

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

BAB II

LANDASAN TEORI

2.1 Visualisasi

Menurut Cormick, dkk, 1997, Visualisasi adalah sebagai berikut:

1. Metode penggunaan komputer untuk mentransformasi simbol menjadi *geometric*.
2. Memungkinkan peneliti mengamati simulasi dan komputasi
3. Memberian cara melihat yang tidak terlihat
4. Memperkaya proses penemuan ilmiah dan mengembangkan pemahaman yang lebih dalam dan tidak diduga
5. Dalam beberapa bidang telah merevolusikan cara ilmuwan meneliti sains.

Visualisasi adalah penggunaan komputer pendukung, penggambaran data visual interaktif untuk memperkuat pengamatan (Shneiderman, 1991).

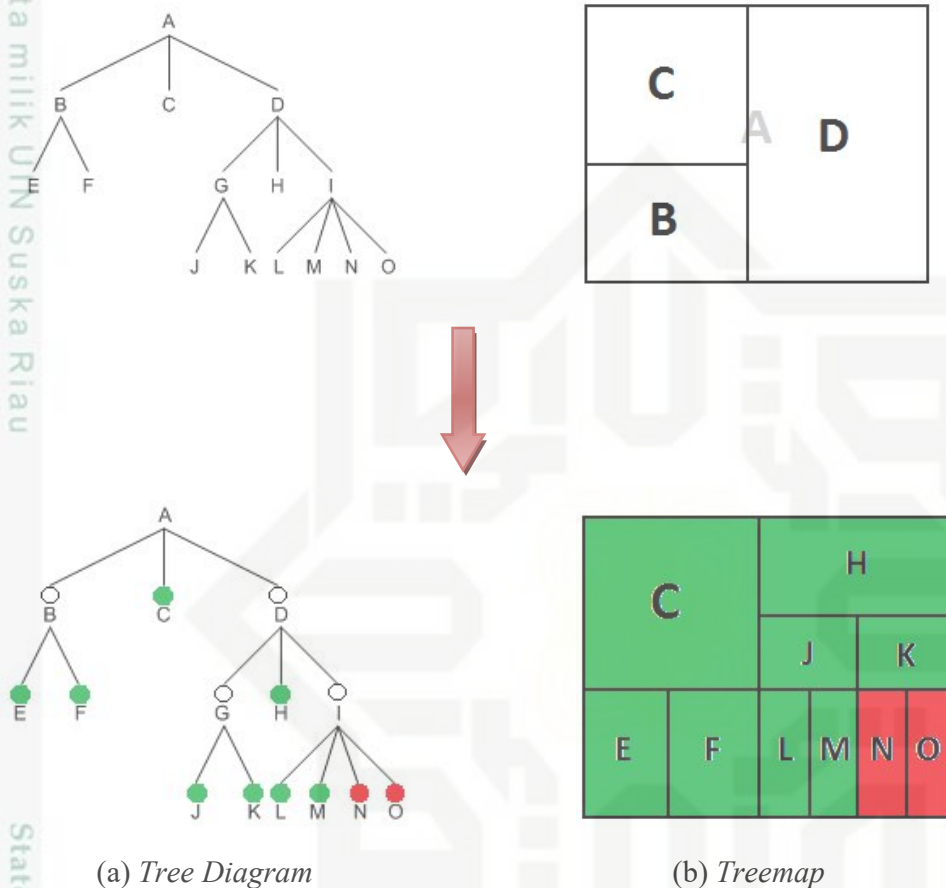
Jadi visualisasi adalah upaya yang dilakukan untuk menyajikan data dan mengubah bentuk data ke dalam bentuk grafis dengan tujuan membantu pengguna dalam menganalisa dan penalaran tentang data dan bukti sehingga informasi dapat sampai ke pengguna secara jelas dan efisien.

Berikut ini prinsip Visualisasi Informasi:

1. Fokus pada konten
2. Perbandingan
3. Integritas
4. Resolusi Tinggi
5. Utilisasi konsep terdahulu dan teruji

Manusia memiliki kemampuan membangun yang baik untuk menganalisa sejumlah besar informasi yang dipresentasi secara visual. Dalam visualisasi terdapat istilah representasi yang berarti pemetaan informasi menjadi format visual.

Jumlah *file* dan direktori dapat ditampilkan secara bersamaan (memiliki *limit* dalam menampilkannya), sehingga seseorang dapat tahu apa yang harus dicarinya.



(a) Tree Diagram

(b) Treemap

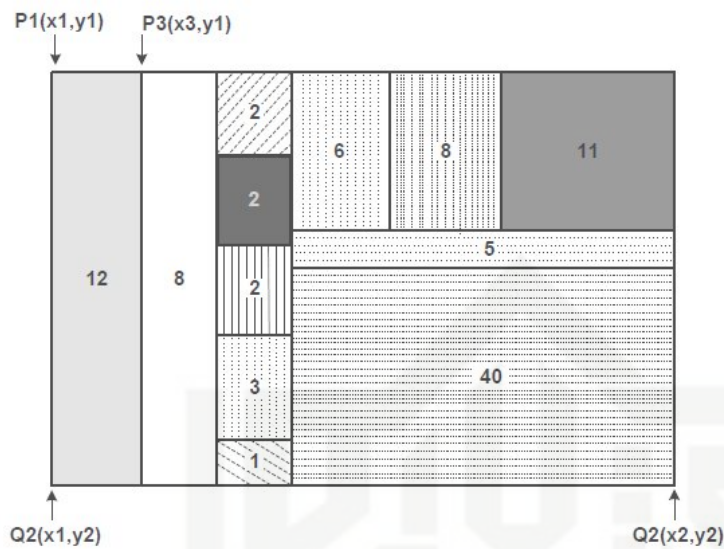
Gambar 2.1 Tree diagram dan Treemap

Ada banyak cara untuk memvisualisasikan struktur tersebut secara efektif. Seperti gambar diatas (a), elemen-elemennya ditampilkan dengan *nodes*, relasinya ditunjukkan dari *link parent* ke *nodes child*-nya. Semakin majunya teknologi, ada banyak teknologi yang disajikan untuk meningkatkan efisiensi dan estetika teknologi, seperti diagram baik dalam 2D dan 3D.

Diagram tersebut sangat efektif untuk *tree* yang kecil, namun biasanya akan mengalami kegagalan ketika lebih dari beberapa ratus elemen yang harus divisualisasikan secara bersamaan. Alasan utama dari *limit*-nya yaitu bahwa *node* dan *link* diagram menggunakan ruang *display* yang tidak efisien, sebagian besar menggunakan *pixel* sebagai *background*-nya.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 2.2 Implementasi Treemap

Area kotak yang didefinisikan dengan koordinat kiri atas dan kanan bawah $P1(x_1,y_1)$, $Q1(x_2,y_2)$. Jumlah sisi yang keluar dari akar menunjukkan jumlah partisi dari daerah $[x_1, y_2]$. *Subtree* paling kiri terdiri dari suatu *fraction* ($Size(child[1]) / Size(root)$) dari jumlah total *bytes* pada akar (*root*). Agar dapat dilihat dengan jelas, maka digunakan warna yang berbeda untuk tiap daerah. Efek dari melihat ribuan kotak kecil adalah seperti papan catur dengan ukuran kotak yang berbeda-beda. Pengkodean yang berwarna dapat menggambarkan besar kecilnya data, data mempunyai warna yang berbeda berdasarkan variabel warna yang telah ditentukan.

Metode *Treemap* mengasumsikan ada sebuah struktur *tree* yang mana tiap titik (*node*) itu terdiri dari *record* dengan direktorinya atau nama file (*nama*), jumlah anak (*jum_anak*), dan suatu *array of pointers* ke level berikutnya (*anak[1..jum_anak]*). Argumen atau parameter dari metode *Treemap* ini adalah:

P,Q: *array* dengan panjang 2 dengan pasangan koordinat (x,y) dari sudut yang berseberangan dari suatu kotak tertentu (asumsikan bahwa P terdiri dari koordinat atas dan Q koordinat bawah).

Warna : menunjukkan warna yang akan digunakan untuk suatu kotak tertentu.

Sedangkan prosedur yang dibutuhkan untuk metode *Treemap* ini adalah:

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Ukuran: Suatu fungsi yang mengembalikan nilai berupa jumlah *byte* pada titik yang ditunjuk oleh parameter. Sebagai alternatif, ukuran dapat dihasilkan terlebih dahulu dan disimpan di tiap titik.

Pemanggilan awal adalah:

$Treemap(akar, P, Q, warna)$

Dimana P dan Q adalah sudut kanan paling atas dan sudut kiri paling bawah dari tampilan. Dengan memberi nilai awal nol pada parameter axis, maka partisi mula-mula adalah secara vertikal. Dalam hal ini diasumsikan bahwa parameter P dan Q adalah *passed by value* : $Treemap(akar, P[0..1], Q[0..1], warna)$

1. *For* $I := 1$ to jum_anak *do*
2. $Lebar := Size(anak[I]) / size(akar) * lebar$
3. $Treemap(anak[I], P, Q, warna)$
4. *Endfor*

2.2.3 Penelitian Terkait *Treemap*

Berikut ini beberapa penelitian terkait yang telah dilakukan sebelumnya:

1. penelitian Aimi Kobayashi, Kazuo Misue dan Jiro Tanaka, pada tahun 2012 yang berjudul *Edge Equalized Treemaps*, penelitian tersebut menyatukan lebar sumbu horizontal masing-masing grafik yang ditandai dengan persegi panjang dengan ukuran *leaf* yang sama. Ketinggian masing-masing *leaf* persegi panjang dalam *Treemap* disamakan sesuai dengan kebutuhan data. Interval skala grafik dalam persegi panjang dapat bersatu, yang berarti seseorang dapat membandingkan grafik hanya dengan melihat gambarnya.
2. Berdasarkan penelitian Abon Chaudhuri dan Hai-Wen Shen pada tahun 2009 yang berjudul *A Self Adaptive TreeMap Based Technique for visualizing Hierarchical Data in 3D*, penelitian tersebut membangun visualisasi hirarki data *Treemap* berbasis 3D yang struktur informasinya dapat digunakan secara luas, visualiasi ini dapat menangani masalah seperti tampilan yang berantakan dan kurang halusny dalam *zooming*. *Treemap* memanfaatkan teknik fleksibilitas yang bertujuan untuk

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

mencegah adanya *occlusion*, yaitu apabila peta terangkat ada beberapa bagian peta yang berada didasar ikut terangkat, ini merupakan masalah yang sering terjadi pada visualisasi 3D

3. Penelitian Chris Muelder dan Kwan-Liu Ma pada tahun 2008 yang berjudul *A Treemap Based Method for Rapid Layout of Large Graphs*, pada penelitian tersebut tata letak grafik dapat ditampilkan secara efektif dan cepat serta konsep tata letak metode *Treemap* sangat fleksibel. Metode ini dapat digunakan sebagai masukan algoritma yang berulang, yaitu dimana dapat mengurangi jumlah iterasi yang dibutuhkan untuk meng-*converge* tata letak optimal yang paling dekat.
4. Penelitian Ben Shneiderman pada tahun 1991 yaitu yang berjudul *Tree Visualization with Tree-maps: A 2-d space-filling approach*, Ben ingin mendapatkan representasi yang lebih baik dari pemanfaatan penyimpanan pada *hardisk* yang dilihat dari perspektif direktori level *multiple* subdirektori dan *files*, seperti di Unix, Macintosh Finder atau Ms-Dos. Visualisasi 2D dalam mengisi ruang dengan metode *Treemap* untuk memvisualisasikan berjalan dengan baik dan cepat. Pewarnaan dan gambaran singkat mengenai ukuran dari *node* yang secara jelas memberikan kemudahan kepada pengguna.
5. Penelitian Yulia pada tahun 2002 yaitu yang berjudul *Visualisasi dan Navigasi Media Penyimpanan Disk dengan menggunakan metode Split Screen System* dengan hasil aplikasi berbasis windows dapat mempermudah pemakaian komputer untuk mengetahui informasi dari file atau folder mana pada media penyimpanan disk yang menggunakan tempat yang besar pertama, kedua, ketiga dan seterusnya.
6. Penelitian Lukman Arie Susanto, dkk pada tahun 2003 yaitu yang berjudul *Implementasi dan Analisis Visualisi Graph pada Graph Statis menggunakan Representasi Visual Treemap* dengan hasil visualisasi treemap efektif karena dapat dibaca dan di pahami dengan menggunakan node yang dipilih terhubung ke node mana saja. Namun, treemap memiliki kelemahan dalam hal waktu untuk melihat informasi yang diinginkan.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2.3 Pengujian *White Box*

Pengujian *White Box* adalah pengujian yang didasarkan pada pengecekan detail perancangan, menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian kedalam beberapa kasus pengujian. Pengujian *White Box* ini bertujuan untuk memastikan kode yang telah dibuat memenuhi persyaratan sesuai yang diharapkan. Salah satu metode pengujian *White Box* yaitu dengan menggunakan *Codeception*. Cara membuat pengujian ini dapat diakses di www.codeception.com. *Codeception* menyediakan PHP unit testing yang cukup modern. *Codeception* dapat menjalankan tes aplikasi dengan cepat. Berikut ini beberapa tahapan pengujian *White Box* dengan menggunakan *Codeception*:

1. Membuat *bootstrap file*

Hal pertama yang dilakukan untuk meng-*install Codeception* adalah mengeksekusi perintah untuk membuat *bootstrap*-nya, ketika perintah ini dijalankan maka *Codeception* akan meng-*generate file* yang dibutuhkan untuk proses *testing*. Perintahnya adalah **php.codecept.phar bootstrap**. Saat perintah itu dijalankan maka akan muncul *folder* bernama *tests*, disinilah *Codeception* meletakkan *file testing* dan Konfigurasi.

2. Konfigurasi

Konfigurasi akan digunakan oleh *Codeception* dalam melakukan pengujian, seperti alamat url yang akan diuji, untuk *acceptance test* maka *file* yang digunakan adalah **acceptance.suite.yml**, buka maka isinya akan seperti:

actor: AcceptanceTester

modules:

enabled:

- **PhpBrowser:**

url: http://localhost/namaprojek

- **\Helper\Acceptance**

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

3. Membuat File *Acceptance Testing*

Langkah selanjutnya adalah membuat file *acceptance test*, dengan perintah **php.codecept.phar generate:cept acceptance login**

Jika berhasil maka akan ada *file* dengan nama **LoginCept.php** dalam *folder acceptance testing*. Di *file* inilah akan dibuat *codingtest*-nya. Berikut ini code untuk skenario login berhasil:

```
<?php
$I = new AcceptanceTester($scenario);
$I->wantTo('Proses Login');
$I->amOnPage('/login.php');
$I->fillField('username','admin');
$I->fillField('password','admin');
$I->click('Submit');
$I->seeInCurrentUrl('/cek_login.php');
$I->amOnPage('/beranda.php');
$I->see('HOME');
```

4. Menjalankan Pengujian

Perintah untuk menjalankan pengujian adalah **codecept.phar run acceptance -steps** maka *Codeception* akan menampilkan *list* pengujian untuk *acceptance test* beserta langkah-langkahnya.