id1 And varedge(i).id2 = id2) Or
(varedge(i).id1 = id2 And varedge(i).id2 = id1) Then
                        ada = True
                        Exit For
                    End If
                Next

                If ada = False Then
                    nedge += 1
                    varedge(nedge).id1 = id1
                    varedge(nedge).id2 = id2
                    varedge(nedge).x1 = m_x1
                    varedge(nedge).y1 = m_y1
                    varedge(nedge).x2 = m_x2
                    varedge(nedge).y2 = m_y2
                    g.DrawLine(pen1, m_x1, m_y1, m_x2, m_y2)
                    rtb.Text += "e" & nedge.ToString + ": " + " V " &
tampi1.ToString + " dihubungkan dengan V " & tampi2.ToString + Chr(13)
                    Label3.Text = "JUMLAH EDGE : " & nedge.ToString
                    initial2 = ""
                    initial2 += "e" + i.ToString
                    g.DrawString(initial2, f2, Brushes.Black, (m_x1 + m_x2) /
2, ((m_y2 + m_y1) / 2) + 5)
                    nundo += 1

                    ReDim varundo(nundo).varvertex(1000)
                    ReDim varundo(nundo).varedge(1000)
                    For i = 1 To nvertex
                        varundo(nundo).varvertex(i) = varvertex(i)
                    Next i

                    varundo(nundo).nvertex = nvertex
                    For i = 1 To nedge
                        varundo(nundo).varedge(i) = varedge(i)
                    Next i
                    varundo(nundo).nedge = nedge
                    'TextBox1.Text = "1"
                    'TextBox2.Text = nundo.ToString
                    cursorundo = nundo
                    'TextBox3.Text = cursorundo.ToString
                End If

            End If
            End If
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        ' hitung derajat vertex
        Dim i As Integer
        Dim j As Integer
```

```
        Dim selesai As Boolean



        Dim tampung As Integer
        Dim tamp_urut As Integer
        Dim tampung2(100) As Integer
        Dim urut(100) As Integer

        For i = 1 To nvertex
            ' hitung vertex ke-i punya berapa edge
            varvertex(i).derajat = 0
            tampung2(i) = varvertex(i).derajat
            If nedge = 0 Then
                varvertex(i).derajat = 1
            End If
            For j = 1 To nedge
                ' cari dari id1 dari edge
                If varedge(j).id1 = i Then
                    varvertex(i).derajat += 1
                    tampung2(i) = varvertex(i).derajat
                End If
                ' cari dari id2 dari edge
                If varedge(j).id2 = i Then
                    varvertex(i).derajat += 1
                    tampung2(i) = varvertex(i).derajat
                End If
            Next j
        Next i

        For i = 1 To nvertex
            urut(i) = i
        Next
        'sorting pake exchangesort
        For i = 1 To nvertex
            For j = (i + 1) To nvertex
                If tampung2(i) < tampung2(j) Or urut(i) > urut(j) Then
                    tampung = tampung2(i)
                    tamp_urut = urut(i)
                    tampung2(i) = tampung2(j)
                    urut(i) = urut(j)
                    tampung2(j) = tampung
                    urut(j) = tamp_urut
                End If
            Next
        Next
        s = ""
        s += ("URUTAN VERTEX BERDASARKAN DERAJAT") + Chr(13)
        For i = 1 To nvertex
            s += ("(V " + urut(i).ToString + " )")
            s += ("=" + tampung2(i).ToString) + Chr(13)
        Next
        s += ("================================================") +
Chr(13)


        selesai = False
```

```
        Dim max As Single
        Dim idmax As Integer
        Dim r As Byte
        Dim gr As Byte
        Dim b As Byte
        Dim adaedge As Boolean
        Dim selesai2 As Boolean
        Dim jumlahcari As Integer



        Dim warna As Integer

        'Mulai Welch-Powell
        While selesai = False
            ' cari max derajat dari vertex yang belum diwarnai
            cek += 1
            max = 0
            idmax = 0
            For i = 1 To nvertex
                If varvertex(i).statuswarna = False And varvertex(i).derajat
> max Then
                    max = varvertex(i).derajat
                    idmax = i
                End If
            Next

            If idmax = 0 Then 'sudah ga ada lagi
                selesai = True
            End If

            warna = Val(cek)
            If selesai = False Then
                s += ("derajat vertex paling besar ke- " & cek) + (" adalah V
" & idmax.ToString)
                s += " dengan jumlah derajat= " & max.ToString
                rtb2.Text = s
                s += Chr(13)
                ntabu = 0
                ' warnai idmax dengan warna sesuai random color
                MsgBox(("vertex yang diwarnai warna ke-" & cek) + " adalah "
+ ("V " & idmax.ToString))
                s += ("(1)vertex yang diwarnai warna ke- " & cek) + " adalah
" + ("V " & idmax.ToString)
                s += Chr(13)

                If warna = 1 Then
                    varvertex(idmax).r = 255
                    varvertex(idmax).g = 0
                    varvertex(idmax).b = 0
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                ElseIf warna = 2 Then
```

```
                    varvertex(idmax).r = 0
                    varvertex(idmax).g = 0
                    varvertex(idmax).b = 255
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                ElseIf warna = 3 Then
                    varvertex(idmax).r = 255
                    varvertex(idmax).g = 255
                    varvertex(idmax).b = 0
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                ElseIf warna = 4 Then
                    varvertex(idmax).r = 0
                    varvertex(idmax).g = 255
                    varvertex(idmax).b = 0
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                ElseIf warna = 5 Then
                    varvertex(idmax).r = 255
                    varvertex(idmax).g = 140
                    varvertex(idmax).b = 0
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                ElseIf warna = 6 Then
                    varvertex(idmax).r = 255
                    varvertex(idmax).g = 215
                    varvertex(idmax).b = 0
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                ElseIf warna = 7 Then
                    varvertex(idmax).r = 154
                    varvertex(idmax).g = 205
                    varvertex(idmax).b = 50
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                ElseIf warna = 8 Then
                    varvertex(idmax).r = 0
                    varvertex(idmax).g = 250
                    varvertex(idmax).b = 154
```

```
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                ElseIf warna = 9 Then
                    varvertex(idmax).r = 148
                    varvertex(idmax).g = 0
                    varvertex(idmax).b = 211
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                ElseIf warna = 10 Then
                    varvertex(idmax).r = 255
                    varvertex(idmax).g = 20
                    varvertex(idmax).b = 147
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                ElseIf warna = 11 Then
                    varvertex(idmax).r = 30
                    varvertex(idmax).g = 144
                    varvertex(idmax).b = 255
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                ElseIf warna = 12 Then
                    varvertex(idmax).r = 184
                    varvertex(idmax).g = 134
                    varvertex(idmax).b = 11
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                ElseIf warna = 13 Then
                    varvertex(idmax).r = 139
                    varvertex(idmax).g = 69
                    varvertex(idmax).b = 19
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                ElseIf warna = 14 Then
                    varvertex(idmax).r = 255
                    varvertex(idmax).g = 127
                    varvertex(idmax).b = 80
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
```

```
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                Else
                    ' warnai idmax dengan warna sesuai random color
                    r = Int(Rnd() * 256)
                    gr = Int(Rnd() * 256)
                    b = Int(Rnd() * 256)
                    varvertex(idmax).r = r
                    varvertex(idmax).g = gr
                    varvertex(idmax).b = b
                    brush.Color = Color.FromArgb(varvertex(idmax).r,
varvertex(idmax).g, varvertex(idmax).b)
                    g.FillEllipse(brush, varvertex(idmax).x,
varvertex(idmax).y, 20, 20)
                    Delay(1)
                End If
                varvertex(idmax).statuswarna = True
                Dim max2 As Single

                ' cari yang tetangganya masukin ke tabulist
                For i = 1 To nvertex
                    If i <> idmax Then
                        adaedge = False
                        For j = 1 To nedge
                            ' cek dari id1
                            If varedge(j).id1 = idmax And varedge(j).id2 = i
Then
                                adaedge = True
                                ntabu += 1
                                tabuvertex(ntabu) = i
                            End If
                            ' cek dari id2
                            If varedge(j).id2 = idmax And varedge(j).id1 = i
Then
                                adaedge = True
                                ntabu += 1
                                tabuvertex(ntabu) = i
                            End If
                        Next
                    End If
                Next

                'tampung max yang ada
                For i = 1 To nvertex
                    For j = i To nvertex
                        If varvertex(i).statuswarna = False And
varvertex(i).derajat < varvertex(j).derajat Then
                            max2 = varvertex(j).derajat
                        End If
                    Next
                Next

                ' cari yang bukan tetangganya
                For i = 1 To nvertex
                    If i <> idmax And varvertex(i).statuswarna = False And
varvertex(i).derajat >= max2 Then
```

```
                        adaedge = False
                        For j = 1 To nedge
                            ' cek dari id1
                            If varedge(j).id1 = idmax And varedge(j).id2 = i
Then
                                adaedge = True
                            End If
                            ' cek dari id2
                            If varedge(j).id2 = idmax And varedge(j).id1 = i
Then
                                adaedge = True
                            End If
                        Next
                        If adaedge = False Then 'ketemu vertex yg bukan
tetangga idmax ,warnai sama dengan idmax
                            MsgBox(("vertex yang diwarnai sama dengan warna
ke-" & cek) + " berikutnya adalah " + ("V " & i.ToString))
                            If warna = 1 Then
                                varvertex(i).r = 255
                                varvertex(i).g = 0
                                varvertex(i).b = 0
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 2 Then
                                varvertex(i).r = 0
                                varvertex(i).g = 0
                                varvertex(i).b = 255
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 3 Then
                                varvertex(i).r = 255
                                varvertex(i).g = 255
                                varvertex(i).b = 0
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 4 Then
                                varvertex(i).r = 0
                                varvertex(i).g = 255
                                varvertex(i).b = 0
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 5 Then
                                varvertex(i).r = 255
                                varvertex(i).g = 140
                                varvertex(i).b = 0
```

```
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 6 Then
                                varvertex(i).r = 255
                                varvertex(i).g = 215
                                varvertex(i).b = 0
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 7 Then
                                varvertex(i).r = 154
                                varvertex(i).g = 205
                                varvertex(i).b = 50
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 8 Then
                                varvertex(i).r = 0
                                varvertex(i).g = 250
                                varvertex(i).b = 154
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 9 Then
                                varvertex(i).r = 148
                                varvertex(i).g = 0
                                varvertex(i).b = 211
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 10 Then
                                varvertex(i).r = 255
                                varvertex(i).g = 20
                                varvertex(i).b = 147
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 11 Then
                                varvertex(i).r = 30
                                varvertex(i).g = 144
                                varvertex(i).b = 255
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
```

```
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 12 Then
                                varvertex(i).r = 184
                                varvertex(i).g = 134
                                varvertex(i).b = 11
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 13 Then
                                varvertex(i).r = 139
                                varvertex(i).g = 69
                                varvertex(i).b = 19
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 14 Then
                                varvertex(i).r = 255
                                varvertex(i).g = 127
                                varvertex(i).b = 80
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            Else
                                ' warnai idmax dengan warna sesuai random
color
                                r = Int(Rnd() * 256)
                                gr = Int(Rnd() * 256)
                                b = Int(Rnd() * 256)
                                varvertex(idmax).r = r
                                varvertex(idmax).g = gr
                                varvertex(idmax).b = b
                                brush.Color =
Color.FromArgb(varvertex(idmax).r, varvertex(idmax).g, varvertex(idmax).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            End If
                            varvertex(i).statuswarna = True
                            s += ("(2)vertex yang diwarnai warna ke- " & cek)
+ " adalah " + ("V " & i.ToString)
                            s += Chr(13)
                            Exit For
                        End If
                    End If
                Next

                ' cari yang bukan tetangga dari bukan tetangganya
                selesai2 = False
                idmax = i
```

```
                Dim adatabu As Boolean
                While selesai2 = False

                    For i = 1 To nvertex
                        If i <> idmax Then
                            adaedge = False
                            For j = 1 To nedge
                                ' cek dari id1
                                If varedge(j).id1 = idmax And varedge(j).id2
= i Then

                                    adaedge = True
                                    ntabu += 1
                                    tabuvertex(ntabu) = i
                                End If
                                ' cek dari id2
                                If varedge(j).id2 = idmax And varedge(j).id1
= i Then

                                    adaedge = True
                                    ntabu += 1
                                    tabuvertex(ntabu) = i
                                End If
                            Next
                        End If
                    Next


                    ' cari yang bukan tetangganya
                    jumlahcari = 0
                    For i = 1 To nvertex
                        ' cari ada di tabu g
                        adatabu = False
                        For j = 1 To ntabu
                            If i = tabuvertex(j) Then
                                adatabu = True
                                Exit For
                            End If
                        Next

                        If i <> idmax And varvertex(i).statuswarna = False
And adatabu = False Then

                            adaedge = False
                            For j = 1 To nedge
                                ' cek dari id1
                                If varedge(j).id1 = idmax And varedge(j).id2
= i Then

                                    adaedge = True
                                    ntabu += 1
                                    tabuvertex(ntabu) = i
                                End If
                                ' cek dari id2
                                If varedge(j).id2 = idmax And varedge(j).id1
= i Then

                                    adaedge = True
                                    ntabu += 1
                                    tabuvertex(ntabu) = i
                                End If
                            Next
```

```
                        If adaedge = False Then 'ketemu vertex yg bukan
tetangga idmax ,warnai sama dengan idmax
                            MsgBox(("vertex yang diwarnai sama dengan
warna ke-" & cek) + " berikutnya adalah " + ("V " & i.ToString))
                            If warna = 1 Then
                                varvertex(i).r = 255
                                varvertex(i).g = 0
                                varvertex(i).b = 0  'merah
                                brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 2 Then
                                varvertex(i).r = 0
                                varvertex(i).g = 0
                                varvertex(i).b = 255 'biru
                                brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 3 Then
                                varvertex(i).r = 255
                                varvertex(i).g = 255
                                varvertex(i).b = 0 'kuning
                                brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 4 Then
                                varvertex(i).r = 0
                                varvertex(i).g = 255
                                varvertex(i).b = 0 ' hijau
                                brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 5 Then
                                varvertex(i).r = 255
                                varvertex(i).g = 140
                                varvertex(i).b = 0 'orange
                                brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
                                g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                Delay(1)
                            ElseIf warna = 6 Then
                                varvertex(i).r = 255
                                varvertex(i).g = 215 'gold
                                varvertex(i).b = 0
                                brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
```

```
                                    g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                    Delay(1)
                                ElseIf warna = 7 Then
                                    varvertex(i).r = 154
                                    varvertex(i).g = 205
                                    varvertex(i).b = 50 'yellowgreen
                                    brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
                                    g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                    Delay(1)
                                ElseIf warna = 8 Then
                                    varvertex(i).r = 0
                                    varvertex(i).g = 250
                                    varvertex(i).b = 154 'mediumspringgreen
                                    brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
                                    g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                    Delay(1)
                                ElseIf warna = 9 Then
                                    varvertex(i).r = 148
                                    varvertex(i).g = 0
                                    varvertex(i).b = 211 'darkviolet
                                    brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
                                    g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                    Delay(1)
                                ElseIf warna = 10 Then
                                    varvertex(i).r = 255
                                    varvertex(i).g = 20
                                    varvertex(i).b = 147 'deeppink
                                    brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
                                    g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                    Delay(1)
                                ElseIf warna = 11 Then
                                    varvertex(i).r = 30
                                    varvertex(i).g = 144
                                    varvertex(i).b = 255 'dodgerblue
                                    brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
                                    g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                    Delay(1)
                                ElseIf warna = 12 Then
                                    varvertex(i).r = 184
                                    varvertex(i).g = 134
                                    varvertex(i).b = 11 'darkgoldenrod
                                    brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
                                    g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                    Delay(1)
```

```
                                ElseIf warna = 13 Then
                                    varvertex(i).r = 139
                                    varvertex(i).g = 69
                                    varvertex(i).b = 19 'brown
                                    brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
                                    g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                    Delay(1)
                                ElseIf warna = 14 Then
                                    varvertex(i).r = 255
                                    varvertex(i).g = 127
                                    varvertex(i).b = 80 'coral
                                    brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
                                    g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                    Delay(1)
                                Else
                                    ' warnai idmax dengan warna sesuai random
color
                                    r = Int(Rnd() * 256)
                                    gr = Int(Rnd() * 256)
                                    b = Int(Rnd() * 256)
                                    varvertex(idmax).r = r
                                    varvertex(idmax).g = gr
                                    varvertex(idmax).b = b
                                    brush.Color =
Color.FromArgb(varvertex(i).r, varvertex(i).g, varvertex(i).b)
                                    g.FillEllipse(brush, varvertex(i).x,
varvertex(i).y, 20, 20)
                                    Delay(1)
                                End If
                                varvertex(i).statuswarna = True

                                s += ("(3)vertex yang diwarnai warna ke- " &
cek) + " adalah " + ("V " & i.ToString)
                                s += Chr(13)

                                idmax = i
                                jumlahcari += 1
                                Exit For
                            End If
                        End If
                    Next
                    If jumlahcari = 0 Then
                        selesai2 = True
                    End If
                End While
            End If
                    s += Chr(13)
        End While

        'ngeset ulang ketika tombol welch-powell dijalankan lagi

        For i = 1 To nvertex
            varvertex(i).statuswarna = False
```

```
        Next

        MessageBox.Show("PEWARNAAN SELESAI", "Pewarnaan",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        cek = cek - 1
        Label2.Text = "JUMLAH WARNA : " & cek
        cek = 0

'=================================================================================
================================================================'
        rtb2.Text = s
    End Sub
    Public Sub Delay(ByVal Detik As Integer)
        Dim t As Double = Microsoft.VisualBasic.Timer
        Do Until (Microsoft.VisualBasic.Timer - t) >= Detik
            Application.DoEvents()
        Loop
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        ' cari matrik keterhubungan
        Dim i As Integer
        Dim j As Integer
        Dim k As Integer
        For i = 1 To nvertex
            jmltetangga(i) = 0
            For k = 1 To nvertex
                For j = 1 To nedge
                    ' cek dari id1
                    If varedge(j).id1 = i And varedge(j).id2 = k Then
                        jmltetangga(i) += 1
                        matriktetangga(i, jmltetangga(i)) = k
                    End If
                    ' cek dari id2
                    If varedge(j).id2 = i And varedge(j).id1 = k Then
                        jmltetangga(i) += 1
                        matriktetangga(i, jmltetangga(i)) = k
                    End If
                Next j
            Next k
        Next i

        ' tampilin matrik keterbuhungan

        For i = 1 To nvertex
            rtb1.Text += "V" + i.ToString + " bertetangga dengan: "
            For j = 1 To jmltetangga(i)
                rtb1.Text += "V" + matriktetangga(i, j).ToString + " , "
            Next
            rtb1.Text += Chr(13)
            rtb1.Text += Chr(13)
        Next

    End Sub
```

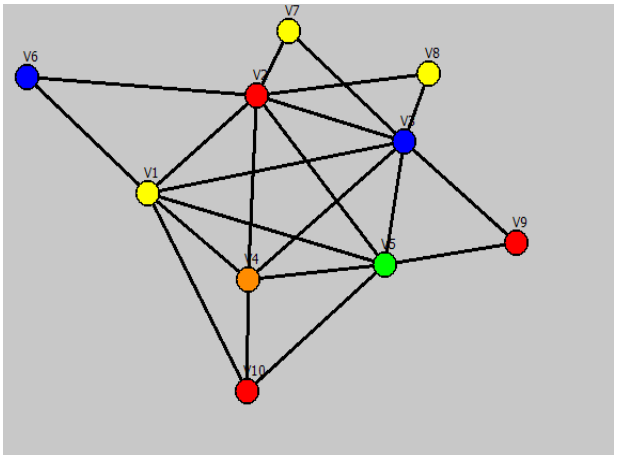
```
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        cursorundo = 0
        nundo = 0
        PictureBox1.Refresh()
        nvertex = 0
        nedge = 0
        cek = 0
        Label2.Text = "JUMLAH WARNA : " & cek
        Label3.Text = "JUMLAH EDGE : " & nedge
        Label4.Text = "JUMLAH VERTEX : " & nvertex
        rtb.Text = ""
        rtb1.Text = ""
        rtb2.Text = ""
    End Sub

    Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged
        statusgambar = False

    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click

        Dim initial As String
        Dim i As Integer
        cursorundo = cursorundo - 1
        If cursorundo <= 0 Then
            cursorundo = 1
        End If
        'TextBox3.Text = cursorundo.ToString

        ' update varvertex dan varedge sesuai cursor undo
        nvertex = varundo(cursorundo).nvertex
        For i = 1 To nvertex
            varvertex(i) = varundo(cursorundo).varvertex(i)
        Next i

        nedge = varundo(cursorundo).nedge
        For i = 1 To nedge

            varedge(i) = varundo(cursorundo).varedge(i)
        Next i

        PictureBox1.Refresh()
        ' gambar ulang vertex
        For i = 1 To nvertex
            g.DrawEllipse(pen1, varvertex(i).x, varvertex(i).y, 20, 20)
            initial = ""
            initial += "V" + i.ToString
            g.DrawString(initial, f, Brushes.Black, varvertex(i).x + 5,
varvertex(i).y - 15)
        Next
        ' gambar ulang edge
        For i = 1 To nedge
```

```
            g.DrawLine(pen1, varedge(i).x1, varedge(i).y1, varedge(i).x2,
varedge(i).y2)
        Next

    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
        Dim initial As String
        Dim i As Integer

        cursorundo = cursorundo + 1
        If cursorundo > nundo Then
            cursorundo = nundo
        End If
        ' TextBox2.Text = nundo.ToString
        'TextBox3.Text = cursorundo.ToString

        ' update varvertex dan varedge sesuai cursor undo
        ' update varvertex dan varedge sesuai cursor undo
        nvertex = varundo(cursorundo).nvertex
        For i = 1 To nvertex
            varvertex(i) = varundo(cursorundo).varvertex(i)
        Next i

        nedge = varundo(cursorundo).nedge
        For i = 1 To nedge

            varedge(i) = varundo(cursorundo).varedge(i)
        Next i

        PictureBox1.Refresh()
        ' gambar ulang vertex
        For i = 1 To nvertex
            g.DrawEllipse(pen1, varvertex(i).x, varvertex(i).y, 20, 20)
            initial = ""
            initial += "V" + i.ToString
            g.DrawString(initial, f, Brushes.Black, varvertex(i).x + 5,
varvertex(i).y - 15)
        Next
        ' gambar ulang edge
        For i = 1 To nedge
            g.DrawLine(pen1, varedge(i).x1, varedge(i).y1, varedge(i).x2,
varedge(i).y2)
        Next
    End Sub
    Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ExitToolStripMenuItem.Click
        Me.Close()
        formUTAMA.Show()
    End Sub
    Private Sub NewToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles NewToolStripMenuItem.Click
        PictureBox1.Refresh()
        nvertex = 0
        nedge = 0
        cursorundo = 1
```

```
        nundo = 1
    End Sub
    Private Sub Form1_Resize(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Resize
        Dim initial As String
        Dim i As Integer

        PictureBox1.Refresh()
        ' gambar ulang vertex
        For i = 1 To nvertex
            g.DrawEllipse(pen1, varvertex(i).x, varvertex(i).y, 20, 20)
            initial = ""
            initial += "V" + i.ToString
            g.DrawString(initial, f, Brushes.Black, varvertex(i).x + 5,
varvertex(i).y - 15)
        Next
        ' gambar ulang edge
        For i = 1 To nedge
            g.DrawLine(pen1, varedge(i).x1, varedge(i).y1, varedge(i).x2,
varedge(i).y2)
        Next
        Exit Sub
    End Sub
    Private Sub Rdelete_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Rdelete.CheckedChanged
        statusgambar = False
    End Sub

    Private Sub CaraMenggambarGraphToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CaraMenggambarGraphToolStripMenuItem.Click
        Help1.Show()
    End Sub
    Private Sub AlgortimaWelchPowellToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
AlgortimaWelchPowellToolStripMenuItem.Click
        help2.Show()
    End Sub


End Class
```

LAMPIRAN B

ANALISIS PENELITIAN PERCOBAAN PEWARNAAN Graph



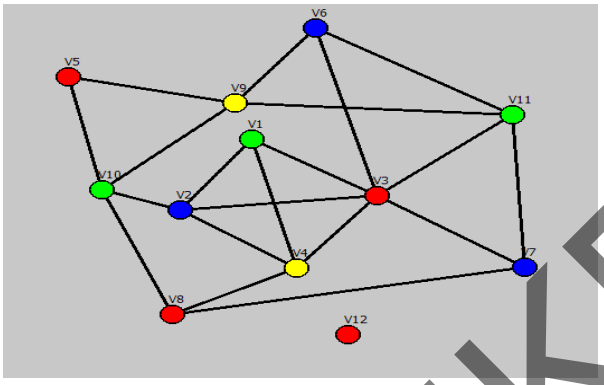
Percobaan 1

Hasil pengamatan :

1. Ada 10 vertex dan 21 edge.
2. Graph komplit (Kn) yang menjadi bagian dari graph sembarang adalah **K4**
3. Hasil pewarnaan dengan Welch-Powell adalah **4**

Kesimpulan :

Welch-Powell sesuai



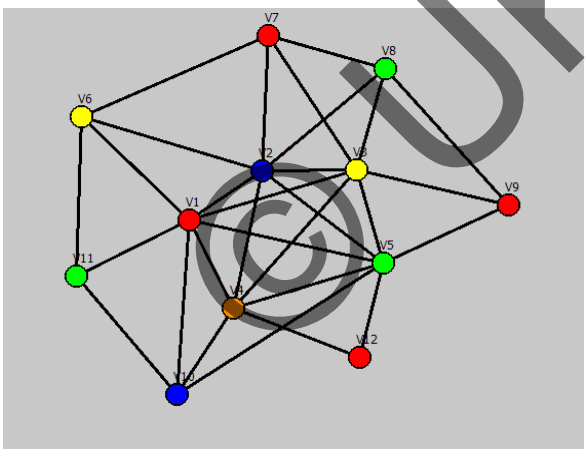
Percobaan 2

Hasil pengamatan :

1. Ada 10 vertex dan 21 edge.
2. Graph komplit (Kn) yang menjadi bagian dari graph sembarang adalah **K4**
3. Hasil pewarnaan dengan Welch-Powell adalah **4**

Kesimpulan :

Welch-Powell sesuai



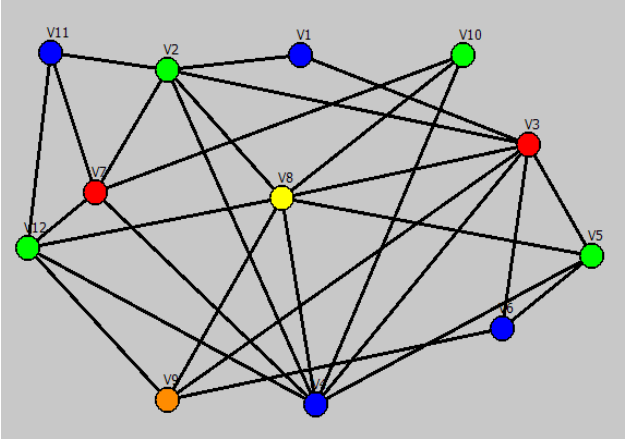
Percobaan 3

hasil pengamatan :

1. Ada 12 vertex dan 30 edge
2. Graph komplit (Kn) yang ada adalah **K5**
3. Pewarnaan menggunakan algoritma Welch-Powell adalah 5

Kesimpulan:

Welch-Powell sesuai



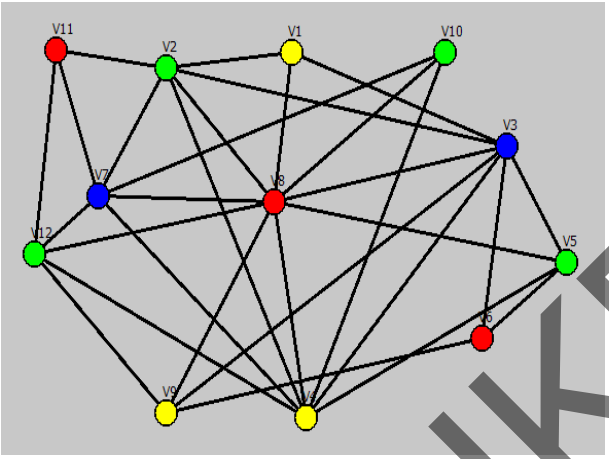
Percobaan 18

Hasil pengamatan :

1. Ada 12 vertex dan 35 edge.
2. Graph komplit (Kn) yang menjadi bagian dari graph sembarang adalah **K4**
3. Hasil pewarnaan dengan Welch-Powell menghasilkan bilangan Kromatik K(G) adalah **5.**

Kesimpulan:

Welch-Powell tidak menjamin solusi maksimum



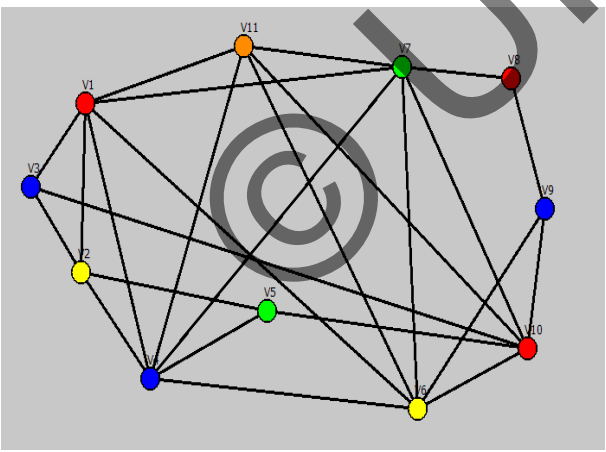
Percobaan 19

Hasil pengamatan

1. Ada 12 vertex dan 37 edge.
2. Graph komplit (Kn) yang menjadi bagian dari graph sembarang adalah **K4**
3. Hasil pewarnaan dengan Welch-Powell menghasilkan bilangan Kromatik K(G) adalah **4.**

Kesimpulan:

Welch-Powell sesuai.



Percobaan 20

Hasil pengamatan

1. Ada 11 vertex dan 25 edge.
2. Graph komplit (Kn) yang menjadi bagian dari graph sembarang adalah **K5**
3. Hasil pewarnaan dengan Welch-Powell menghasilkan bilangan Kromatik K(G) adalah **5.**

Kesimpulan:

Welch-Powell sesuai.

Fakultas Teknologi Informasi Program Studi Teknik Informatika
Universitas Kristen Duta Wacana - Yogyakarta

FORMULIR CATATAN REVISI SKRIPSI

Dicetak tanggal: 16-05-2012 13:39:47
(Diisi oleh Ketua Tim Penguji)

Pada hari ini : Jumat, 18 Mei 2012

Setelah dilakukan Ujian Skripsi maka dengan ini Ketua Tim Penguji Skripsi menyatakan bahwa mahasiswa tersebut dibawah ini:

Nama Mahasiswa	:	**LEONARD RINALDY**
No. Induk Mahasiswa	:	**22064173**
Judul Skripsi	:	SIMULASI ALGORITMA WELCH-POWELL UNTUK MELAKUKAN PEWARNAAN GRAPH.
Dosen Pembimbing I	:	**Drs. R GUNAWAN SANTOSA, M.Si.**
Dosen Pembimbing II	:	**ANTONIUS RACHMAT C, SKom.,M.Cs**

Memiliki beberapa perubahan/catatan yang harus dilakukan oleh mahasiswa tersebut diatas terkait dengan skripsi yang dikerjakannya:

NO.	CATATAN PERBAIKAN
1	Diberi informasi tentang edge yg berupa pasangan verteks.
2	flowchart ≥ Mohon dibenarkan!
3	Ketidak mampuan Welch Powell perlu diterangkan.
4	
5	
6	
7	
8	
9	
10	

Perubahan diatas harus sudah diselesaikan paling lambat tanggal : **Senin, 18 Juni 2012**

Yogyakarta, 18 Mei 2012
Ketua Tim Penguji

Drs. R. Gunawan Santosa MSi.

Catatan:

- 1 (satu) lembar untuk mahasiswa
- 1 (satu) lembar untuk arsip

Universitas Kristen Duta Wacana
Fakultas Teknologi Informasi Prodi Teknik Informatika
Jl. Dr. Wahidin Sudirohusodo 5 - 25 Yogyakarta 55224
Telp. : (0274) 563929 Faks : (0274) 513235

FORMULIR PERBAIKAN (REVISI) TUGAS AKHIR

Dengan ini kami menyatakan bahwa mahasiswa yang melakukan tugas akhir dibawah ini :

Nama Mahasiswa	: Leonard Rinaldy Mahakam Putra Ginting
NIM	: 22064173
Judul Tugas Akhir	: Simulasi Algoritma Welch-Powell untuk Melakukan Pewarnaan Graph.
Tgl. Pendadaran	: 18 Mei 2012
Tgl. Revisi	: 23 Mei 2012

Telah melakukan perbaikan tugas akhir dengan lengkap.
Demikian pernyataan kami agar dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 23 Mei 2012

Dosen Pembimbing I	Dosen Pembimbing II
(Drs. R. Gunawan Santosa, M.Si)	(Antonius Rachmat C, S.Kom., M.Cs.)

Program Studi Teknik Informatika
Fakultas Teknik - Universitas Kristen Duta Wacana

KARTU UJIAN TUGAS AKHIR / SKRIPSI

NIM 22064173

Nama LEONARD RINALDY GINTING

Judul Skripsi SIMULASI ALGORITMA WELCH-POWELL UNTUK MELAKUKAN PEWARNAAN GRAPH

Yogyakarta,

Koordinator Tugas Akhir

Dosen Pembimbing Tugas Akhir / Skripsi
Dosen Pembimbing I : DRS. R. GUNAWAN SANTOSA, M.Si
Dosen Pembimbing II: ANTONIUS RACHMAT C, S.Kom, M.Cs.

Ujian ke	Tanggal	Keterangan	Paraf Ketua Tim
1	8-05-2012	Lulus/~~Tidak Lulus~~	
2		Lulus/Tidak Lulus	
3		Lulus/Tidak Lulus	

Catatan: Kartu Ujian ini dikembalikan kepada Mahasiswa yang bersangkutan

UNIVERSITAS KRISTEN DUTA WACANA
FAKULTAS TEKNOLOGI INFORMASI
Program Studi Teknik Informatika

KARTU KONSULTASI SKRIPSI

LEMBAR INI DIISI OLEH DOSEN PEMBIMBING 1

N I M : **22 06 4173**
Nama Mahasiswa : **Leonard Rinaldy Mahakam P. G**
Judul Tugas Akhir : Simulasi Algoritma Welch-Powell untuk Melakukan Pewarnaan Graph.

Dosen Pembimbing I : Drs. R. Gunawan Santosa, M.Si.

Tanggal : 02-12-2011 **1**	Tanggal : 28-02-2012 **4**	Tanggal : 10-04-2012 **7**
Catatan Perkembangan/Revisi Skripsi : + lanjutkan Bab 2.	Catatan Perkembangan/Revisi Skripsi : - urutan berdasarkan urutan derajat - jika berderajat sama berdasarkan urutan penomeran vertex	Catatan Perkembangan/Revisi Skripsi : + pewarnaan manual + — dgn W. P.
Tanda Tangan Dosen	Tanda Tangan Dosen	Tanda Tangan Dosen
Tanggal : 12-01-2012 **2**	Tanggal : 31-03-2012 **5**	Tanggal : 25-04-2012 **8**
Catatan Perkembangan/Revisi Skripsi : * Bab 2, & bab 1 oke. * Lanjutkan Bab 3.	Catatan Perkembangan/Revisi Skripsi : - Ketika menjalankan program, ada keterangan untuk menjelaskan langkah per langkah - Nama vertex harus ada - penjelasan keterhubungan vertex	Catatan Perkembangan/Revisi Skripsi : - Program oke - Keterangan sudah oke
Tanda Tangan Dosen	Tanda Tangan Dosen	Tanda Tangan Dosen
Tanggal : 23-02-2012 **3**	Tanggal : 04-04-2012 **6**	Tanggal : 28-04-2012 **9**
Catatan Perkembangan/Revisi Skripsi : - Kanvas besar - Edge bisa mengikuti vertex (grak lebih)	Catatan Perkembangan/Revisi Skripsi : - Demo program · Algoritma Welch-Powell OK	Catatan Perkembangan/Revisi Skripsi : * Bab 3 revisi → lengkapi flowchart → tampilan rancangan input & output sistem masih ada yang ditambah
Tanda Tangan Dosen	Tanda Tangan Dosen	Tanda Tangan Dosen

UNIVERSITAS KRISTEN DUTA WACANA
FAKULTAS TEKNOLOGI INFORMASI
Program Studi Teknik Informatika

KARTU KONSULTASI SKRIPSI

LEMBAR INI DIISI OLEH DOSEN PEMBIMBING 2

N I M : **22 06 4173**
Nama Mahasiswa : **Leonard Rinaldy Mahakam P. G**
Judul Tugas Akhir : Simulasi Algoritma Welch-Powell untuk Melakukan Pewarnaan Graph.

Dosen Pembimbing II : Antonius Rachmat C, S.Kom., M.Cs.

Tanggal : 16/2 2012 **1**	Tanggal : 11-4-2012 **4**	Tanggal : 3-5-2012 **7**
Catatan Perkembangan/Revisi Skripsi : Bab I revisi penomoran Bab II tinjauan pustaka blm ada ! Tanda Tangan Dosen	Catatan Perkembangan/Revisi Skripsi : Demo program → manual ? masih ada bug di pewarnaan. Tanda Tangan Dosen	Catatan Perkembangan/Revisi Skripsi : Intisari revisi Kesimpulan revisi Pseudo code revisi Tanda Tangan Dosen
Tanggal : 23-2-2012 **2**	Tanggal : 1-5-2012 **5**	Tanggal : 8-5-2012 **8**
Catatan Perkembangan/Revisi Skripsi : Bab I & II Ok. Demo I → sudah tampil vertex & edge Tanda Tangan Dosen	Catatan Perkembangan/Revisi Skripsi : Bab III kurang struktur data + uji Demo tambah delete & koordinat Tanda Tangan Dosen	Catatan Perkembangan/Revisi Skripsi : Hasil penelitian ok - temuan berapa ? - tulis penyelesaian manual yg manual Tanda Tangan Dosen
Tanggal : 3-4-2012 **3**	Tanggal : 2-5-2012 **6**	Tanggal : 10-5-2012 **9**
Catatan Perkembangan/Revisi Skripsi : Demo tampilan graf sudah ok cuma belum bergerak. Tanda Tangan Dosen	Catatan Perkembangan/Revisi Skripsi : - Bab III ok - delete masih belum sempurna. Tanda Tangan Dosen	Catatan Perkembangan/Revisi Skripsi : Hasil penelitian III. ok, lampirkan. Tanda Tangan Dosen