Jurnal Widya
Volume 3, Nomor 2,Oktober 2022: halaman 331-343
https://jurnal.amikwidyaloka.ac.id/index.php/awl
jurnal@amikwidyaloka.ac.id / editor.jurnalwidya@gmail.com

P-ISSN: 2746-5411
E-ISSN: 2807-5528

Implementasi Algoritma Heapsort dalam Game Pembelajaran Algoritma Sorting

[1]Fenisa Lourence Tobing, [2]Fenina Adline Twince Tobing, [3]Jimmy Peranginangin

[1]Program Studi Manajemen Informatika, AMIK Widyalka Medan, Sumatera Utara, Indonesia
[2]Program Studi Informatika, FakultasTeknik dan Informatika, Universitas Multimedia Nusantara, Tangerang, Banten, Indonesia
[3]Program Studi Teknik Informatika, Universitas Mandiri Bina Prestasi, Medan, Sumatera Utara, Indonesia

Email : fenishatobing@gmail.com, fenina.tobing@umn.ac.id, Jimmy.mbp@gmail.com

Received: 29-09-2022, **Revised:** 19-10-2022, **Accepted:** 24-10-2022

Abstrak

Pembelajaran algoritma merupakan salah satu bagian terpenting dalam pembelajaran algoritma pemrograman. Untuk mempelajari algoritma Heapsort memerlukan metode pembelajaran yang menarik untuk pelajar, dengan menggunakan permainan kartu untuk pembelajaran algoritma heapsort tersebut supaya meningkatkan motivasi belajar para pelajar dalam mempelajari algoritma sorting terutama heapsort.Dari hasil simulasi yang sudah dilaksanakan, algoritma Heapsort ini telah berfungsi dan dapat diimplementasikan dalam bentuk permainan kartu sebagai bahan pembelajaran materi sorting dalam pemrograman. Dengan diimplementasikannya rancangan algoritma ini, pembelajaran pemrograman pada materi sorting dapat menjadi lebih mudah dan menarik bagi pelajar yang akan mendalaminya.

Kata Kunci: *Algoritma, Heap Sort, Game, Pembelajaran, Sorting*

Abstract

Learning algorithms is one of the most important parts in learning programming algorithms. To learn the Heapsort algorithm requires an interesting learning method for students, by using card games for learning the Heapsort algorithm in order to increase students' learning motivation in learning the sorting algorithm, especially heapsort. From the simulation results that have been carried out, the Heapsort algorithm has functioned and can be implemented in the form of a card game as a learning material for sorting programming in programming. With the implementation of this algorithm design, programming learning on sorting material can be easier and more interesting for students who will explore it.

Keywords—components : *Algorithm, Heap Sort, Game, Learning, Sorting*

Jurnal Widya
Volume 3, Nomor 2,Oktober 2022: halaman 331-343
https://jurnal.amikwidyaloka.ac.id/index.php/awl
jurnal@amikwidyaloka.ac.id / editor.jurnalwidya@gmail.com
P-ISSN: 2746-5411
E-ISSN: 2807-5528

1 Pendahuluan

Pemrograman merupakan sebuah proses pengambilan sebuah algoritma dan mengubah algoritma tersebut menjadi suatu bahasa pemrograman yang dapat dijalankan oleh komputer. Meskipun terdapat banyak bahasa pemrograman dan banyak tipe komputer, langkah terpenting pertama adalah mempunyai sebuah solusi. Tanpa algoritma program tidak ada program (Miller dan David, 2013).

Banyak pelajar Teknik Informatika yang baru pertama kali belajar pemrograman mengalami kesusahan untuk mempelajari algoritma (Jenkins, 2002). Pemrograman tidak mudah untuk dipelajari karena membutuhkan pemahaman yang benar terhadap konsep-konsep abstrak. Banyak pelajar yang kesulitan belajar pemrograman karena kekurangan bahan pembelajaran dan instruksi personal, yang mengakibatkan tingkat *drop-out* yang tinggi (Lahtinen dkk, 2005).

Dari permasalahan tersebut, ada salah satu cara untuk menyelesaikan permasalahan ini adalah dengan mengganti metode pembelajarannya. Metode yang dapat dilakukan adalah membuat sebuah proses pembelajaran yang *fun* untuk pelajar, misalnya dengan membuat proses pembelajaran tersebut menjadi suatu permainan, atau disebut juga *learning through playing* (Roussou, 2004).

Implementasi pembelajaran algoritma sorting, terutama HeapSort, dapat dibuat menarik dengan dijadikan sebuah permaianan kartu. Dengan ini diharapkan agar pembelajaran pemrograman dapat menjadi lebih menarik dan lebih mudah bagi pelajar yang akan mendalami materi bersangkutan.

2 Tinjauan Literatur

A. Algoritma HeapSort

HeapSort adalah struktur data yang berbentuk pohon yang memenuhi sifat-sifat heap yaitu jika B adalah anak dari A, maka nilai yang tersimpan di simpul A lebih besar atau sama dengan nilai yang tersimpan di simpul B. Hal ini mengakibatkan elemen dengan nilai terbesar selalu berada pada posisi akar, dan heap ini disebut max heap (bila perbandingannya diterbalikkan, yaitu elemen terkecilnya selalu berada di simpul akar, heap ini disebut adalah min heap). Karena itulah, heap biasa dipakai untuk mengimplementasikan priority queue.

Operasi-operasi yang digunakan untuk heap adalah :

1. Delete-max atau delete-min: menghapus simpul akar dari sebuah max- atau minheap.
2. Increase-key atau decrease-key : mengubah nilai yang tersimpan di suatu simpul.
3. Insert: menambahkan sebuah nilai ke dalam heap.
4. Merge: menggabungkan dua heap untuk membentuk sebuah heap baru yang berisi semua elemen pembentuk heap tersebut.

B. PengurutanHeapSort

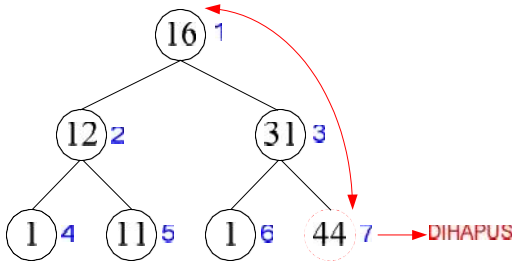
Pengurutan dengan *heap-sort* mengambil data dari pohon *heap* satu per satu. Data yang diambil adalah data pada *node* 1, data pada *node* 1 kemudian ditukarkan dengan data pada *node* terakhir dan *node* terakhirdihapus.

1. Ambil angka ‘44’ pada *node* 1, sehingga deretan angka menjadi:

[12, 31, 1, 11, 1, 16], 44

Tukar *node* 1 dengan *node* 7 (*node* terakhir) dan hapus *node* terakhir.

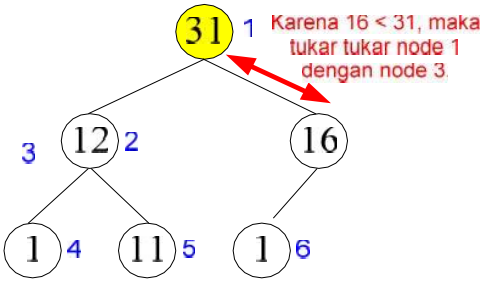
Jurnal Widya
Volume 3, Nomor 2,Oktober 2022: halaman 331-343
https://jurnal.amikwidyaloka.ac.id/index.php/awl
jurnal@amikwidyaloka.ac.id / editor.jurnalwidya@gmail.com
P-ISSN: 2746-5411
E-ISSN: 2807-5528



Gambar 1. Proses penukaran angka pada *node* 1 dan *node* 7 dihapus

Kemudian lakukan proses Heapify(A, 1).

Prosedur Heapify(A,1)



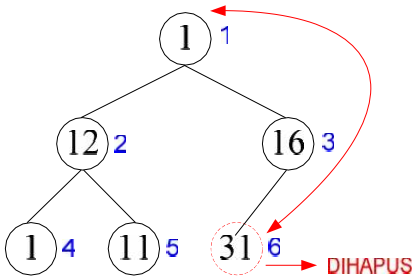
Gambar 2. Proses penukaranangka pada *node* 1 dengan *node*3

Proses Heapify berlanjut pada *node* 3 (*node* yang ditukar dengan *node* 1), tetapi karena *node* 3 memiliki data terbesar dibandingkan dengan *node* anaknya, maka struktur pohon tidak berubah dan prosedur Heapify berakhir.

2. Ambil angka '31' pada *node* 1, sehingga deretan angka menjadi:

[12, 1, 11, 1, 16], 31, 44

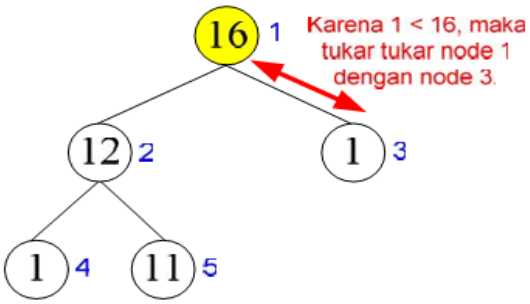
Tukar *node* 1 dengan *node* 6 (*node* terakhir) dan hapus *node* terakhir.



Gambar 3. Proses penukaran angka pada *node* 1 dan *node* 6 dihapus

Jurnal Widya
Volume 3, Nomor 2,Oktober 2022: halaman 331-343
https://jurnal.amikwidyaloka.ac.id/index.php/awl
jurnal@amikwidyaloka.ac.id / editor.jurnalwidya@gmail.com
P-ISSN: 2746-5411
E-ISSN: 2807-5528

Kemudian lakukan proses Heapify(A, 1).



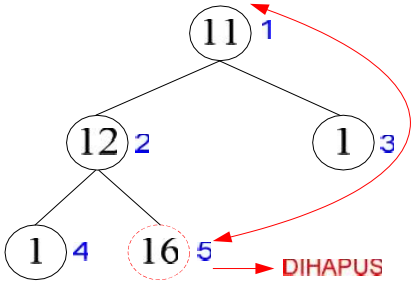
Prosedur Heapify(A,1)

Proses Heapify berlanjut pada *node* 3 (*node* yang ditukar dengan *node* 1), tetapi karena *node* 3 tidak memiliki anak, maka struktur pohon tidak berubah dan prosedur Heapify berakhir.

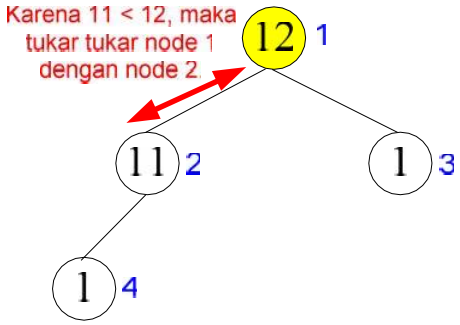
3. Ambilangka ‘16’ pada node 1, sehinggaderetanangkamenjadi:

[12, 1, 11, 1], 16, 31, 44

Tukar node 1 dengan node 5 (node terakhir) dan hapus node terakhir.



Gambar 5. Proses penukaran angka pada *node* 1 dengan *node* 5 dihapus



Gambar 6. Proses penukaran angka pada *node* 1 dengan *node* 2

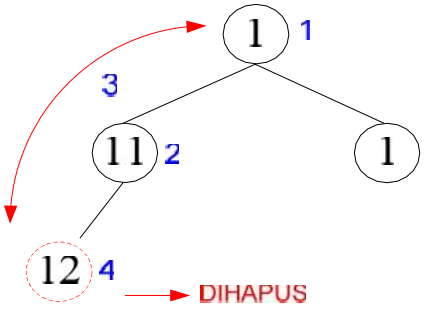
Proses Heapify berlanjut pada node 2 (node yang ditukar dengan node 1), tetapi karena node 2 tidak memiliki anak, maka struktur pohon tidak berubah dan prosedur Heapify berakhir.

Jurnal Widya
Volume 3, Nomor 2,Oktober 2022: halaman 331-343
https://jurnal.amikwidyaloka.ac.id/index.php/awl
jurnal@amikwidyaloka.ac.id / editor.jurnalwidya@gmail.com
P-ISSN: 2746-5411
E-ISSN: 2807-5528

4. Ambil angka ‘12’ pada node 1, sehingga deretan angka menjadi:

[1, 11, 1], 12, 16, 31, 44

Tukar node 1 dengan node 4 (node terakhir) dan hapus node terakhir.

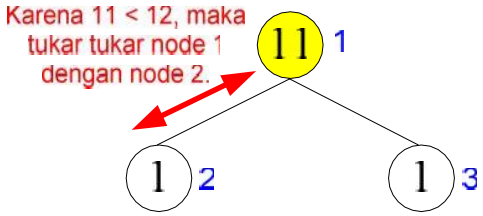


Gambar 7. Proses penukaran angka pada *node* 1 dan *node* 4 dihapus

1

Kemudian lakukan proses Heapify(A, 1)lagi.

Prosedur Heapify(A,1).



Gambar 8. Proses penukaran angka pada *node* 1 dengan *node* 2

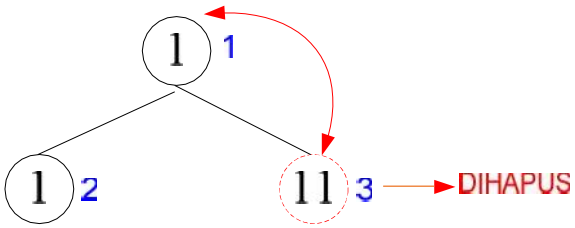
Proses Heapify berlanjut pada node 2 (node yang ditukar dengan node 1), tetapi karena node 2 tidak memiliki anak, maka struktur pohon tidak berubah dan prosedur Heapify berakhir.

5. Ambil angka ‘11’ pada node 1, sehingga deretan angka menjadi:

[1, 1], 11, 12, 16, 31, 44

Tukar node 1 dengan node 3 (node terakhir) dan hapus node terakhir.

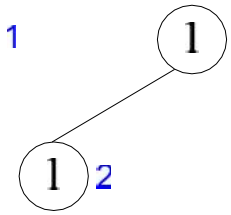
Jurnal Widya
Volume 3, Nomor 2,Oktober 2022: halaman 331-343
https://jurnal.amikwidyaloka.ac.id/index.php/awl
jurnal@amikwidyaloka.ac.id / editor.jurnalwidya@gmail.com
P-ISSN: 2746-5411
E-ISSN: 2807-5528



Gambar 9. Proses penukaran angka pada *node* 1 dan *node* 3 dihapus

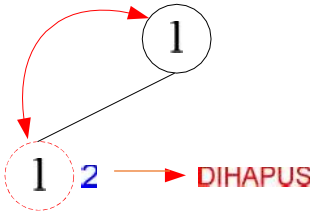
Kemudian lakukan proses Heapify (A,1) lagi.

Prosedur Heapify (A,1)



Gambar 10. Proses penukaran angka pada *node* 1 dengan *node* 2

6. Ambil angka '1' pada node 1, sehingga deretan angka menjadi:
   [1], 1, 11, 12, 16, 31, 44
   Tukar *node* 1 dengan *node* 2 (*node* terakhir) dan hapus *node* terakhir.



Gambar 10. Proses penukaran angka pada node 1 dan node 2 dihapus.

Kemudian lakukan proses Heapify (A, 1).

Prosedur Heapify (A,1)

Gambar 11. Proses pertukaran angka pada node 1 dengan *node 2*

Jurnal Widya
Volume 3, Nomor 2,Oktober 2022: halaman 331-343
https://jurnal.amikwidyaloka.ac.id/index.php/awl
jurnal@amikwidyaloka.ac.id / editor.jurnalwidya@gmail.com
P-ISSN: 2746-5411
E-ISSN: 2807-5528

Proses Heapify tidak mengubah struktur pohon karena node 1 tidak memiliki anak lagi.

7. Ambil angka‘1’ pada node 1, sehingga deretan angka menjadi:

1, 1, 11, 12, 16, 31, 44

Hasil akhir adalah berupa deretan angka terurut secara menaik (ascending). Untuk angka terurut menurun (descending), tempatkan angka yang dikeluarkan dari pohon heap di depan deretan.

3. Hasil dan Pembahasan

Program ini dirancang untuk pembelajaran melalui permainan kartu menggunakan Heapsort. Proses perancangan program bantu pembelajaran dilakukan dalam beberapa tahap melalui algoritma heap sort menurut McMillan. Tahap pertama, hilangkan *node* pada *root* (posisi teratas).Lalu, tahap kedua yaitu pindahkan *node* pada posisi terakhir ke*root*.

A. Perancangan Heapsort *melalui permainan kartu*

Pertama-tama sebelum menguji, penulis menginput data array kartunya sebagai berikut :

Tabel 1. TabelKartu

Nama Kartu	Angka
Queen	12
Tiga	3
King	13
Lima	5
Dua	2
Jack	11
Delapan	8
AS	14
Tujuh	**7**
Sembilan	9
Empat	4
Enam	6
Sepuluh	10

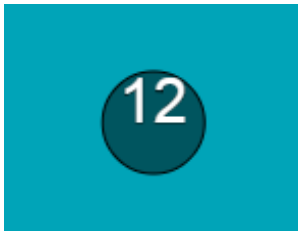
B. Simulai Implementasi

Dari rancangan algoritma yang sudah dijabarkan, dibangun program yang dapat mensimulasikan Heapsort. Berikut merupakan hasil simulasi dari implementasi algoritma Heapsort:

Jurnal Widya
Volume 3, Nomor 2,Oktober 2022: halaman 331-343
https://jurnal.amikwidyaloka.ac.id/index.php/awl
jurnal@amikwidyaloka.ac.id / editor.jurnalwidya@gmail.com
P-ISSN: 2746-5411
E-ISSN: 2807-5528

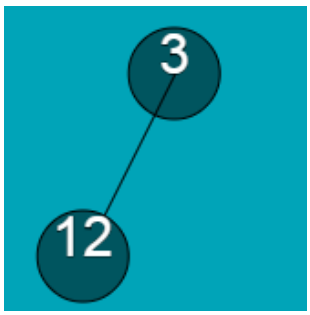
Step 1.

Masukkan [12] sebagai angka pertama.



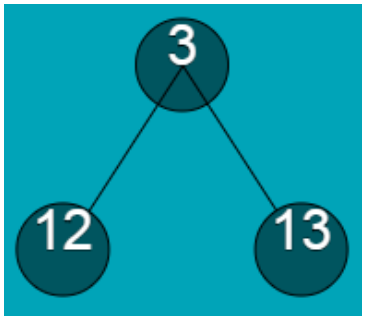
Step 2.

Masukkan [3] sebagai angka kedua. Karena [12] > [3], maka posisi akan ditukar.



Step 3.

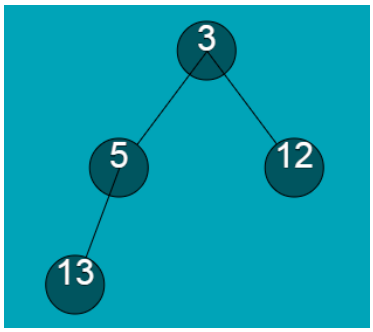
Masukkan [13] sebagai angka berikutnya. Karena [3] < [13], maka tidak terjadi pertukaran.



Step 4.

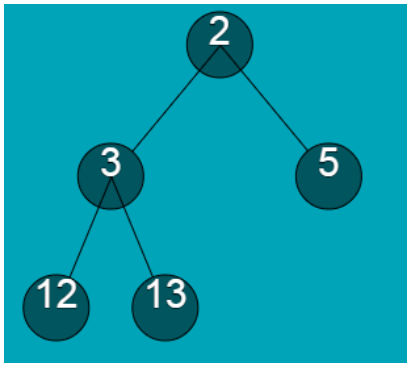
Masukkan [5] sebagai angka berikutnya. Karena [12] > [5], maka terjadi pertukaran. Terjadi juga pertukaran antara [12] dan [13] karena [13] > [12].

Jurnal Widya
Volume 3, Nomor 2,Oktober 2022: halaman 331-343
https://jurnal.amikwidyaloka.ac.id/index.php/awl
jurnal@amikwidyaloka.ac.id / editor.jurnalwidya@gmail.com
P-ISSN: 2746-5411
E-ISSN: 2807-5528



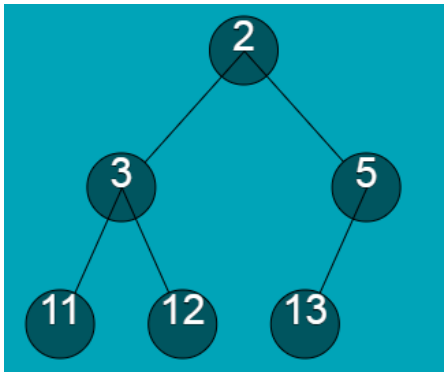
Step 5.

Masukkan [2] sebagai angka berikutnya. Terjadi pertukaran sehingga terbentuk pohon sebagai berikut:



Step 6.

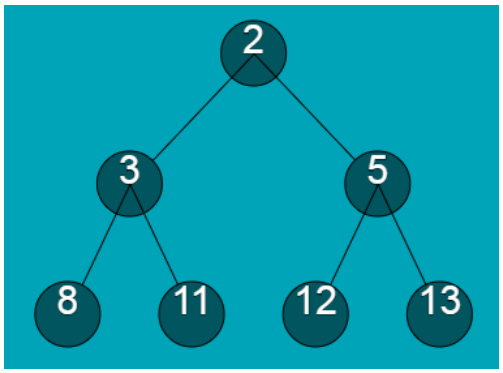
Masukkan [11] sebagai angka berikutnya. Terjadi pertukaran sehingga terbentuk pohon sebagai berikut:



Step 7.

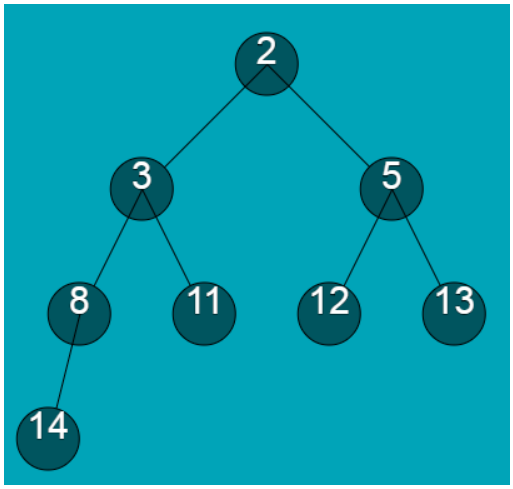
Masukkan [8] sebagai angka berikutnya. Terjadi pertukaran sehingga terbentuk pohon sebagai berikut:

Jurnal Widya
Volume 3, Nomor 2,Oktober 2022: halaman 331-343
https://jurnal.amikwidyaloka.ac.id/index.php/awl
jurnal@amikwidyaloka.ac.id / editor.jurnalwidya@gmail.com
P-ISSN: 2746-5411
E-ISSN: 2807-5528



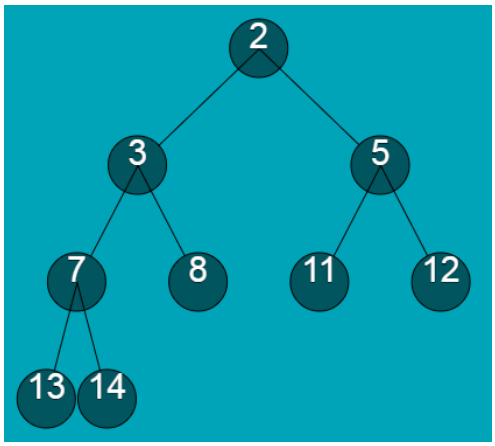
Step 8.

Masukkan [14] sebagai angka berikutnya. Terjadi pertukaran sehingga terbentuk pohon sebagai berikut:



Step 9.

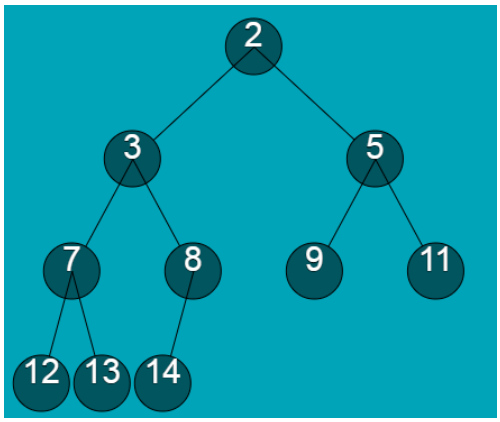
Masukkan [7] sebagai angka berikutnya. Terjadi pertukaran sehingga terbentuk pohon sebagai berikut:



Jurnal Widya
Volume 3, Nomor 2,Oktober 2022: halaman 331-343
https://jurnal.amikwidyaloka.ac.id/index.php/awl
jurnal@amikwidyaloka.ac.id / editor.jurnalwidya@gmail.com
P-ISSN: 2746-5411
E-ISSN: 2807-5528

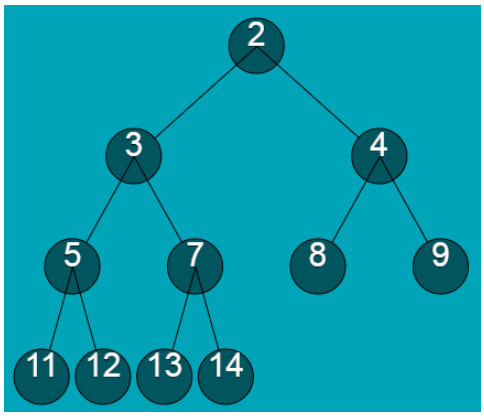
Step 10.

Masukkan [9] sebagai angka berikutnya. Terjadi pertukaran sehingga terbentuk pohon sebagai berikut:



Step 11.

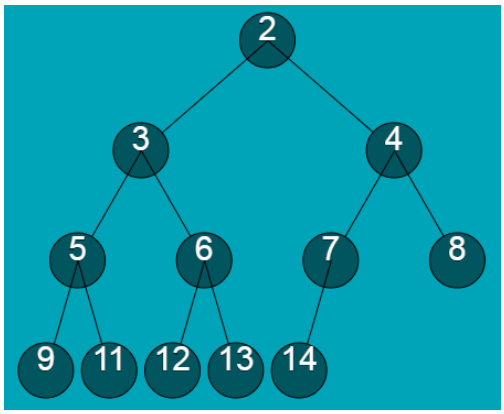
Masukkan [4] sebagai angka berikutnya. Terjadi pertukaran sehingga terbentuk pohon sebagai berikut:



Step 12.

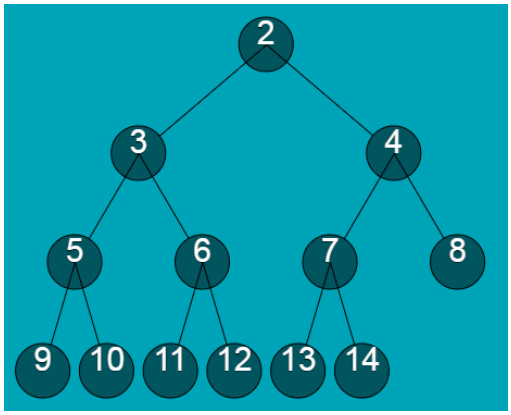
Masukkan [6] sebagai angka berikutnya. Terjadi pertukaran sehingga terbentuk pohon sebagai berikut:

Jurnal Widya
Volume 3, Nomor 2,Oktober 2022: halaman 331-343
https://jurnal.amikwidyaloka.ac.id/index.php/awl
jurnal@amikwidyaloka.ac.id / editor.jurnalwidya@gmail.com
P-ISSN: 2746-5411
E-ISSN: 2807-5528



Step 13.

Masukkan [10] sebagai angka terakhir. Terjadi pertukaran sehingga terbentuk pohon sebagai berikut:



Setelah melalui proses tersebut, array data input yang pada awalnya terisi seperti berikut:

[12], [3], [13], [5], [2], [11], [8], [14], [7], [9], [4], [6], [10]

telah tersusun berurutan sebagai berikut:

[2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]

Dengan ini, simulasi dapat dinyatakan berhasil.

C. Kesimpulan

Algoritma pengurutan HeapSort yang terdapat dalam perangkat lunak dapat dikembangkan dengan class object oriented sehingga dapat dipisahkan secara independen dan dapat digunakan untuk membangun perangkat lunak lain yang membutuhkan algoritma pengurutan.

Dari hasil simulasi yang sudah dilaksanakan, dapat disimpulkan bahwa rancangan algoritma Heapsort ini telah berfungsi seperti yang dimaksudkan. Dengan demikian, rancangan algoritma ini dapat diimplementasikan dalam bentuk permainan kartu sebagai bahan pembelajaran materi sorting dalam pemrograman. Dengan diimplementasikannya rancangan algoritma ini, pembelajaran pemrograman pada materi sorting dapat menjadi lebih mudah dan menarik bagi pelajar yang akan mendalaminya.

Jurnal Widya
Volume 3, Nomor 2,Oktober 2022: halaman 331-343
https://jurnal.amikwidyaloka.ac.id/index.php/awl
jurnal@amikwidyaloka.ac.id / editor.jurnalwidya@gmail.com

P-ISSN: 2746-5411
E-ISSN: 2807-5528

Referensi

[1] Bhalchandra Parag & Deshmukh Nilesh, A Comprehensive Note on Complexity Issues in Sorting Algorithms, School of Computational Sciences, Swami Ramanand Teerth Marathwada University, Nanded, MS

[2] Firdi Mulia, Penerapan Pohon Dalam Heap Sort, ITB Bandung

[3] *https://en.wikipedia.org/wiki/Heapsort*

[4] Tobing, F. A. T., & Tambunan, J. R. (2020). Analisis Perbandingan Efisiensi Algoritma Brute Force dan Divide and Conquer dalam Proses Pengurutan Angka. *Ultimatics: Jurnal Teknik Informatika*, *12*(1), 52-58.

[5] Schaffer, R., & Sedgewick, R. (1993). The analysis of heapsort. *Journal of Algorithms*, *15*(1), 76-100.

[6] Battistella, P. E., WANGENHEIM, C. G. V., WANGENHEIM, A. V., & Martina, J. E. (2017). Design and large-scale evaluation of educational games for teaching sorting algorithms. *Informatics in Education*, *16*(2), 141-164.

[7] A. J. Irawan, F. A. T. Tobing and E. E. Surbakti, "Implementation of Gamification Octalysis Method at Design and Build a React Native Framework Learning Application," *2021 6th International Conference on New Media Studies (CONMEDIA)*, 2021, pp. 118-123

[8] Ali, H., Nawaz, H., & Maitlo, A. (2021). Performance Analysis of Heap Sort and Insertion Sort Algorithm. *International Journal*, *9*(5).

[9] H. Stephen, Implementasi Fungsi Rekursif Dalam Algoritma dan Perbandingannya dengan Fungsi Iteratif, Jurusan Teknik Informatika, Institut Teknologi Bandung, Bandung, 2008

[10] Wikimedia Foundation. "AlgoritmaPencarianStrting". Wikipedia.org. *https://id.wikipedia.org/wiki/Algoritme_pencarian_string* (accessed 17 November 2021)