



## Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)

Tahun / Semester : 2020-2021 / Genap

Modul : 8 - *Algorithms*

Hari, Tanggal Praktikum : Rabu, 14 April 2021

---

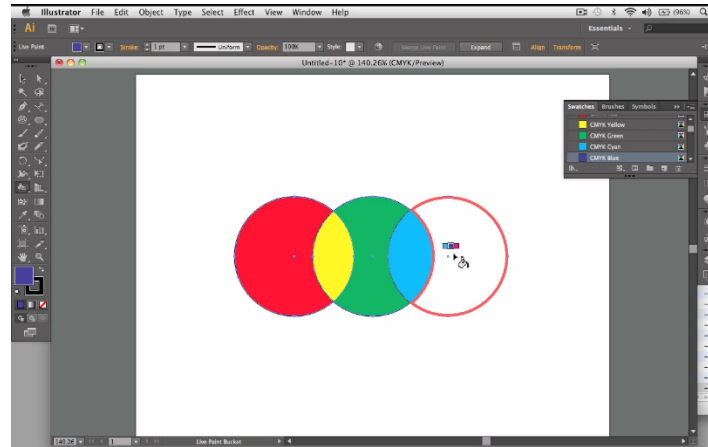
### Naskah Soal Praktikum

Pembuat Naskah: David Fauzi, Oktavianus Irvan Sitanggang

#### Ketentuan:

1. Kerjakanlah satu dari dua soal berikut pada *template repository* yang anda peroleh ketika mengambil *assignment* di GitHub Classroom praktikum!
2. *Commit* yang dilakukan setelah sesi praktikum berakhir tidak akan dipertimbangkan dalam penilaian.
3. *Header* setiap *file* harus mengikuti format yang telah disediakan pada *file template repository*. *Header* yang tidak mengikuti format tersebut tidak akan dinilai.
4. Buku catatan laboratorium yang berisi *flowchart*, *data flow diagram*, dan analisis kompleksitas waktu dan ruang Big O dari solusi yang anda buat dikumpulkan ke tugas.stei.itb.ac.id paling lambat 24 jam setelah sesi praktikum. Tulis pula alasan anda memilih mengerjakan soal yang anda kerjakan dan mengapa bukan soal yang lainnya!
5. Solusi soal pertama harus dapat dikompilasi dengan perintah `make soal-01` dan menghasilkan *file executable* dengan nama `soal-01`. Demikian pula, soal kedua harus dapat dikompilasi dengan perintah `make soal-02` dan menghasilkan *file executable* dengan nama `soal-02`.
6. Bila diperlukan, sesuaikanlah isi *Makefile* yang tersedia pada *template repository* untuk memenuhi syarat kompilasi dan *file* keluaran di atas!

## Soal 1

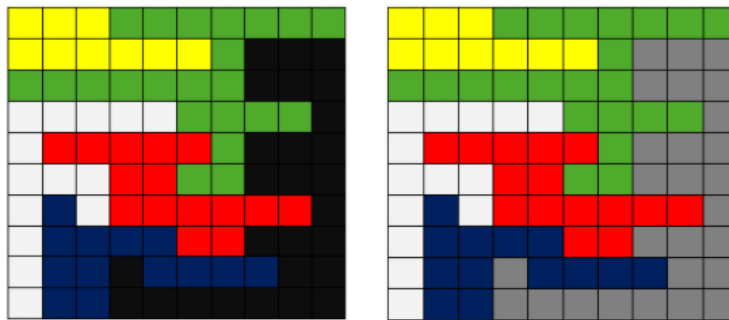


Kak Araragi sedang meng-*edit* gambar dengan menggunakan Photoshop. *Paint bucket tools* merupakan salah satu tools yang biasa digunakan dalam aplikasi *image editing* untuk mengisi suatu area gambar berdasarkan kesamaan warna agar menjadi warna lain yang diinginkan. Kak Araragi ingin mengimplementasikan *paint bucket tools* ini menggunakan bahasa C. Algoritma yang biasa diimplementasikan untuk tools ini adalah *flood algorithm (seed algorithm)* yang menggunakan BFS (Breadth First Search). Bantulah Kak Araragi untuk membuat suatu program C yang menerima input file .txt yang berisi matriks gambar lalu memilih koordinat pixel yang ingin di-klik lalu pilihan warnanya. Diasumsikan bahwa suatu gambar direpresentasikan dalam matriks dengan ukuran baris dan kolom tertentu yang masing-masing selnya berisi warna yang disimbolkan dengan karakter. Untuk menyederhanakan persoalan, diberikan beberapa permisalan warna dalam simbol karakter

- X : Hitam
- W : Putih
- R : Merah
- G : Hijau
- B : Biru
- Y : Kuning
- A : Abu-abu

Sebagai contoh, lihat gambar dibawah ini. Pada file .txt, *line* pertama berisi jumlah baris, *line* kedua berisi jumlah kolom, dan *line* selanjutnya adalah matriks dari gambar.

Isi File .txt	Setelah <i>paint bucket</i>
10	YYYYGGGGGG
10	YYYYYYGAAA
YYYYGGGGGG	GGGGGGGAAA
YYYYYYGXXX	WWWWWGGGGA
GGGGGGGXXX	WRRRRRGAAA
WWWWWGGGGX	WWWRRGGAAA
WRRRRRGXXX	WBWRRRRRRA
WWWRRGGXXX	WBBBBRRAAA
WBWRRRRRX	WBBABBBBAA
WBBBBRRXXX	WBBAAAAAAA
WBBXBBBBXX	
WBBXXXXXXX	



Jika kita ingin mengubah bagian hitam menjadi abu-abu, dapat di-klik pixel hitam di koordinat 2,8 (baris 2, kolom 8) untuk dilakukan *fill* menggunakan *paint bucket tool* menjadi abu-abu. Berikut adalah beberapa informasi mengenai pengerjaan soal :

- Soal harus diimplementasikan menggunakan *queue* dan Algoritma BFS (*Breadth First Search*). Pada algoritma *flood* menggunakan BFS, perhatikan bahwa pencarian *pixel-pixel adjacent* (tetangga) adalah ke-8 tetangga dari *pixel* tersebut (tidak hanya atas-bawah dan kanan-kiri saja).
- Telah diberikan template kode dimana sudah terdapat struct *queue* serta sudah diisi juga implementasi fungsi-fungsi untuk *parsing* dan pembuatan matriks dan telah diisi juga implementasi fungsi *main()*. Tugas anda adalah mengisi fungsi-fungsi belum terisi yang terdiri dari fungsi-fungsi dasar *queue* (*create queue*, *enqueue*, *dequeue*, *peek*), fungsi pencetakan matriks gambar, dan fungsi algoritma dari *fill* (*paint bucket*).
- Disarankan agar anda tidak merubah kode selain implementasi fungsi. Namun, anda tetap diperbolehkan untuk membuat fungsi baru yang dapat membantu implementasi kode anda.
- Asumsikan seluruh *input* dan file .txt adalah valid sehingga tidak perlu validasi lagi.

### Contoh eksekusi program (garis bawah menunjukkan input)

#1

Nama file: 01.txt

Gambar:

```
XXXXXXXXXXXXX
XXYYYYXXYYXX
XXYYYYXXYYXX
XXXXXXXXXXXXX
XXXXXXXXXXXXX
XXYYYYYYXX
XXYYYYYYXX
XXXXXXXXXXXXX
```

Masukan koordinat pixel yang ingin diklik serta warna penggantinya

Titik X(baris): 1

Titik Y(kolom): 1

Ganti warna menjadi: W

Titik (1,1) dengan warna X -> W

Gambar setelah di bucket fill:

```
WWWWWWWWWWWW
WWYYYWWYYYWW
WWYYYWWYYYWW
WWWWWWWWWWWW
WYWWWWWWWWY
WWYYYYYYWW
WWWWWWWWWWWW
```

#2

Nama file: 02.txt

Gambar:

```
YYYGGGGGG
YYYYYYGXXX
GGGGGGGXXX
WWWWWGGGGX
WRRRRRGXXX
WWRRRGXXX
WBWRRRRRX
WBBBRRXXX
WBBXBBBXX
WBBXXXXXX
```

Masukan koordinat pixel yang ingin diklik serta warna penggantinya

Titik X(baris): 1

Titik Y(kolom): 4

Ganti warna menjadi: A  
Titik (1,4) dengan warna G -> A

Gambar setelah di bucket fill:

```
YYYYAAAAAA  
YYYYYYAXXX  
AAAAAAAXXX  
WWWWWAAAAX  
WRRRRRAXXX  
WWWRRAXXX  
WBWRRRRRX  
WBBBBRRXXX  
WBBXBBBBXX  
WBBXXXXXXX
```

**#3**

Nama file: 03.txt

Gambar:

```
WRWWWWWRW  
WWRWWWRWW  
WWWRRWWWW  
WWWRRWWWW  
WWWRRWWWW  
WWWRRWWWW  
WWRWWWRWW  
WRWWWWWRW
```

Masukan koordinat pixel yang ingin diklik serta warna penggantinya

Titik X(baris): 1

Titik Y(kolom): 2

Ganti warna menjadi: B

Titik (1,2) dengan warna R -> B

Gambar setelah di bucket fill:

```
WBWWWWWBW  
WWBWWWBWW  
WWWBWBWWW  
WWWBBWWW  
WWWBBWWW  
WWWBWBWWW  
WWBWWWBWW  
WBWWWWWBW
```

## Soal 2

Anda diberikan sebuah angka. Lalu, seorang asisten gabut menyuruh anda untuk melihat tingkat kecantikan angka tersebut dan melihat potensi kecantikan terbaik dari angka tersebut. Menurut asisten tersebut, angka yang cantik memenuhi syarat sebagai berikut :

- Jika angka terdiri dari 1 digit, maka tingkat kecantikan angka tersebut 1
- Jika angka terdiri dari 2 digit, tingkat kecantikan angka tersebut 1
- Jika angka tersebut terdiri dari  $n$  digit, maka
  - o Jika  $n$  genap, maka tingkat kecantikan angka tersebut adalah nilai paling besar antara kecantikan setengah digit pertama dan setengah digit kedua lalu nilai tersebut ditambah 1
  - o Jika  $n$  ganjil, maka digit dibagi 2 dengan
    - Jika nilai tengah ganjil, maka nilai tengah ditambahkan ke akhir setengah digit pertama
    - Jika nilai tengah genap, maka nilai tengah dimasukkan ke awal setengah digit terakhir
    - e.g. 12345  $\rightarrow$  123 45      dan      12845  $\rightarrow$  12 845

Nilai kecantikan angka merupakan nilai paling besar kecantikan antara setengah digit pertama dan setengah digit kedua lalu nilai tersebut ditambah 1

Untuk memperoleh potensi kecantikan terbaik, maka cara memperolehnya :

- Jika angka terdiri dari 1 digit, biarkan
- Jika angka 2 digit, jika digit pertama lebih besar dari digit kedua, maka tukar posisi digit
- Jika terdiri dari  $n$  digit, dengan menggunakan pembagian yang sama dengan angka cantik, tukar bagian setengah digit pertama dengan setengah digit kedua.

Tentukanlah tingkat kecantikan angka dan potensi terbaik! (asumsikan input selalu valid dan jumlah digit selalu lebih kecil dari 100)

## Catatan

Contoh Kecantikan

Kecantikan 12345 : 3

Kecantikan 123 : 2

Kecantikan 1 : 1

Kecantikan 23 : 1

Kecantikan 45 : 1

Contoh Potensi :

Potensi 12345 : 45231

Potensi 123 : 231

Potensi 1 : 1

Potensi 23 : 23

Potensi 45 : 45

**Contoh eksekusi program (garis bawah menandakan input)**

```
#1
12345
Tingkat Kecantikan = 3
Potensi = 45231
```

```
#2
123456
Tingkat Kecantikan = 3
Potensi = 645231
```