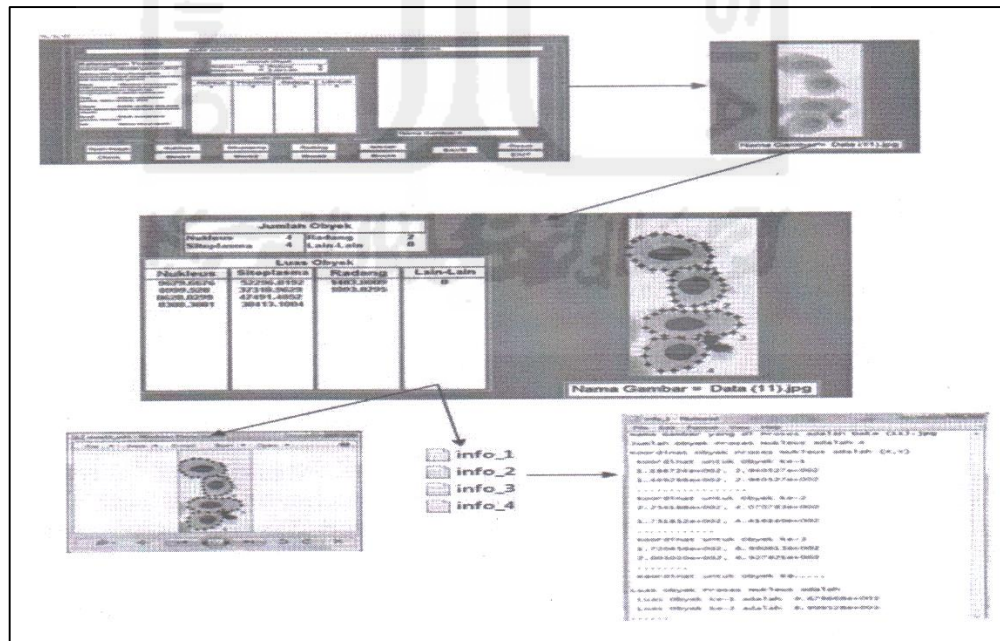


# LANDASAN TEORI

Pada penelitian tersebut anotasi yang dilakukan oleh pengguna adalah dengan memberikan batas objek secara manual dalam bentuk titik-titik koordinat dan aplikasi akan secara otomatis menghubungkan titik-titik koordinat tersebut dengan garis lurus. Hal ini mengakibatkan kurva objek yang dibentuk tidak mengcover secara utuh objek yang dianotasi. Data keluaran dari aplikasi ini adalah data koordinat untuk setiap objek dalam format teks (.txt) dan citra teranotasi dalam format .jpg. Ilustrasi aplikasi dapat dilihat pada gambar 2.1 berikut.

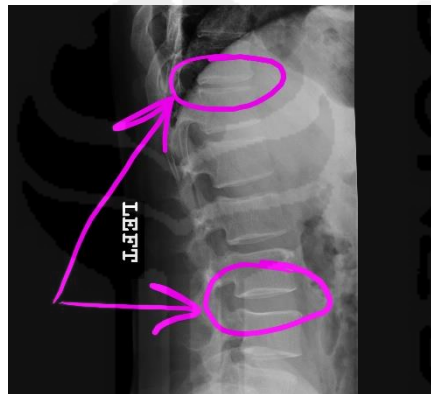


(Muhimmah & Pranajaya, 2012)

## 2.2 Anotasi Digital

Anotasi menurut Kamus Besar Bahasa Indonesia adalah catatan yang dibuat oleh pengarang atau orang lain untuk menerangkan, mengomentasi, atau mengkritik teks karya sastra atau bahan tertulis lainnya. Catatan tersebut ditambahkan langsung dengan menandai bagian yang ingin diterangkan, dikomentari, atau dikritik.

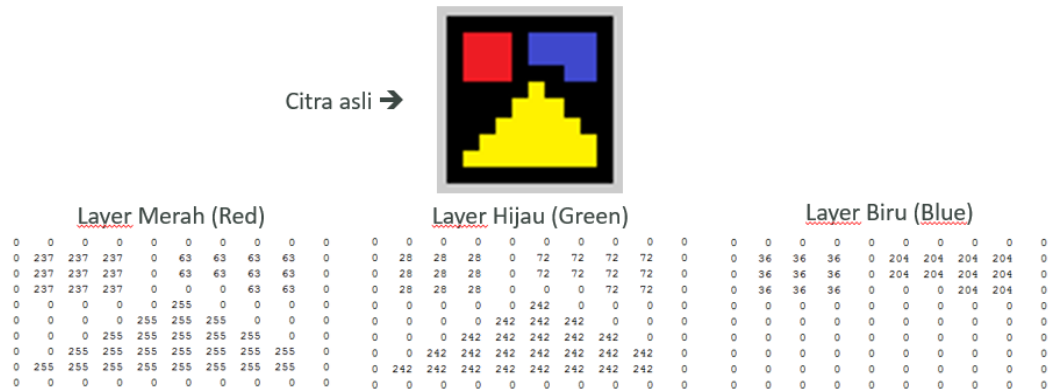
Anotasi citra digital pada dunia medis adalah proses meningkatkan citra dengan informasi untuk menyampaikan atau menerangkan objek temuan ahli radiologi (Preim & Bartz, 2007). Anotasi tersebut dibuat dengan menempatkan panah yang menunjukkan lokasi anotasi atau dengan melingkari objek yang ditemukan. Anotasi tersebut berfungsi untuk menunjukkan lokasi, bentuk, dan fitur-fitur lain dari objek yang ditemukan. Ilustrasi dapat dilihat pada gambar 2.2.



**Gambar 2.2** Ilustrasi anotasi citra medis  
(sumber : [www.wikipedia.org](http://www.wikipedia.org))

## 2.3 Citra

Citra adalah versi digital dari dokumen, seperti hasil fotografi, literatur, karya seni, dan sebagainya. Citra direpresentasikan ke dalam sebuah matriks dengan baris dan kolom sebanyak piksel citra tersebut, ilustrasi dapat dilihat pada gambar 2.3. Persimpangan antara baris dan kolom pada matriks citra disebut piksel. Sebuah piksel memiliki indeks baris (x) dan kolom (y), dan berisikan bilangan bulat dari nilai keabuan atau warna pada piksel tersebut (Low, 1991). Piksel (0,0) atau piksel awal terletak pada sudut kiri atas, kemudian indeks x mengarah ke kanan dan indeks y mengarah ke bawah (Usman, 2005).



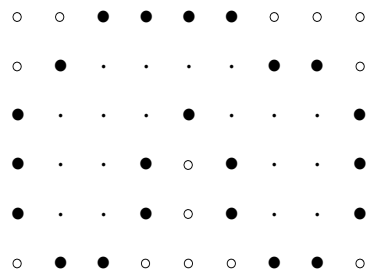
**Gambar 2.3** Ilustrasi citra RGB (*red, green, blue*)

## 2.4 *Region Filling* dengan Metode *Scanline Fill*

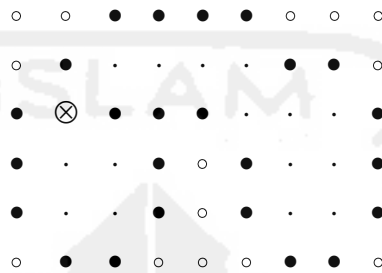
*Region filling* adalah proses pada pengolahan citra yang digunakan untuk mengisi piksel pada wilayah objek atau sesuai dengan batas objek yang ditentukan. Proses ini membutuhkan batas objek sebagai batas wilayah yang akan diisi. Berikut adalah langkah-langkah *region filling* dengan metode *scanline fill*.

1. Tentukan piksel awal (*seed*), dengan syarat piksel tersebut berada di dalam batas objek.
2. Isi piksel pada baris yang sama dengan *seed*.
3. Cari *seed* selanjutnya pada baris di atas dan di bawah *seed* awal, dengan syarat:
  - a. Piksel pertama setelah batas objek paling kiri,
  - b. Piksel selanjutnya di dalam batas objek dengan tetangga kirinya adalah piksel di luar batas objek.
4. Ulangi langkah 2 dan 3 sampai tidak ditemukan lagi piksel kosong di dalam batas objek atau tidak ada lagi *seed* baru.

Ilustrasi *region filling* dengan metode *scanline fill* dapat dilihat pada gambar 2.4 sampai dengan gambar 2.7 berikut.

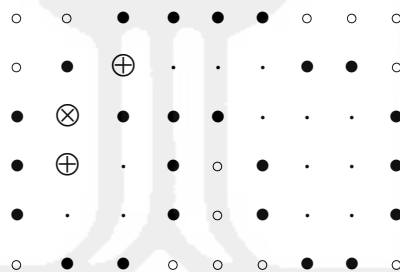


**Gambar 2.4** Ilustrasi batas objek



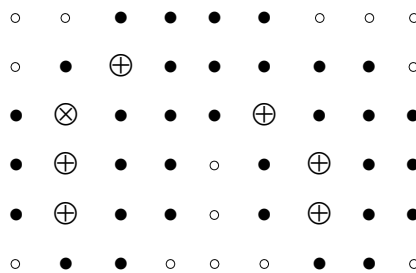
**Gambar 2.5** Langkah 1 dan 2

( $\otimes$  merupakan *seed* awal)



**Gambar 2.6** Langkah 3

( $\oplus$  merupakan *seed* baru)

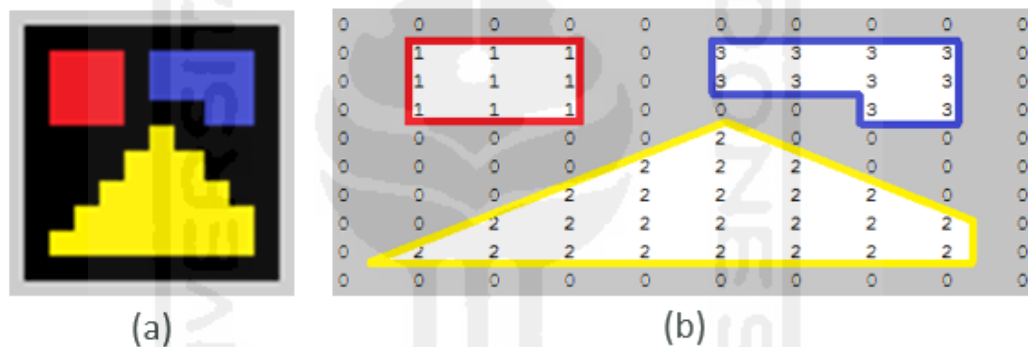


**Gambar 2.7** Hasil akhir

## 2.5 Labeling

*Labeling* pada pemrosesan citra adalah proses pemberian label yang sama pada piksel yang berdekatan atau piksel yang merupakan komponen pembentuk objek pada citra. Label yang diberikan pada piksel tersebut biasanya berupa nomor objek, sedangkan pada piksel yang tidak terdapat objek atau piksel yang tidak membentuk suatu objek akan dikenakan label 0 (Mozef, 2004).

*Labeling* merupakan tahap yang penting pada proses pengolahan citra. *Labeling* membantu untuk mempelajari ciri dari objek yang diberikan label, mulai dari besar objek, bentuk objek, sampai dengan tekstur objek atau pola objek. Ilustrasi dapat dilihat pada gambar 2.8 dengan (a) merupakan citra asli dan (b) merupakan hasil *labeling*.



**Gambar 2.8** Ilustrasi *labeling*

## 2.6 Invers Matriks dengan Metode Gauss-Jordan

Matriks adalah kumpulan bilangan, simbol, atau ekspresi yang disusun menurut baris dan kolom. Pertemuan baris dan kolom (bilangan, simbol, atau ekspresi) disebut dengan elemen atau anggota matriks, ilustrasi pada gambar 2.9 yaitu matriks M dengan 3 baris dan 3 kolom atau dapat juga disebut matriks persegi.

$$M = \begin{bmatrix} 3 & -0,1 & -0,2 \\ 0,1 & 7 & -0,3 \\ 0,3 & -0,2 & 10 \end{bmatrix}$$

**Gambar 2.9** Matriks M

Invers dari sebuah matriks adalah kebalikan dari matriks tersebut. Invers dari matriks  $M$  pada gambar 2.9 dinotasikan dengan  $M^{-1}$ . Matriks dikatakan dapat mempunyai invers hanya apabila memenuhi syarat berikut.

1. Mempunyai baris dan kolom yang sama (matriks  $n \times n$ ).
2. Determinan matriks tersebut tidak sama dengan 0.

Metode Gauss-Jordan adalah pengembangan dari metode eliminasi Gauss. Metode Gauss-Jordan ini menyajikan teknik metode numerik yang lebih sederhana dan mudah diaplikasikan untuk komputasi invers matriks (Chapra & Canale, 1990). Berikut adalah langkah-langkah mencari invers matriks dengan metode Gauss-Jordan.

1. Masukkan matriks, dinotasikan dengan  $M$ .
2. Buat matriks augmentasi dengan matriks  $I$  (identitas) dari matriks  $M$ , dinotasikan dengan  $A$  (matriks  $M$  pada sisi kiri dan matriks  $I$  pada sisi kanan).
3. Lakukan perulangan  $i$  dari 1 sampai dengan  $n$  baris matriks  $A$ :
  - a. Tetapkan  $A_{i,i}$  sebagai *pivot*.
  - b. Lakukan normalisasi, yaitu membagi setiap elemen baris  $i$  dengan *pivot*.
  - c. Lakukan perulangan  $j$  dari 1 sampai dengan  $n$  baris matriks  $A$ :  
Jika  $j$  tidak sama dengan  $i$ , maka lakukan:
    - i. Tetapkan  $A_{j,i}$  sebagai *dummy*.
    - ii. Lakukan perulangan  $k$  dari 1 sampai dengan  $m$  kolom matriks  $A$ :  

$$A_{j,k} = A_{j,k} - \text{dummy} \times A_{i,k}$$
4. Invers dari matriks  $M$  berada pada sisi kanan matriks  $A$  dan matriks identitas berada pada sisi kiri.

Ilustrasi invers matriks dengan metode ini dapat dilihat pada gambar 2.10 sampai dengan gambar 2.14 berikut.

$$M = \begin{bmatrix} 3 & -0,1 & -0,2 \\ 0,1 & 7 & -0,3 \\ 0,3 & -0,2 & 10 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A = \left[ \begin{array}{ccc|ccc} 3 & -0,1 & -0,2 & 1 & 0 & 0 \\ 0,1 & 7 & -0,3 & 0 & 1 & 0 \\ 0,3 & -0,2 & 10 & 0 & 0 & 1 \end{array} \right]$$

**Gambar 2.10** Matriks augmentasi A

$$A = \left[ \begin{array}{ccc|ccc} 1 & -0,0333 & -0,0667 & 0,3333 & 0 & 0 \\ 0 & 7,0033 & -0,2933 & -0,0333 & 1 & 0 \\ 0 & -0,19 & 10,02 & -0,0999 & 0 & 1 \end{array} \right]$$

**Gambar 2.11** Langkah 3 dengan  $i = 1$

$$A = \left[ \begin{array}{ccc|ccc} 1 & 0 & -0,0681 & 0,3332 & 0,0047 & 0 \\ 0 & 1 & -0,0418 & -0,0047 & 0,1422 & 0 \\ 0 & 0 & 10,0121 & -0,1009 & 0,027 & 1 \end{array} \right]$$

**Gambar 2.12** Langkah 3 dengan  $i = 2$

$$A = \left[ \begin{array}{ccc|ccc} 1 & 0 & 1 & 0,3325 & 0,0049 & 0,0068 \\ 0 & 1 & 0 & -0,0052 & 0,1423 & 0,0042 \\ 0 & 0 & 1 & -0,0101 & 0,0027 & 0,0999 \end{array} \right]$$

**Gambar 2.13** Langkah 3 dengan  $i = 3$

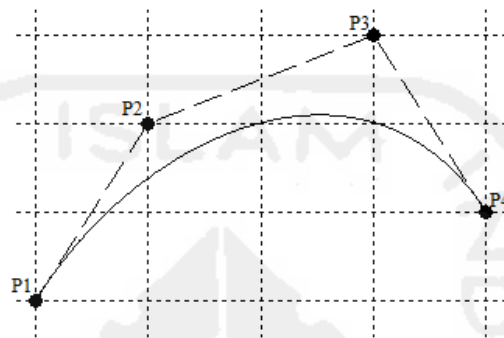
$$M^{-1} = \begin{bmatrix} 0,3325 & 0,0049 & 0,0068 \\ -0,0052 & 0,1423 & 0,0042 \\ -0,0101 & 0,0027 & 0,0999 \end{bmatrix}$$

**Gambar 2.14** Invers dari matriks M

Pada penulisan tugas akhir ini, invers matriks digunakan untuk menyelesaikan persamaan 2.7, yaitu untuk menghitung titik kontrol B-Splines.

## 2.7 Kurva Bézier

Kurva bézier adalah kurva yang diperkenalkan oleh Pierre Bézier pada tahun 1962. Kurva bézier pada awalnya dikembangkan oleh Pierre Bézier untuk merancang bodi mobil Renault. Kurva ini adalah kurva yang sering digunakan pada bidang grafika komputer. Kurva bézier dibuat dengan menyesuaikan beberapa titik kontrol (Marsh, 2005) seperti pada gambar 2.15.



**Gambar 2.15** Kurva bézier dengan 4 titik kontrol

Saat diberikan  $n + 1$  kontrol poin  $(P_0, P_1, \dots, P_n)$ , maka kurva bézier derajat  $n$  akan didefinisikan pada persamaan 2.1 berikut.

$$C(t) = \sum_{i=0}^n P_i B_{in}(t) \quad (2.1)$$

$B_{in}(t)$  merupakan polynomial Bernstein. Persamaan polynomial Bernstien didefinisikan pada persamaan 2.2 sampai dengan 2.4 berikut.

$$B_{in}(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad (2.2)$$

$$t \in [0, 1] \quad (2.3)$$

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \quad (2.4)$$

Koefisien kurva bézier direpresentasikan dengan matriks berikut.



$$koefBezier = \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Persamaan untuk membuat kurva bézier didefinisikan pada persamaan 2.6 berikut.

$$\begin{bmatrix} x1' & x2' & \dots & x101' \\ y1' & y2' & \dots & y101' \end{bmatrix} = \begin{bmatrix} xP1 & xP2 & xP3 & xP4 \\ yP1 & yP2 & yP3 & yP4 \end{bmatrix} \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t^0 \\ t^1 \\ t^2 \\ t^3 \end{bmatrix} \quad (2.6)$$

Dengan  $t = \{0; 0,01; 0,02; \dots; 1\}$  agar menghasilkan kurva bézier yang lebih halus.

## 2.8 Relaxed Cubic Splines dengan Interpolasi

Interpolasi adalah proses untuk mengestimasi nilai diantara beberapa titik data (Chapra & Canale, 1990). Sebuah *cubic splines* adalah *splines* yang dibentuk dari potongan-potongan polinomial orde ketiga dengan turunan kedua yang kontinu (Bartels, Beatty, & Barsky, 1998). Sebuah *cubic splines* dikatakan *relaxed* jika turunan keduanya sama dengan 0 pada setiap ujung.

Metode ini dipilih dikarenakan titik kontrol yang dibutuhkan untuk membuat kurva tepat melewati titik data atau garis kurva. Hal ini berbeda dari metode-metode pembentuk kurva yang lain yang titik kontrolnya berada diluar titik data atau garis kurva.

Diberikan titik data  $S_0, S_1, \dots, S_n$  dan kurva *cubic splines* adalah  $P(t)$  untuk  $0 \leq t \leq n$  dengan  $P_i = S_i$ . Pendekatannya adalah dengan menggunakan B-Splines sebagai titik kontrol ( $B_0, B_1, \dots, B_n$ ) untuk mendapatkan titik kontrol kurva bézier untuk setiap potongannya.

Dengan bahasa sederhana, untuk membuat interpolasi *relaxed cubic splines* tahap pertama yang dilakukan setelah mempunyai koordinat batas kurva adalah menghitung koordinat titik kontrol *B-Splines*. Titik kontrol B-Splines yang telah dihitung akan digunakan untuk menghitung titik kontrol kurva bézier untuk setiap

potongan kurva. Tahap selanjutnya ialah menggambar semua kurva bézier dan menggabungkannya sehingga menjadi sebuah kurva *relaxed cubic splines*.

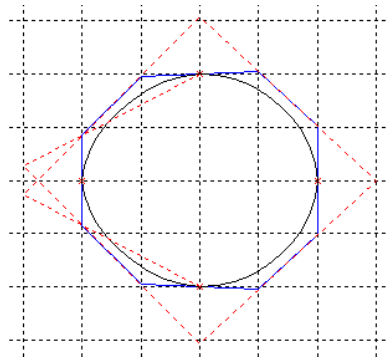
Untuk menghitung titik kontrol B-Splines, persamaannya didefinisikan sebagai berikut (dengan banyaknya titik sama dengan 6).

$$\begin{bmatrix} 4 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} = \begin{bmatrix} 6S_1 - S_0 \\ 6S_2 \\ 6S_3 \\ 6S_4 - S_5 \end{bmatrix} \quad (2.7)$$

Matriks koefisien dari *cubic splines* (yang biasa disebut dengan matriks “141”) untuk membentuk *relaxed cubic splines* akan menjadi sebuah M matriks persegi, dengan  $M = n - 2$ . Persamaan untuk menghitung titik kontrol bézier pada setiap potongannya didefinisikan sebagai berikut.

$$kontrolBezier = \begin{bmatrix} S_i - 1 \\ \frac{2}{3}B_{i-1} + \frac{1}{3}B_i \\ \frac{1}{3}B_{i-1} + \frac{2}{3}B_i \\ S_i \end{bmatrix} \quad (2.8)$$

Untuk memperhalus bentuk kurva tertutup sederhana, pada penulisan tugas akhir ini titik batas kurva sama dengan  $n+2$ . Jika kurva mempunyai 5 titik, yaitu  $P_1, P_2, P_3, P_4$  dan  $P_5$ , dengan  $P_1=P_5$  maka menjadi 7 titik dengan urutan  $P_4, P_1, P_2, P_3, P_4, P_5$  dan  $P_2$ . Ilustrasi *relaxed cubic splines* untuk kurva tertutup sederhana dapat dilihat pada gambar 2.16.



**Gambar 2.16** *Relaxed cubic splines*

## 2.9 Kriteria Pengujian

Pengujian adalah tahap penting dalam pengembangan perangkat lunak. Pada tahap ini akan diukur dan dilihat apakah perangkat lunak yang dikembangkan sesuai dengan perancangan dan kebutuhan.

Pengujian perangkat lunak dilakukan dengan mengikuti *checklist* berdasarkan kualitas karakteristik yang diajukan oleh standar ISO 9126-1 (ISO, 2001). Pengujian dibagi menjadi dua karakteristik, yaitu fungsionalitas dan kegunaan (Dipper, Götze, & Stede, 2004).

### 2.8.1 Karakteristik Fungsionalitas

Pada tahap ini akan dilakukan pengujian pada aspek fungsi atau fitur yang sesuai dengan tujuan perangkat lunak tersebut dikembangkan. Terdapat dua ISO subkarakteristik yang sesuai dengan konteks perangkat lunak atau sistem yang dikembangkan ini, yaitu sebagai berikut.

1. *Suitability* (kecocokan/kesesuaian), mengindikasikan apakah perangkat lunak yang dikembangkan telah menyediakan fungsi yang sesuai.
2. Interoperabilitas, melihat apakah perangkat lunak yang dikembangkan mampu berinteraksi dengan perangkat lunak lainnya.

### 2.8.2 Karakteristik Kegunaan

Selain dari aspek fungsionalitas, perangkat lunak juga perlu diuji apakah pengguna mampu menggunakan perangkat lunak dengan baik. Subkarakteristik yang digunakan untuk menguji adalah sebagai berikut.

1. Operabilitas, mengukur apakah perangkat lunak mudah untuk digunakan.
2. Dokumentasi, mengukur kebutuhan tutorial atau dokumentasi lainnya yang diperlukan atau yang tersedia.
3. *Compliance* (kelengkapan), mengukur apakah perangkat lunak menyediakan fitur pelengkap yang sesuai dan dibutuhkan oleh pengguna.

