



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

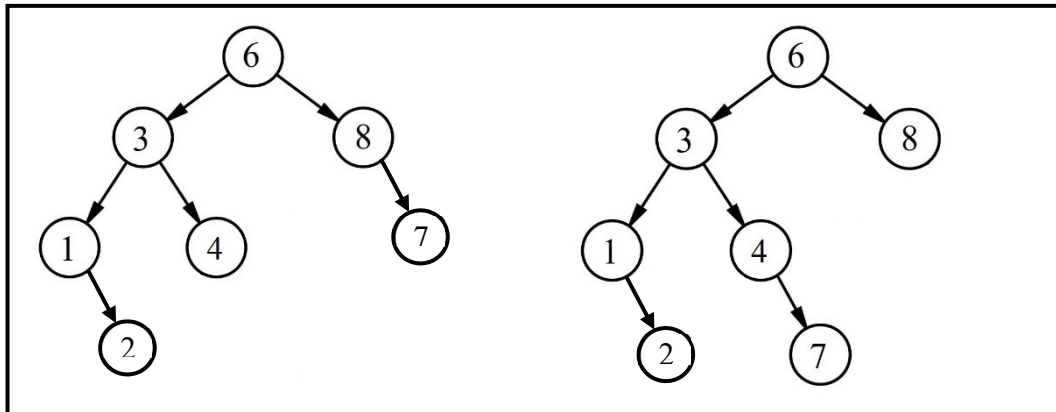
This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Metode Binary Search Tree (BST)

Binary search tree (BST) adalah algoritma struktur data yang mendukung banyak operasi dinamis, seperti *search*, mencari nilai minimum, mencari nilai maksimum, dan *insert* data (Weiss, 2006). Umumnya setiap aplikasi yang menggunakan *binary tree* itu digunakan untuk pencarian dan setiap simpul / *node* memiliki sebuah nilai. Setiap simpul yang terhubung dapat diasumsikan memiliki nilai yang berbeda-beda. Bisa dikatakan sebuah *binary search tree* apabila nilai untuk simpul kiri lebih kecil dibandingkan nilai simpul Root dan nilai simpul kanan lebih besar dari nilai simpul Root. Root dalam *tree* ini adalah sebuah simpul / *node* yang menjadi titik utama sebagai acuan penilaian.



Gambar 2.1 Dua Binary Tree (hanya pohon kiri yang search tree)

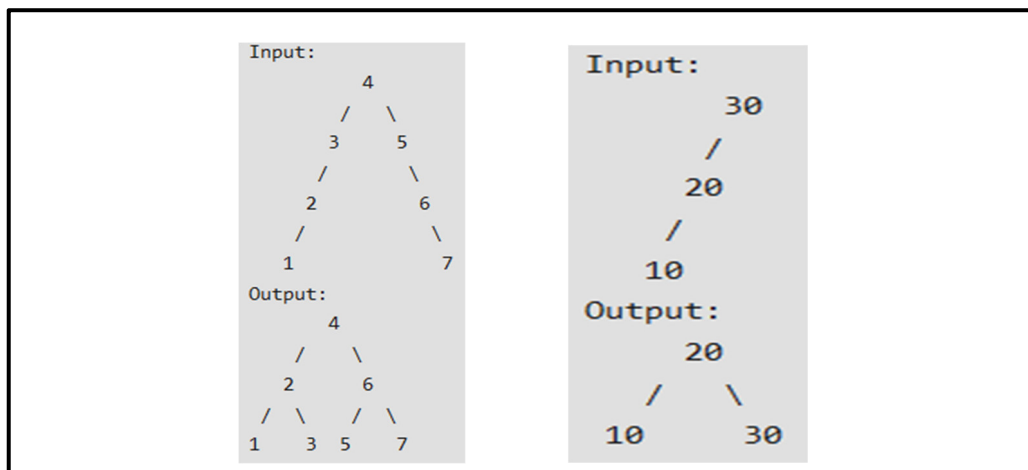
Berikut ini adalah langkah – langkah kerja untuk proses *insert*, yaitu

- Menentukan simpul mana yang akan menjadi Root.
- Membandingkan data yang akan dimasukkan dengan Root, jika lebih kecil rekursif kiri, jika lebih besar rekursif kanan.

- c. Ketika sampai di titik akhir, simpan data jika lebih kecil simpan di kiri dan jika lebih besar simpan di kanan.

Untuk menyeimbangkan *Binary Search Tree* bisa digunakan metode *Balanced Binary Search Tree* agar susunan dari *tree* seimbang antara bagian kiri dan kanan. Solusi yang efisien untuk menyeimbangkan *binary search tree* adalah sebagai berikut :

1. *Traverse BST* dalam *inorder* dan menyimpan hasil dalam *array*. Langkah ini membutuhkan $O(n)$ waktu.
2. Membuat *BST* yang seimbang menggunakan fungsi rekursif, langkah ini juga membutuhkan $O(n)$ waktu saat melintasi setiap elemen atau *node* satu kali dan memproses element membutuhkan $O(1)$ waktu.



Gambar 2.2 Contoh Tree Seimbangkan

2.2 Skala Likert

Skala Likert adalah metode pengukuran yang digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang fenomena sosial (Supriyatna, 2015). Pada pengisian kuisioner, responden akan diberi pilihan untuk masing-masing jawaban di mulai dari sangat setuju, setuju, ragu-ragu, tidak setuju, dan sangat tidak setuju masing-masing akan diberi skor yang berbeda-beda.

Tabel 2.1 Skala Likert

No	Pilihan Jawaban	Skor
1	Sangat Setuju (SS)	5
2	Setuju (S)	4
3	Ragu-ragu (RR)	3
4	Tidak Setuju (TS)	2
5	Sangat Tidak Setuju (STS)	1

Tabel di atas digunakan sebagai skala jawaban pada skala likert. Nilai yang ada pada Tabel tersebut dapat digunakan untuk mendapatkan nilai skor akhir dari pengukuran yang dilakukan. Digunakan rumus sebagai berikut.

$$\text{Skor Kriterion} = \text{nilai skala} \times \text{jumlah responden} \quad \dots(2.1)$$

Setelah mendapatkan hasil perhitungan, maka skor masing-masing setiap perhitungan dijumlahkan untuk mendapatkan total hasilnya. Kemudian dilakukan perhitungan persentase untuk mengetahui jumlah jawaban dari para responden dengan rumus sebagai berikut.

$$p = \frac{f}{n} \times 100 \quad \dots(2.2)$$

Dimana :

p = Persentase

f = Frekuensi dari setiap jawaban

n = Jumlah skor kriterion

100 = Bilangan tetap.

Setelah mendapatkan hasil perhitungan total keseluruhan maka hasil akhir dari persentase diubah ke dalam nilai interval seperti pada Tabel 2.2.

Tabel 2.2 Nilai Interval

Skor	Penilaian
0% - 19,99%	Sangat Tidak Setuju
20% - 39,99%	Tidak Setuju
40% - 59,99%	Ragu –ragu
60% - 79,99%	Setuju
80% - 100%	Sangat Setuju

Dalam penelitian ini, skala likert akan digunakan untuk memberikan penilaian terhadap pertanyaan yang terdapat pada kuisioner.

2.3 Elisitasi

Elisitasi adalah sekumpulan aktifitas yang ditujukan untuk menemukan kebutuhan suatu sistem baru melalui komunikasi dengan pelanggan dan pihak yang memiliki kepentingan dalam pengembangan sistem (Sommerville, 2011). Elisitasi dilakukan melalui tiga tahap yaitu (Bachtiar & Atikah, 2015).

1. Elisitasi tahap I, berisi seluruh rancangan sistem baru yang diusulkan oleh pihak manajemen terkait melalui proses wawancara.
2. Elisitasi tahap II, merupakan hasil pengklasifikasian elisitasi tahap I berdasarkan metode MDI (*Mandatory, Desirable, dan inessential*). Metode MDI (*Mandatory, Desirable, dan inessential*) bertujuan memisahkan antara rancangan sistem yang penting dan harus ada pada sistem baru dengan rancangan yang disanggupi oleh penulis untuk dieksekusi.

- a. “M” pada MDI itu artinya Mandatory (penting). Maksudnya *requirement* tersebut harus ada dan tidak boleh dihilangkan pada saat membuat sistem baru.
 - b. “D” pada MDI itu artinya Desirable. Maksudnya *requirement* tersebut tidak terlalu penting dan boleh dihilangkan. Tetapi jika *requirement* tersebut digunakan dalam pembuatan sistem, akan membuat
 - c. “I” pada MDI itu artinya inessential. Maksudnya bahwa *requirement* tersebut bukanlah bagian dari sistem yang dibahas dan merupakan bagian dari luar sistem.
3. Elisitasi tahap III, merupakan hasil penyusutan elisitasi tahap II dengan cara mengeliminasi semua *requirement* dengan option I pada metode MDI. Selanjutnya, semua *requirement* yang tersisa diklasifikasikan kembali melalui metode TOE, yaitu:
- a. T artinya Technical, maksudnya bagaimana tata cara/teknik pembuatan *requirement* tersebut dalam sistem yang diusulkan.
 - b. O artinya Operational, maksudnya bagaimana tata cara penggunaan *requirement* tersebut dalam sistem yang akan dikembangkan.
 - c. E artinya Economy, maksudnya berapakah biaya yang diperlukan guna membangun *requirement* tersebut didalam sistem.

Metode TOE tersebut dibagi kembali menjadi beberapa option, yaitu:

- a. High (H) : Sulit untuk dikerjakan, karena tehnik pembuatan dan pemakaiannya sulit serta biayanya mahal, sehingga *requirement* tersebut harus dieliminasi.
- b. Middle (M) : Mampu untuk dikerjakan.

- c. Low (L) : Mudah untuk dikerjakan.
4. Final Draft Elisitasi Merupakan hasil akhir yang dicapai dari suatu proses elisitasi yang dapat digunakan sebagai dasar pembuatan suatu sistem yang akan dikembangkan.

2.4 Kepuasan Pengguna

Pengujian aplikasi dapat dilakukan dengan penyebaran kuesioner kepada para pengguna yang telah menggunakan aplikasi. Terdapat beberapa cara untuk mengukur tingkat kepuasan pengguna, salah satunya dengan menggunakan metode *End-user Computing Satisfaction (EUCS)*. Menurut Dewa (2016), pada metode *EUCS* ini mencakup 5 komponen, yaitu isi, keakuratan, bentuk, kemudahan penggunaan dan ketepatan waktu. Berikut adalah penjelasan untuk tiap komponen.

1. Isi (*Content*), untuk mengukur kepuasan dari pengguna dapat ditinjau dari isi suatu sistem. Isi sistem umumnya berupa fungsi dan modul yang dapat digunakan oleh pengguna sistem dan juga informasi yang dihasilkan oleh sistem.
2. Akurasi (*Accuracy*), untuk mengukur kepuasan pengguna dari sisi keakuratan data ketika sistem menerima input kemudian mengolahnya menjadi informasi.
3. Bentuk (*Format*), untuk mengukur kepuasan pengguna dari sisi tampilan dan estetika antarmuka sistem.
4. Kemudahan pengguna (*Ease of Use*), untuk mengukur kepuasan pengguna dari sisi kemudahan pengguna atau user friendly dalam menggunakan sistem, seperti proses memasukkan data, mengolah data, dan mencari informasi yang dibutuhkan.

5. Ketepatan waktu (*timeliness*), untuk mengukur kepuasan pengguna dari sisi ketepatan waktu sistem dalam menyajikan atau menyediakan data dan informasi yang dibutuhkan oleh pengguna.

Dengan adanya pengukuran kepuasan pengguna, maka dapat dibentuk beberapa pertanyaan yang digunakan pada kuesioner untuk mengetahui tingkat kepuasan pengguna yang menggunakan aplikasi MyDaun dan registrasi susun ini. Skala Likert digunakan untuk memberikan penilaian terhadap pertanyaan yang ada.