

# REPRESENTASI GRAFIK VEKTOR MENGUNAKAN GRAPH DAN VEKTOR

**Alfa Pramudita – NIM : 13505026**

Program Studi Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Jl. Ganesha 10, Bandung  
Email : [alpha.deeto@gmail.com](mailto:alpha.deeto@gmail.com)

## Abstrak

Grafik vektor adalah sekumpulan simpul (*node/vertex*) yang dihubungkan oleh sisi (*edge/arc*) sehingga membentuk suatu garis maupun bidang dua dimensi. Simpul-simpul dari grafik vektor tersebut berderajat 1 atau 2. Tiap simpul memiliki vektor sejumlah derajat simpul tersebut yang merepresentasikan kelengkungan dari sisi yang menghubungkan simpul tersebut dengan simpul yang bertetangga dengan simpul tersebut.

Prosedur-prosedur yang dapat diimplementasikan untuk grafik vektor antara lain adalah penghapusan simpul suatu grafik vektor (*node deleting*), perubahan kelengkungan suatu sisi sebuah pada grafik vektor (*edge curving*), pemotongan sebuah grafik vektor (*cutting*), serta penggabungan 2 buah atau lebih grafik vektor (*combining*). Prosedur-prosedur tersebut dipergunakan dalam proses desain dalam aplikasi pengolah grafik vektor untuk menghasilkan suatu bentuk grafik vektor yang diinginkan.

**Kata Kunci :** grafik vektor (*vector graphic*), *raster graphic*, simpul (*node/vertex*), sisi (*edge/arc*), vektor, *node deleting*, *edge curving*, *cutting*, *combining*

## A. Pendahuluan

Dewasa ini aplikasi pengolah gambar semakin banyak digunakan baik oleh desainer profesional maupun masyarakat umum untuk membuat suatu desain grafis yang digunakan untuk mempercantik situs web, desain poster, animasi, logo perusahaan, dan lain-lain. Perusahaan penyedia aplikasi tersebut kini juga semakin banyak dengan mengandalkan kelebihan masing-masing aplikasi yang mereka buat.

Aplikasi-aplikasi tersebut bervariasi dalam fitur-fitur, tampilan, serta kemampuannya dalam mengolah gambar. Dengan beragam aplikasi dengan bermacam-macam fitur tersebut, pengguna aplikasi pengolah gambar tersebut dapat memilih aplikasi yang paling memenuhi kebutuhan mereka.

Dua jenis aplikasi pengolah gambar yang populer adalah aplikasi pengolah bitmap dan aplikasi pengolah grafik vektor. Aplikasi pengolah bitmap adalah aplikasi yang memiliki fitur-fitur dan fungsi-fungsi untuk memanipulasi hasil foto maupun file-file bitmap lainnya. Aplikasi ini pada umumnya digunakan untuk memperjelas gambar suatu foto, menambahkan efek-efek pada foto, serta memotong dan menggabungkan beberapa foto. Hasil dari aplikasi ini adalah sebuah *raster graphic*. Sedangkan aplikasi pengolah grafik vektor adalah aplikasi yang digunakan untuk membuat suatu grafik vektor yang memiliki fitur-fitur serta fungsi-fungsi yang dapat memanipulasi grafik vektor tersebut. Aplikasi ini banyak digunakan untuk membuat suatu desain yang memiliki komponen-komponen yang bukan berupa foto, misalnya kotak, lingkaran, elips, maupun bentuk-bentuk lainnya. Grafik vektor yang telah dibuat dapat dimanipulasi dengan fungsi-fungsi yang terdapat dalam aplikasi tersebut. Hasil dari aplikasi ini adalah sebuah grafik vektor (*vector graphic*)

Perbedaan *raster graphic* dan *vector graphic* adalah representasi image yang ditampilkan. Pada *raster graphic*, image yang ditampilkan direpresentasikan dengan titik-titik kecil yang ditempatkan pada kolom-kolom dan baris-baris yang masing-masing memiliki warna sendiri-sendiri dan membentuk sebuah image pada keseluruhan titik-titik tersebut. Sedangkan pada *vector*

*graphic*, image yang ditampilkan direpresentasikan dalam simpul-simpul dan sisi-sisi serta vektor-vektor yang mendeskripsikan image tersebut.



Figure 1. *raster graphic*

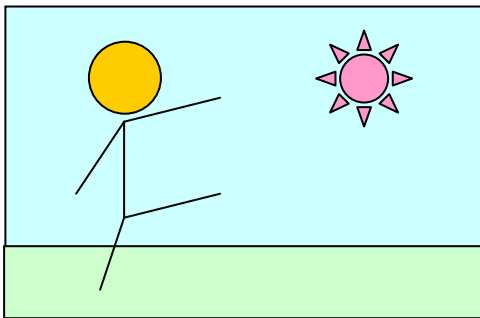


Figure 2. *vector graphics*

Pada *vector graphic* atau grafik vektor, simpul-simpul serta sisi-sisi yang merepresentasikannya menyerupai sebuah graph. Sedangkan vektor-vektor pada tiap simpul merepresentasikan kelengkungan sisi yang menghubungkan simpul tersebut. Prosedur-prosedur yang diimplementasikan dalam aplikasi pengolah grafik vektor menggunakan teori graph dalam pengimplementasiannya. Prosedur-prosedur tersebut akan dijelaskan lebih lanjut pada bab selanjutnya.

## B. Grafik Vektor

### Definisi

Grafik vektor adalah sebuah image yang direpresentasikan oleh sekumpulan simpul (*node/vertex*) yang tiap sisi terhubung dengan sisi lain oleh sebuah sisi (*edge/arc*) serta memiliki vektor-vektor pada tiap simpulnya.

Sebuah grafik vektor memiliki properti berupa posisi dan jarak masing-masing

simpul yang akan digunakan untuk menentukan panjang serta letak sisi-sisi yang menghubungkan simpul-simpul tersebut. Tiap simpul pada grafik vektor terhubung dengan sebuah simpul lainnya oleh suatu sisi. Oleh karena itu, grafik vektor merupakan graph terhubung, dan juga merupakan graph berbobot (*weighted graph*), karena tiap sisi memiliki panjang. Karena grafik vektor tidak berarah, maka grafik vektor adalah graph tidak berarah. Selain itu sebuah grafik vektor juga memiliki informasi atas seluruh vektor yang ada berikut arah dan panjang vektor-vektor tersebut untuk menentukan kelengkungan sisi-sisi pada grafik vektor tersebut.

Vektor-vektor yang berada pada simpul-simpul suatu grafik vektor merepresentasikan kelengkungan dari sisi yang menghubungkan simpul tersebut dengan simpul lain yang bertetangga. Kelengkungan dari sisi tersebut tergantung dari panjang dan arah vektor yang berada pada simpul-simpul yang dihubungkan oleh sisi tersebut. Kelengkungan dari suatu sisi atas pengaruh vektor-vektor tersebut akan dijelaskan lebih lanjut pada bagian implementasi prosedur *edge curving*.

Simpul-simpul suatu grafik vektor memiliki derajat 1 atau 2. Jika suatu simpul berderajat 1, maka grafik vektor yang direpresentasikan dengan simpul tersebut merupakan grafik vektor terbuka (grafik vektor berupa garis). Sebaliknya, jika semua simpul pada suatu grafik vektor berderajat 2, maka grafik vektor tersebut merupakan grafik vektor tertutup (grafik vektor berupa bentuk/blok). Grafik vektor tidak dapat memiliki derajat 0 (berupa titik, namun tidak dapat direpresentasikan dalam bentuk grafik), atau lebih dari 2. Dengan definisi tersebut, maka grafik vektor tersebut merupakan graph lingkaran dengan  $N$  simpul ( $C_n$ ).

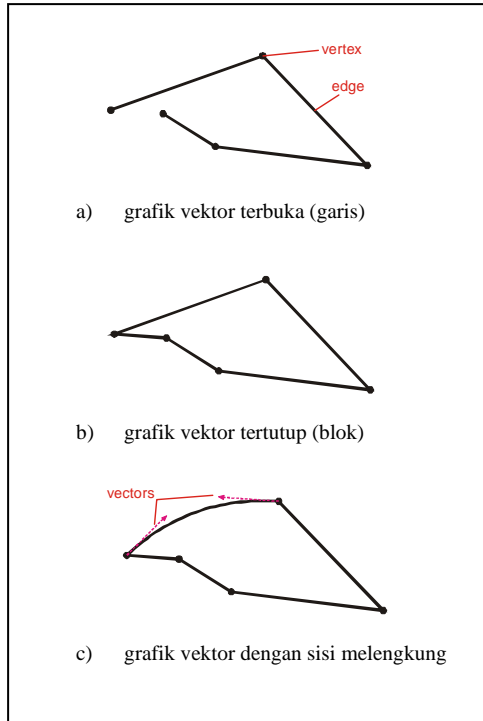


Figure 3. grafik vektor & elemennya

Lain halnya dengan graph pada umumnya, pada grafik vektor tidak berlaku adanya *isomorphic graph*, karena posisi simpul-simpul grafik vektor akan berbeda antara grafik vektor yang satu dengan yang lainnya. Hal ini berpengaruh pada sifat-sifat graph lainnya. Sebuah graph planar pada grafik vektor tidak akan memiliki graph bidang, karena graph bidang merupakan graph yang isomorfik dengan graph planar tersebut, dimana dalam konteks grafik vektor kali ini sifat isomorfik tidak berlaku. Meskipun demikian, untuk teorema Kuratowski, ketidakberlakuan sifat tersebut diabaikan, jadi setiap grafik vektor yang homeomorfik dengan suatu grafik vektor yang isomorfik dengan graph Kuratowski merupakan graph non-planar atau grafik vektor kompleks.

### C. Implementasi Prosedur-Prosedur untuk Grafik Vektor

Untuk memperjelas, grafik vektor akan direpresentasikan dalam sebuah tipe bentukan sebagai berikut :

```
Type vector : <
  len : integer
  //panjang vektor
  dir : integer
```

```
//arah vektor dalam derajat
>
Type node : <
  x : integer
  //posisi node pada sumbu-X
  y : integer
  //posisi node pada sumbu-Y
  dis : integer
  //jarak dg node selanjutnya
  //jika node terakhir, d = 0
  deg : integer
  //derajat node tsb
  skip : boolean
  //apakah node telah dihapus
  vectors : array[1..2] of vector
  //vektor-vektor pada node
>
Type vector_graphic : <
  N : integer
  //jumlah simpul
  nodes : array[1..N] of node
  //simpul-simpul pd grafik vektor
>
```

dimana simpul-simpul didefinisikan berurutan, sehingga sisi E1 akan bersisian dengan simpul node[1] dan node[2] dan memiliki panjang node[1].dis sehingga tidak perlu didefinisikan dalam sebuah tipe bentukan.

Dengan tipe bentukan di atas, prosedur-prosedur untuk grafik vektor diimplementasikan sebagai berikut :

#### 1. Node deletion

Prosedur ini menghapus simpul ke-i dari sebuah grafik vektor sesuai spesifikasi berikut :

- jika simpul ke-i berderajat 2, maka simpul tersebut dihapus dan sisi-sisi yang bersisian dengan node tersebut digabungkan menjadi 1 dan menghubungkan simpul ke-(i-1) dan simpul ke-(i+1)
- jika simpul ke-i berderajat 1 dan merupakan simpul pertama pada grafik vektor tersebut, maka simpul tersebut dihapus dan derajat simpul ke-(i+1) dikurangi 1
- jika simpul ke-i berderajat 1 dan merupakan simpul terakhir pada grafik vektor tersebut, maka simpul tersebut dihapus dan derajat simpul ke-(i-1) dikurangi 1
- jika sebuah node memiliki derajat 0 maka node tersebut dihapus

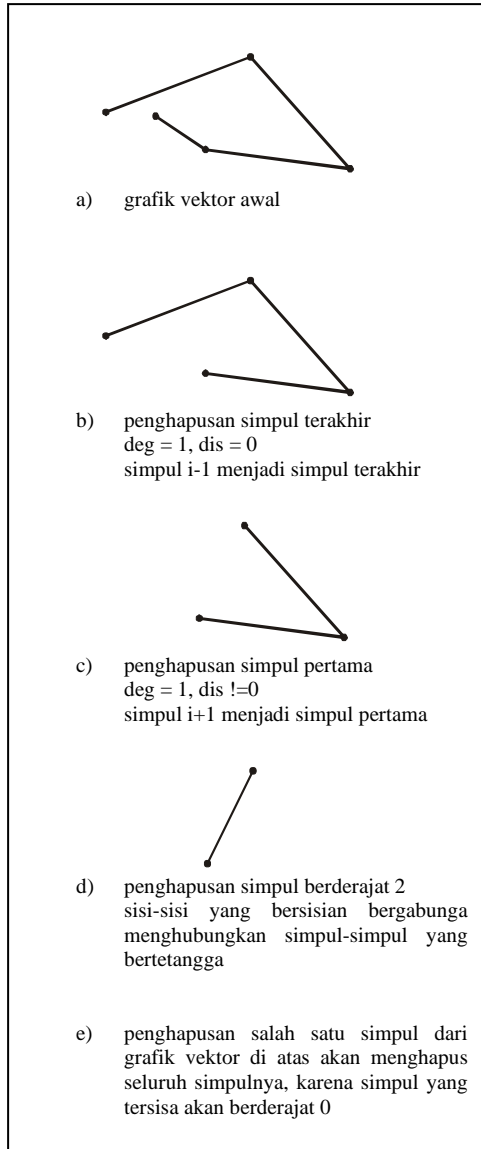


figure 4. contoh node deletion

Prosedur ini memiliki 2 parameter :

- G : vector\_graphic (input/output)  
Grafik vektor yang akan dihapus salah satu simpulnya.
- i: integer (input)  
Integer yang merupakan index dari simpul yang akan dihapus

Procedure NodeDel

(I/O G : grafik\_vektor; i : integer);

Initial state :

Grafik vektor G terdefinisi.  
i terdefinisi sebagai parameter

Final state :

Simpul ke-i akan dihapus sesuai spesifikasi

Algorithm

```
G.nodes[i].skip = 1;
if G.nodes[i].deg = 2 then
  G.nodes[i-1].dis = G.nodes[i-1].dis +
  G.nodes[i].dis
else if G.nodes[i].deg = 1 and
G.nodes[i].dis != 0 then
  G.nodes[i+1].dis = 0;
  G.nodes[i+1].deg --
else if G.nodes[i].deg = 1 and
G.nodes[i].dis = 0 then
  G.nodes[i-1].dis = 0
  G.nodes[i-1].deg --;
for j = i-1 to i+1 do
  if G.nodes[j].deg = 0 then
    G.nodes[j].skip = 1;
```

## 2. Edge curving

Prosedur ini mengubah kelengkungan suatu sisi dari sebuah grafik vektor dengan cara mengubah properti panjang dan arah dari vektor tiap-tiap simpul yang bersisian dengan sisi tersebut. Vektor-vektor tersebut akan mengubah kelengkungan sisi yang bersisian dengan ketentuan sebagai berikut :

- jika simpul ke-i berderajat 1 maka vektor [1] mengubah kelengkungan dari sisi yang bersisian dengan simpul ke-i
- jika simpul ke-i berderajat 2 maka vektor yang digunakan untuk mengubah kelengkungan sisi yang menghubungkan dengan simpul ke(i+1) adalah vektor[2], sedangkan vektor[1] digunakan untuk mengubah sisi yang menghubungkan dengan simpul ke-(i-1)
- jika simpul ke-(i+1) berderajat 2 maka vektor yang digunakan untuk mengubah kelengkungan sisi yang menghubungkan dengan simpul ke-(i-1) adalah vektor[1], sedangkan vektor[2] digunakan untuk mengubah sisi yang menghubungkan dengan simpul ke-(i+2)

Prosedur ini memiliki 6 parameter :

- G : vector\_graphic (input/output)  
Grafik vektor yang akan diubah kelengkungan salah satu sisinya.
- i : integer (input)  
Integer yang merupakan index dari simpul awal dari sisi yang akan diubah kelengkungannya

- dir1 : integer (input)  
Integer yang berisi nilai arah vektor awal yang akan diubah dalam derajat
- dir2 : integer (input)  
Integer yang berisi nilai arah vektor akhir yang akan diubah dalam derajat
- len1 : integer (input)  
Integer yang berisi nilai panjang vektor awal yang akan diubah dalam derajat
- len2 : integer (input)  
Integer yang berisi nilai panjang vektor akhir yang akan diubah dalam derajat

Procedure EdgeCurve

(I/O G : vector\_graphic; i, dir1, dir2, len1, len2 : integer);

Initial State :

Grafik vektor G terdefinisi  
Seluruh parameter terisi  
Simpul ke-i dan simpul ke-(i-1) terhubung

Final State :

Sisi antara simpul ke-i dan simpul ke-(i+1) diubah kelengkungannya sesuai parameter yang dimasukkan

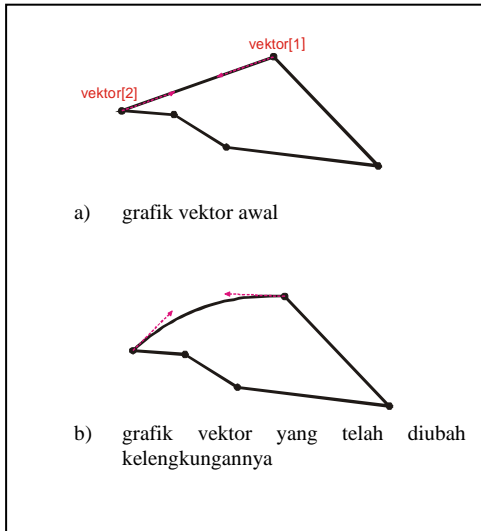


figure 5. contoh edge curving

Algorithm

```

if G.nodes[i].deg = 1 then
  G.nodes[i].vektor[1].dir = dir1;
  G.nodes[i].vektor[1].len = len1
else if G.nodes[i].deg = 2 then
  G.nodes[i].vektor[2].dir = dir1;
  G.nodes[i].vektor[2].len = len1;
  G.nodes[i+1].vektor[1].dir = dir1;
  G.nodes[i+1].vektor[1].len = len1;

```

**3. Cutting**

Prosedur *cutting* digunakan untuk membagi sebuah grafik vektor menjadi 2 buah grafik vektor yang merupakan subgraph dari grafik vektor yang lama. Prosedur ini menggunakan sebuah garis lurus yang memotong sebuah grafik vektor menjadi dua bagian. Garis tersebut direpresentasikan dengan empat buah integer, yaitu kedua koordinat masing-masing ujung dari garis tersebut. Grafik vektor yang terpotong oleh garis tersebut kemudian dibagi menjadi dua bagian sesuai perpotongan garis tersebut

Algoritma serta spesifikasi prosedur yang dibahas di sini mungkin berbeda dengan yang diimplementasikan pada aplikasi-aplikasi pengolah grafik vektor yang ada, namun beberapa poin penting yang merupakan inti dari prosedur *cutting* ini tetap mengacu pada prosedur-prosedur yang telah ada.

Prosedur *cutting* ini memiliki beberapa ketentuan dalam melakukan pembagian suatu grafik vektor, yaitu :

- grafik vektor yang akan dibagi harus memiliki paling tidak 2 buah simpul, serta garis pemotong tidak berada pada salah satu simpul tersebut
- apabila jumlah simpul pada grafik vektor yang akan dipotong lebih dari 3, dan kedua ujung garis pemotong masing-masing berada pada sebuah simpul, maka tidak perlu dibuat simpul baru dalam grafik vektor tersebut
- grafik vektor baru yang dibentuk akan memiliki dua buah simpul tambahan apabila tidak ada ujung garis pemotong yang terletak pada suatu simpul, atau memiliki sebuah simpul tambahan apabila salah satu dari ujung garis pemotong tersebut terletak pada sebuah simpul dari grafik vektor yang akan dipotong
- setelah proses pemotongan selesai, grafik vektor yang lama akan hilang, dan digantikan oleh dua buah grafik vektor baru sesuai perpotongan yang dibuat

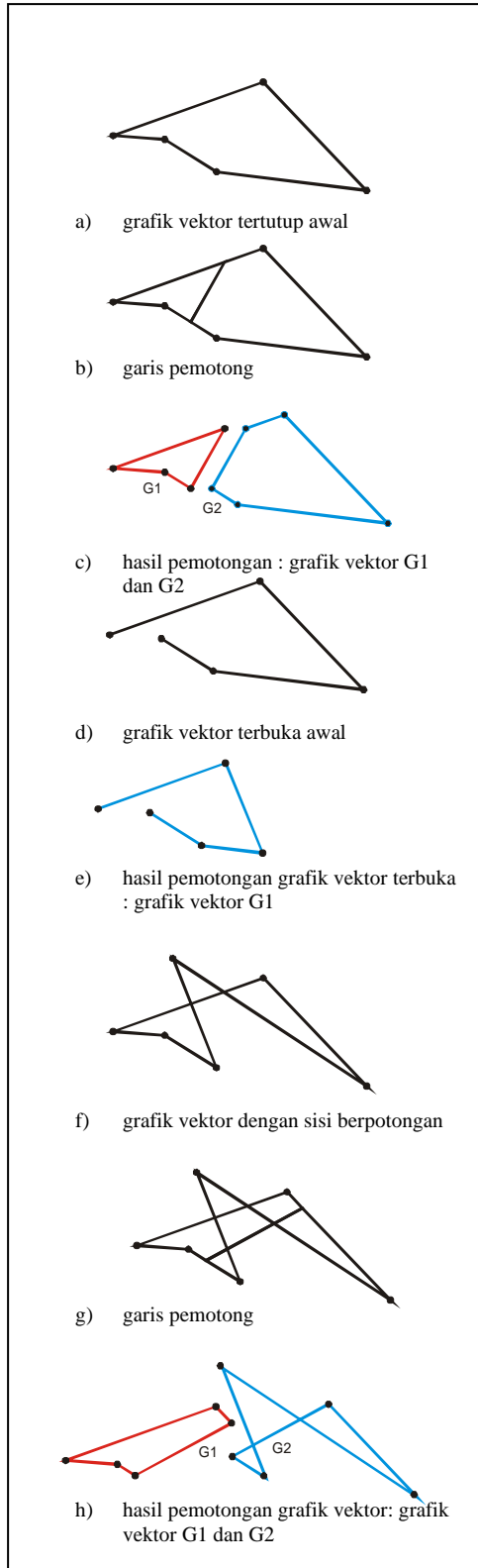


figure 6. contoh cutting

Parameter-parameter yang terdapat dalam prosedur ini antara lain :

- G : vector\_graphic (input/output)  
Grafik vektor yang akan dibagi dua
- G1 : vector\_graphic (output)  
Hasil pemotongan pertama dari grafik vektor G
- G2 : vector\_graphic (output)  
Hasil pemotongan kedua dari grafik vektor G
- x1 : integer (input)  
Koordinat X ujung1 garis pemotong
- x2 : integer (input)  
Koordinat X ujung2 garis pemotong
- y1 : integer (input)  
Koordinat Y ujung1 garis pemotong
- y2 : integer (input)  
Koordinat Y ujung2 garis pemotong

Procedure cut

(I/O G : vector\_graphic; O G1, G2 : vector\_graphic; x1, x2, y1, y2 : integer);

Initial State :

- G, G1, G2 terdefinisi
- Seluruh parameter terisi
- G minimal memiliki 2 buah simpul

Final State :

- G dihilangkan, diganti oleh G1 dan G2 yang merupakan subgraph (dengan 2 simpul tambahan, apabila memenuhi ketentuan no3) dari grafik vektor yang lama

Algorithm

Algoritma dari prosedur ini cukup rumit apabila dituliskan dalam bentuk *pseudo code*. Oleh karena itu, dalam pembahasan ini algoritma prosedur ini dituliskan dalam Bahasa Indonesia sebagai berikut :

- tentukan sebuah garis yang memotong sebuah grafik vektor G menjadi dua bagian
- jika memenuhi ketentuan, buat simpul baru pada perpotongan paling atas dan paling bawah antara garis pemotong dengan grafik vektor G
- cari rute terdekat yang dapat dilalui dari perpotongan atas ke perpotongan bawah
- isikan grafik vektor G1 dengan komponen-komponen lintasan tersebut, lalu hubungkan simpul perpotongan atas dan perpotongan bawah dengan sebuah sisi baru

- isikan grafik vektor G2 dengan komponen-komponen grafik vektor G yang tidak dilalui lintasan ditambah dengan simpul perpotongan atas dan perpotongan bawah, lalu hubungkan simpul perpotongan atas dan simpul perpotongan bawah tersebut dengan sebuah sisi baru
- hapus grafik vektor G

Algoritma di atas juga berlaku untuk grafik vektor terbuka dan grafik vektor kompleks. Untuk grafik vektor terbuka, grafik vektor baru yang dibentuk hanya 1 buah, yaitu yang mengandung simpul-simpul berderajat 1 yang dihubungkan dengan sebuah sisi yang segaris dengan garis pemotong.

#### 4. Combining

Kebalikan dari prosedur *cutting*, prosedur ini menggabungkan dua buah grafik vektor yang sisi-sisinya saling berpotongan. Hasil dari prosedur ini adalah sebuah grafik vektor gabungan dari dua grafik vektor yang sebelumnya. Meskipun disebut sebagai kebalikan dari prosedur *cutting*, namun prosedur ini tidak bekerja secara kebalikan dari prosedur *cutting*. Hasil dari prosedur ini tidak mengembalikan properti dari grafik vektor yang telah dipotong menjadi kembali seperti semula. Prosedur ini memiliki spesifikasi sendiri dalam pengolahan dua grafik vektor tersebut. Spesifikasi-spesifikasi dalam prosedur *combining* ini antara lain :

- dua grafik vektor yang saling berpotongan sisi-sisinya akan digabungkan menjadi sebuah grafik vektor kompleks yang memiliki jumlah simpul sebanyak jumlah simpul kedua grafik vektor yang digabungkan
- dua grafik vektor yang tidak berpotongan atau salah satunya berada di dalam grafik vektor yang lain tidak dapat digunakan dalam prosedur ini, karena hasil dari *combining* kedua grafik vektor tersebut tidak memenuhi definisi sebuah grafik vektor, yaitu tidak semua simpul dari grafik vektor tersebut terhubung dengan simpul lainnya
- hasil dari penggabungan kedua grafik vektor tersebut merupakan sebuah grafik vektor yang memiliki simpul-

simpul dari grafik vektor yang digabungkan ditambah dengan simpul-simpul baru yang terbentuk dari perpotongan grafik vektor yang digabungkan.

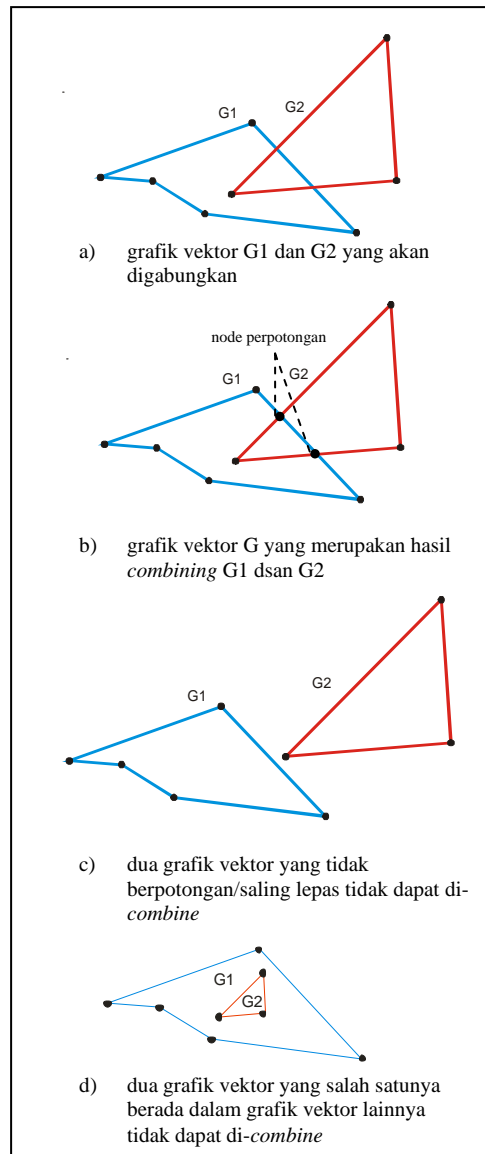


figure 7. contoh *combining*

Parameter-parameter dalam prosedur ini sebanyak 3 buah parameter, yaitu :

- G1 : vector\_graphic (input/output)  
Grafik vektor pertama yang akan di-*combine*
- G2 : vector\_graphic (input/output)  
Grafik vektor kedua yang akan di-*combine*
- G : vector\_graphic (output)

Grafik vektor hasil penggabungan kedua grafik vektor sebelumnya

Procedure combine

(I/O G1, G2 : vector\_graphic; O G : vector\_graphic)

Initial State :

G1, G2, G terdefinisi

G1 dan G2 saling berpotongan

Final State :

G1 dan G2 digabungkan menjadi grafik vektor G

Algorithm :

Algoritma dari prosedur *combining* ini tidak terlalu rumit, namun untuk penulisan dalam *pseudo code* memerlukan beberapa prosedur/fungsi tambahan yang sebaiknya diimplementasikan terlebih dahulu. Sebagaimana prosedur *cutting*, algoritma prosedur ini akan dijabarkan dalam Bahasa Indonesia agar lebih mudah dan lebih jelas untuk dimengerti.

- posisi kedua grafik vektor diketahui dan telah divalidasi
- buat simpul-simpul baru sejumlah perpotongan sisi-sisi dari kedua buah grafik vektor
- isikan simpul-simpul G dengan seluruh simpul G1 dan G2 serta node perpotongan yang telah dibuat, sesuaikan penamaan simpul-simpulnya
- hapus kedua grafik vektor awal yang telah digabungkan

#### D. Kesimpulan

Dari pembahasan di atas, dapat diambil kesimpulan sebagai berikut :

- Grafik vektor adalah sebuah image yang direpresentasikan oleh sekumpulan simpul (*node/vertex*) yang tiap sisi terhubung dengan sisi lain oleh sebuah sisi (*edge/arc*) serta memiliki vektor-vektor pada tiap simpulnya.
- Metode manipulasi grafik vektor dapat diimplementasikan dengan mengaplikasikan teori graph dan vektor di dalamnya, antara lain node deletion, edge curving, cutting, serta combining.

- Node deletion adalah sebuah prosedur untuk menghapus salah satu simpul dari suatu grafik vektor.
- Edge curving adalah sebuah prosedur untuk mengubah kelengkungan suatu sisi grafik vektor dengan mengubah arah serta panjang vektor dari simpul yang bersisian dengan sisi tersebut.
- Cutting adalah sebuah prosedur untuk memisahkan sebuah grafik vektor dengan sebuah garis hingga menghasilkan dua buah grafik vektor dengan masing-masing memiliki minimal satu buah komponen.
- Combining adalah sebuah prosedur untuk menggabungkan dua buah grafik vektor yang berpotongan sehingga terbentuk sebuah grafik vektor yang merupakan gabungan seluruh kedua grafik vektor tersebut ditambah dengan simpul perpotongan antara kedua grafik vektor tersebut.

#### E. Referensi

- [1] Munir, Rinaldi. "Materi Kuliah Graf", 2008.
- [2] Microsoft Encarta Encyclopedia 2007
- [3] Corel Draw X3
- [4] Jenis Grafik Vektor, <http://one.indoskripsi.com/>, Desember 2008.