Hello World
Jurnal Ilmu Komputer
https://jurnal.ilmubersama.com/index.php/hello_world

Sistem Informasi

Implementasi Algoritma Heap Sort pada Aplikasi E-Raport Siswa MTs Swasta Aljamiatul Amaliyah Desa Bandar Selamat Labuhanbatu Utara

Mindi Wahyuni, Khairuddin Nasution

Fakultas Teknik, Program Studi Informatika, Universitas Islam Sumatera Utara, Medan, Indonesia

INFORMASI ARTIKEL

Diterima Redaksi: 29 Oktober 2022
Revisi Akhir: 10 November 2022
Diterbitkan *Online*: 23 November 2022

KATA KUNCI

Aplikasi Berbasis Web, Heap Sort

KORESPONDENSI

Phone: +62 82277329550

E-mail: mindiwahyuni03@gmail.com

A B S T R A K

Pendidikan merupakan salah satu aspek yang penting dalam meningkatkan taraf kehidupan bangsa dan negara, serta merupakan salah satu tolak ukur kemajuan dari suatu bangsa. MTS S Aljamiatul Amaliyah merupakan salah satu lembaga pendidikan swasta di Kecamatan Aek Kuo yang menyelenggarakan pendidikan tingkat sekolah menengah pertama. Namun dalam layanan akademik disekolah tersebut masih dilakukan secara manual diantaranya pengolahan nilai siswa yaitu dengan cara masing-masing guru mata pelajaran menulis nilai siswa kedalam suatu lembaran kertas atau raport. Selain itu, tingkat masalah yang ditimbulkan sistem ini sangat banyak. Diantaranya adalah data-data nilai siswa hilang, raport basah akibat keteledoran sebagian siswa dan faktor alam lainnya yang menyebabkan raport rusak. Tujuan dari penelitian ini adalah untuk membangun aplikasi e-raport dengan mengimplementasikan algoritma *heap sort max-heap* sebagai metode pengurutan yang diterapkan pada nilai rata-rata semester siswa. Dari 15 data yang diuji coba [87.5, 92.47, 91.3, 84.37, 85.70, 87.30, 86.20, 83.90, 87.03, 85.73, 85.47, 85.23, 85.90, 85.23, 84.67] diperoleh hasil akhir [83.90, 84.37, 84.67, 85.13, 85.23, 85.61, 85.70, 85.73, 85.90, 86.20, 87.03, 87.30, 87.50, 91.30, 92.77]. Hasil penelitian menunjukkan bahwa implemetasi algoritma *heap sort max-heap* menghasilkan pengurutan data secara menaik yang berjalan dengan baik.

PENDAHULUAN

Memasuki zaman globalisasi seperti sekarang ini, membuat banyak orang berpikir untuk menggunakan segala sesuatu yang berdasarkan pada hal-hal elektronik. Alasan utamanya adalah dapat membantu dan meningkatkan kinerja. Setiap perusahaan, instansi, dan segala bidang pekerjaan pada umumnya telah menggunakan aplikasi dengan dasar elektronik. Tujuannya adalah agar dapat mengolah data dengan mudah dan cepat. Tidak ketinggalan dalam bidang pendidikanpun sering menggunakan aplikasi perangkat lunak untuk membantu kinerja para pengajar.

Raport adalah salah satu hal yang tidak dapat dilepaskan dari instansi yang bergerak di bidang pendidikan. Untuk proses pembuatan atau pengerjaan raport bermacam-macam, mulai dari proses yang manual sampai dengan proses yang menggunakan perangkat lunak. Tetapi sangat disayangkan, di Indonesia banyak sekolah yang belum menggunakan atau memiliki perangkat lunak untuk pembuatan dan, distribusi raport siswa. Sehingga dibutuhkan perangkat lunak yang dapat membantu menyelesaikan masalah tersebut.

Mts S Aljamiatul Amaliyah Bandar Selamat yang terletak di Labuhanbatu Utara merupakan salah satu sekolah yang sangat penting dalam mencerdaskan kehidupan bangsa di daerah tersebut. Saat ini penyusunan raport di Mts S Aljamiatul Amaliyah Bandar Selamat masih manual, yaitu dengan cara menulis nilai-nilai siswa kedalam raport satu persatu oleh wali kelas. Oleh karena itu tidak mudah dalam menentukan nilai siswa yang terkecil hingga terbesar. Salah satu sistem yang dapat membantu dalam mencari solusi yang sesuai dari sekian banyak kombinasi yang kompleks tanpa harus dilakukan secara manual adalah algoritma *heap sort*.

https://doi.org/10.56211/helloworld.v1i3.144

Heap sort adalah sebuah metode *sorting* (pengurutan) angka pada sebuah *array* dengan cara menyerupai *binary tree*, yaitu dengan cara memvisualisasikan sebuah *array* menjadi sebuah *binary tree* yang nantinya pada *binary tree* tersebut nilai pada masing-masing indeks *array* akan diurutkan.

Beberapa penelitian terkait pada penelitian ini yaitu, penelitian tentang "Simulasi Pengurutan Data dengan Algoritma Heap Sort". Hasil pengurutan yang diinginkan adalah terurut menaik (*ascending*) dan properti *heap* yang dipilih adalah properti dengan data pada *node parent* >= data pada anak sebelah kiri atau anak sebelah kanan [1]. Penelitian lainnya yaitu penelitian tentang "Aplikasi Simulasi Pengurutan Data Menggunakan Algoritma Heap Sort". Membangun aplikasi simulasi pengurutan data dimana data-data yang dikeluarkan merupakan data yang terurut, baik menaik (*ascending*) maupun menurun (*descending*) [2]. Penelitian tentang "Perangkat Lunak Pendukung Pembelajaran Algoritma Heap Sort". Perangkat lunak pendukung pembelajaran algoritma *heap sort* fasilitas untuk menginput data dengan syarat-syarat yang telah ditentukan sebelumnya. Fasilitas input data tersedia pada *button load*, yaitu fasilitas memasukkan angka dari *file* input. *Button* random berfungsi untuk menghasilkan barisan data secara acak. Pada *form* algoritma *heapsort* ini, *user* diminta untuk memilih hasil pengurutan apakah berupa data pengurutan secara menaik (*ascending*) atau berupa data pengurutan secara menurun (*descending*). *User* juga harus memilih sifat *heap* yaitu apakah data pada *node parent* lebih besar atau sama dengan data pada *node* anak sebelah kiri dan anak sebelah kanan, atau sifat *heap* yang *node parent*-nya lebih kecil atau sama dengan data pada anak sebelah kiri dan anak sebelah kanan [3].

Maka, dengan melihat fakta-fakta diatas, jelaslah dibutuhkan suatu sarana yang dapat mengimbangi dan meningkatkan kinerja para guru dan karyawan pada instansi pendidikan khususnya sekolah. Untuk itu sebuah aplikasi raport elektronik (e-raport) diharapkan dapat menjawab kebutuhan ini. Sesuai dengan uraian diatas, perlunya sistem informasi pengolahan nilai pada MTs S Aljamiatul Amaliyah Bandar Selamat.

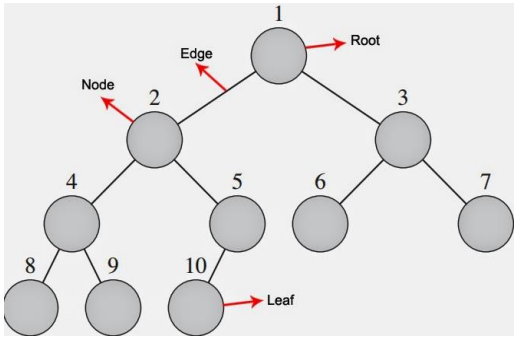
TINJAUAN PUSTAKA

Aplikasi

Menurut Hasan Abdurahman dan Asep Ririh Riswaya (2014), aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan [4]. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya, aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user*.

Algoritma Heap Sort

Menurut Rinaldi (2019), "metode *heap sort* adalah metode dari pengembangan *tree*. Metode *sorting* ini mengurutkan data berdasarkan perbandingan, dan merupakan salah satu algoritma pengurutan tercepat karena mampu mengurutkan data yang sangat banyak dengan waktu yang cepat" [5]. Pada *heap sort* terdapat 3 bagian yaitu *node*, *edge*, dan *leaf* dimana *node* itu adalah setiap indeks yang berada pada *array*, *edge* adalah garis yang menghubungkan tiap *node* dan *leaf* adalah setiap *node* yang tidak memiliki *child node* (*node* turunan). Selain itu juga ada yang bernama *root* yaitu *node* awal pada sebuah *heap*, berikut adalah ilustrasi dari bagian yang dimiliki oleh *heap*:



Gambar 1. Ilustrasi Dari Bagian yang Dimiliki Oleh *Heap Sort*

Heap tree terbagi menjadi 2 jenis yaitu *max-heap* dan *min-heap*, dimana *max-heap* adalah kondisi *heap tree* yang memiliki nilai tertinggi berada di *node root* dan setiap *child node* memiliki nilai yang lebih kecil dari nilai yang dimiliki *parent node*-nya. Sedangkan pada *min-heap* adalah kondisi kebalikan dengan *max-heap*, pada *min-heap* nilai terkecil berada di *node root* dan setiap *child node* memiliki nilai yang lebih besar dari nilai yang dimiliki *parent node*-nya. Pada metode *heap sort* jenis *heap tree* yang digunakan adalah *max-heap*.

Website

Website adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. *Website* merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara animasi sehingga lebih merupakan media informasi yang menarik untuk dikunjungi. Sedangkan menurut Rohi Abdullah (2016) "*Website* merupakan sekumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi dalam bentuk data digital baik berupa *text*, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet" [6].

Javascript

Javascript memiliki karakteristik sebagai berikut [7]:

1. Bahasa pemrograman berjenis *high-level programming* (*syntax* dan struktur mudah dipahami karena menggunakan bahasa yang dimengerti manusia).
2. Bersifat *client-side* (hanya membutuhkan browser untuk menguji Javascript).
3. Berorientasi objek (cocok buat anda yang ingin masuk kedalam konsep pemrograman berbasis objek).
4. Bersifat *loosely typed* (tidak membutuhkan deklarasi variabel terlebih dulu).

PHP

PHP (*Hypertext Preprocessor*) yaitu bahasa pemrograman yang digunakan secara luasuntuk penanganan pembuatan dan pengembangan sebuah situs *web* dan bisa digunakan bersamaan dengan HTML. PHP bersifat *serverside*, artinya Bahasa berbentuk *script* yang disimpan dan dijalankan di komputer *server* (*Web Server*) sedang hasilnya yang dikirimkan ke komputer *client* (*Web Browser*) dalam bentuk *script* HTML (*Hypertext Markup Language*) [8].

MySQL

Menurut Abdullah (2018) SQL berperan sebagai bahasa yang mengatur transaksi data antara aplikasi dengan database sebagai tempat penyimpanan data [9]. Salah satu distro yang menyediakan SQL sebagai bahasa permintaan dan sangat digemari dikalangan programmer *web* adalah MySQL. Penggunaan MySQL biasanya dipadukan dengan menggunakan program aplikasi PHP, karena dengan menggunakan kedua program tersebut diatas telah terbukti akan kehandalannya dalam menangani permintaan data dan juga memiliki kemampuan mendukung *Relational Database Management System* (RDBMS) sehingga dengan kemampuan ini MySQL akan mampu menanganai data-data sebuah perusahaan yang berukuran sangat besar hingga berukuran *giga byte* [10].

METODOLOGI

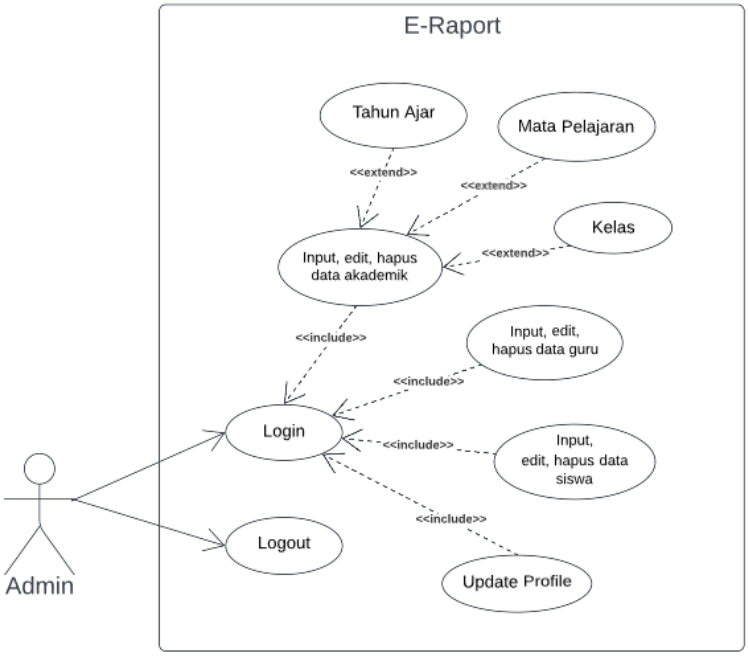
Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan pada penelitian ini, yaitu:

1. Wawancara
   Penelitian ini mengumpulkan data-data yang dibutuhkan melalui wawancara terkait dengan data rapor siswa MTs Aljamiatul Amaliyah Desa Bandar Selamat Labuhanbatu Utara.
2. Studi literatur
   Studi literatur atau studi kepustakaan dilakukan dengan mengambil data dari buku, jurnal ataupun artikel mengenai teori dan langkah-langkah dalam pembuatan aplikasi. Sehingga dapat menggunakannya sebagai dasar landasan teori dan perancangan pembuatan aplikasi e-raport.

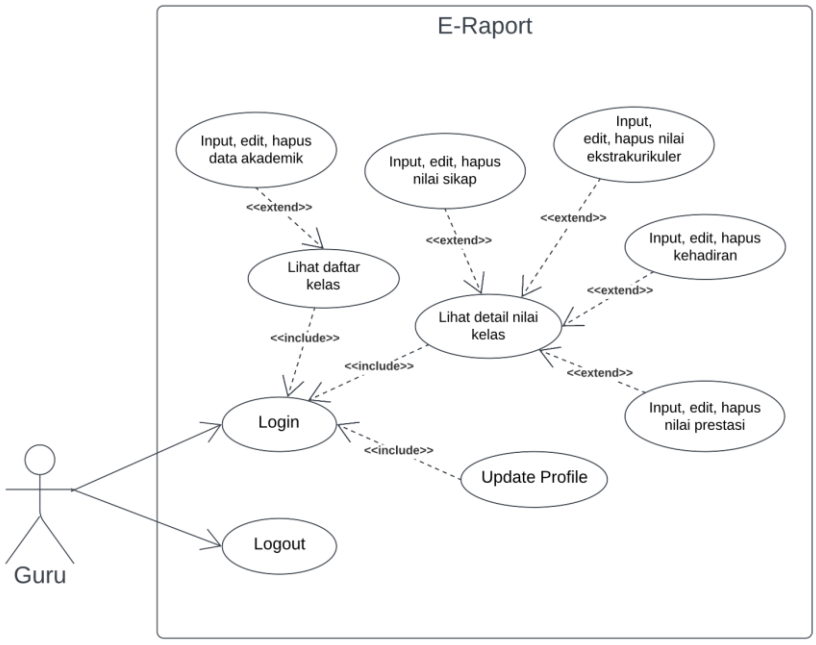
Perancangan Sistem

Use Case Diagram



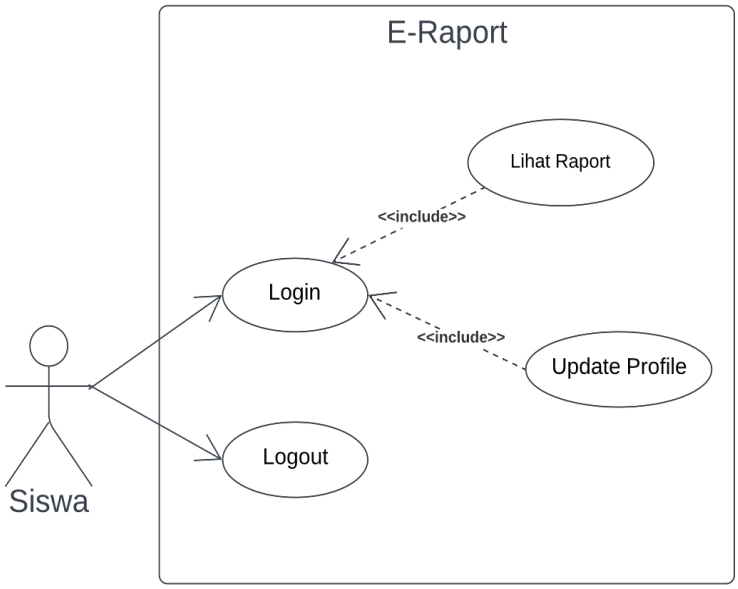
Gambar 2. *Use Case Diagram* Admin

Aktor admin dapat melakukan *login* dan *logout*, pengelolaan data akademik, data guru, data Siswa dan *update profile*. Pengelolaan data yang dapat dilakukan berupa penambahan data, pengeditan data serta penghapusan data.



Gambar 3. *Use Case Diagram* Guru

Aktor guru dapat melihat daftar kelas, detail nilai kelas dan *update profile*. Pada daftar kelas guru dapat melakukan *input*, *edit* dan hapus data akademik. Pada detail nilai kelas, guru dapat melakukan *input*, *edit* dan hapus data ekstrakurikuler, nilai sikap, kehadiran dan nilai prestasi. Dimana, untuk dapat melakukan seluruh aksi tersebut, guru diharuskan melakukan *login* terlebih dahulu. Guru juga dapat melakukan aksi *logout* untuk keluar dari aplikasi.

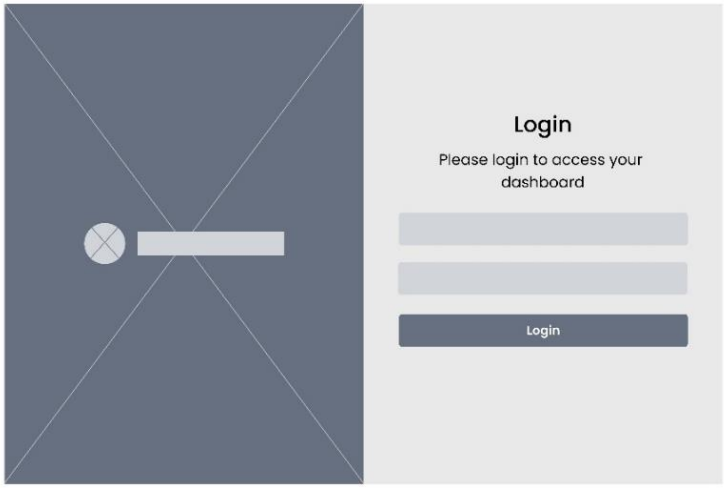


Gambar 4. *Use Case Diagram* Siswa

Aktor siswa dapat melihat raport serta melakukan *update profile* atau pembaharuan *profile*. Namun, untuk dapat melakukan kedua aksi tersebut, Siswa diharuskan melakukan *login* terlebih dahulu. Siswa juga dapat melakukan *logout* untuk keluar dari aplikasi.

Perancangan Antarmuka

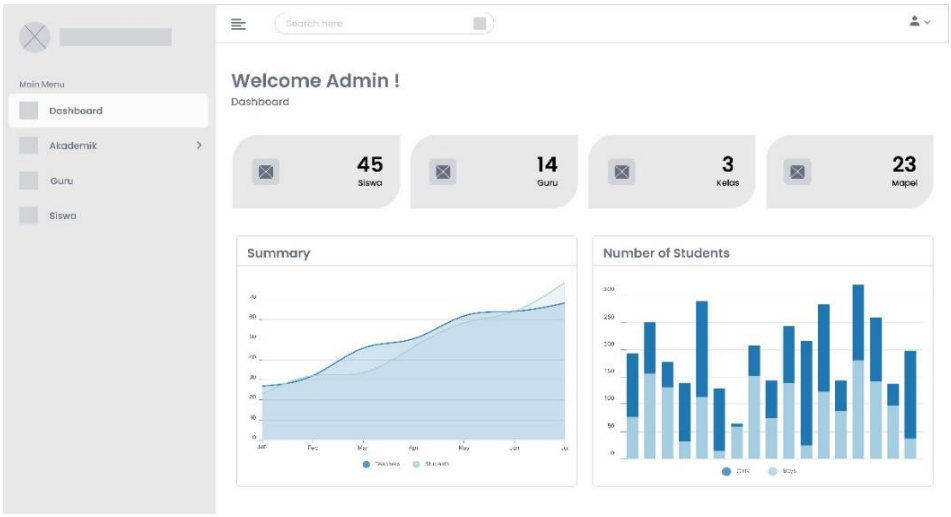
Tampilan perancangan halaman *login* dapat dilihat pada Gambar 5 di bawah ini.



Gambar 5. Perancangan Halaman *Login*

Gambar 5 di atas merupakan tampilan perancangan halaman *login* yang memiliki 2 *field input* yang dapat diisi oleh *user* agar dapat masuk ke aplikasi. Dua *field input* tersebut nantinya merupakan *field input* untuk tipe *email* dan *password*.

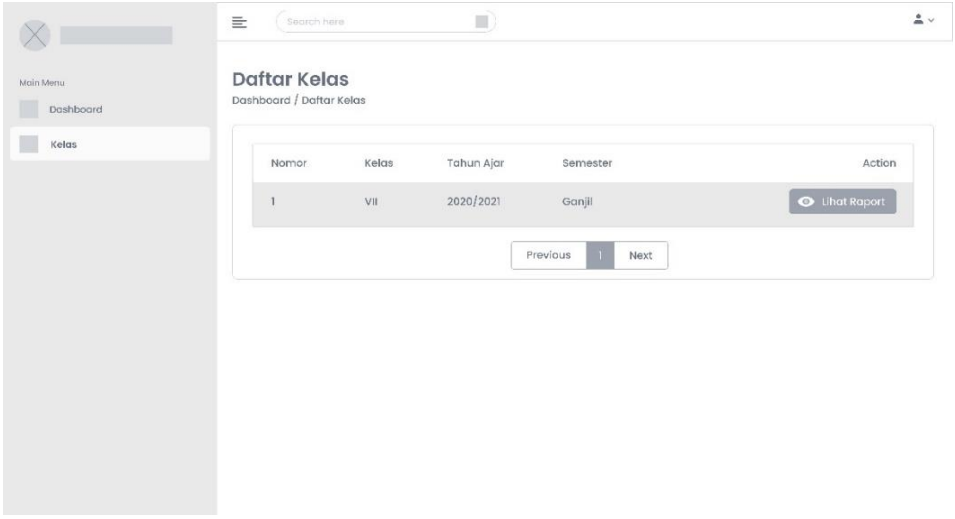
Tampilan perancangan halaman *dashboard* admin dapat dilihat pada Gambar 6 di bawah ini.



Gambar 6. Perancangan Halaman *Dashboard* Admin

Gambar 6 di atas merupakan tampilan perancangan halaman *dashboard* admin yang akan menampilkan ringkasan dari data yang terdapat pada aplikasi.

Tampilan perancangan halaman kelas pada akun siswa dapat dilihat pada Gambar 7 di bawah ini.



Gambar 7. Perancangan Halaman Kelas Pada Akun Siswa

Gambar 7 di atas merupakan tampilan perancangan halaman kelas pada akun siswa yang akan menampilkan data kelas baik yang sedang berlangsung maupun yang telah berlangsung. Pada halaman ini nantinya masing-masing dari data kelas dapat melihat detail raport dengan cara menekan tombol “Lihat Raport”.

HASIL DAN PEMBAHASAN

Proses Sorting dengan Algoritma Heap Sort

Sebagai sampel perhitungan, 15 data nilai rapor siswa digunakan untuk dilakukan pengurutan nilai rata-rata semester dengan menggunakan algoritma *heap sort*. Detail nilai rapor 15 siswa dapat dilihat lebih jelasnya pada Tabel 1.

 https://doi.org/10.56211/helloworld.v1i3.144

Tabel 1. Indeks Nilai Rata-Rata Semester

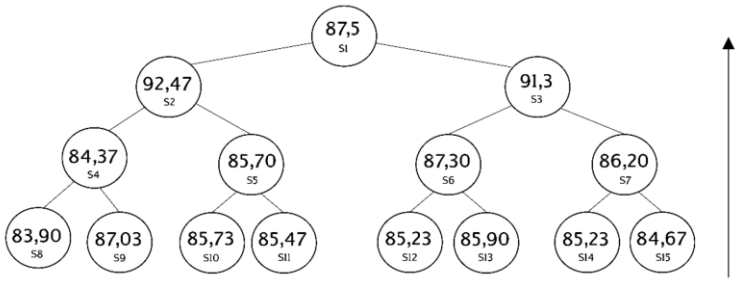
Kode Siswa	Nilai Rata-Rata Semester
S1	87,5
S2	92,47
S3	91,3
S4	84,37
S5	85,70
S6	87,30
S7	86,20
S8	83,90
S9	87,03
S10	85,73
S11	85,47
S12	85,23
S13	85,90
S14	85,23
S15	84,67

Kode siswa digunakan sebagai kode unik dari nilai rata-rata semester yang akan dilakukan pengurutan untuk menghindari adanya kesalahan kepemilikan nilai apabila terdapat nilai yang sama serta sebagai kode dari nama siswa yang terlalu panjang. Oleh karena itu untuk mempermudah, Tabel 1 di atas diubah menjadi Tabel 2 di bawah ini.

Tabel 2. Indeks Nilai Rata-Rata Semester (Lanjutan)

Indeks	NRS
1	87,5 (S1)
2	92,47 (S2)
3	91,3 (S3)
4	84,37 (S4)
5	85,70 (S5)
6	87,30 (S6)
7	86,20 (S7)
8	83,90 (S8)
9	87,03 (S9)
10	85,73 (S10)
11	85,47 (S11)
12	85,23 (S12)
13	85,90 (S13)
14	85,23 (S14)
15	84,67 (S15)

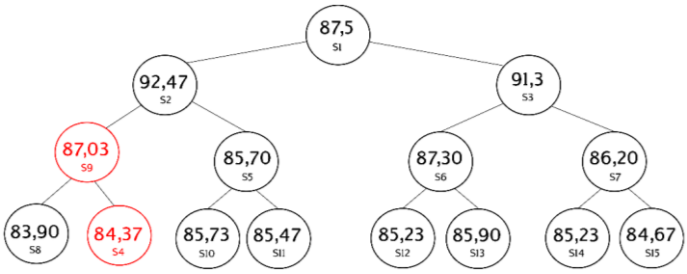
Untuk menentukan *left child* pada *node root* maka, $2i = 2 \times 1 = 2$. Nilai i = 1 dikarenakan *node root* berada pada index 1 sehingga *left child* dari *node root* adalah index 2. Sedangkan untuk menentukan *right child* pada *node root* maka, $2i + 1 = (2 \times 1) + 1 = 3$. Nilai i = 1 dikarenakan *node root* berada pada index 1 sehingga *right child* dari *node root* adalah indeks 3. Langkah ini dilakukan terus-menerus sampai semua indeks telah terbentuk menjadi *heap tree*. Untuk lebih jelasnya dapat dilihat pada Gambar 8.



Gambar 8. *Heap Tree*

Setelah *heap tree* terbentuk langkah selanjutnya adalah membentuk *max-heap*. *Parent node* pada sebuah *max heap* selalu lebih besar daripada atau sama dengan *child node*-nya. *Max heap* dilakukan dari level bawah ke level atas dan dilakukan perbandingan *child node* dengan *parent node*-nya. Langkah ini dilakukan terus-menerus sampai tersisa 1 *node* yang menjadi nilai akhir atau nilai terkecil.

Untuk membentuk *max heap*, dimulai dari level paling bawah dan dilakukan perbandingan *child node* dengan *parent node*-nya. Karena ketentuan dari *max heap* adalah *parent node* pada sebuah *max heap* selalu lebih besar daripada atau sama dengan *child node*-nya. Oleh karena itu, *parent node* (84,37 (S4)) dengan *child node* [83,9 (S8), 87,03 (S9)] dilakukan perbandingan dan ternyata *parent node* (84,37 (S4)) memiliki nilai yang lebih kecil daripada salah satu nilai *child node*-nya yaitu 87,03 (S9) maka, kedua nilai tersebut terjadi pertukaran posisi sehingga *parent node* pada langkah ini menjadi 87,03 (S9) dan memiliki *child node* [83,9 (S8), 84,37 (S4)]. Untuk lebih jelasnya dapat dilakukan perbandingan pada Gambar 8 dengan Gambar 9.



Gambar 9. *Heap Sort* Langkah 1

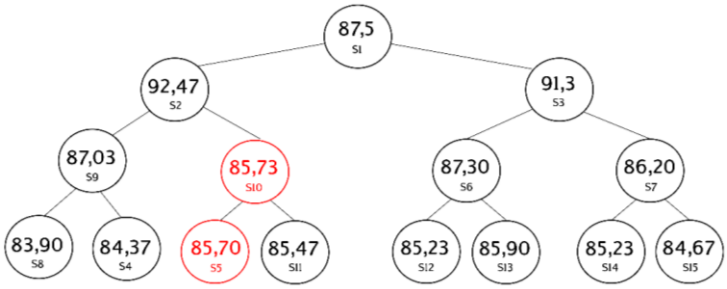
Dari Tabel 1, dapat dilihat bahwa nilai pada indeks ke-4 adalah 84,37 (S4). Namun, pada Tabel 2 di bawah ini nilai pada indeks ke-4 berubah menjadi 87,03 (S9) dimana posisi awalnya berada pada index ke-9. Terjadinya pertukaran pada kedua nilai ini didasarkan pada posisi *parent node* dan *child node* pada *heap tree* saat pembentukan *max heap* yang dapat dilihat pada Gambar 8. Gambar 9 di atas jika dilihat dalam bentuk tabel, dapat dilihat pada Tabel 3.

Tabel 3. Representasi *Array Heap Sort* Langkah 1

Indeks	NRS
1	87,5 (S1)
2	92,47 (S2)
3	91,3 (S3)
4	87,03 (S9)
5	85,70 (S5)
6	87,30 (S6)
7	86,20 (S7)
8	83,90 (S8)
9	84,37 (S4)
10	85,73 (S10)
11	85,47 (S11)
12	85,23 (S12)
13	85,90 (S13)
14	85,23 (S14)
15	84,67 (S15)

Selanjutnya, jika dilihat pada Gambar 9 *parent node* selanjutnya adalah 85,7 (S5) dengan *child node* [85,73 (S10), 85,47 (S11)]. Dilakukan perbandingan nilai pada *parent node* dengan *child node* nya. Diperoleh bahwa nilai dari salah satu *child node*-nya yaitu 85,73 (S10) memiliki nilai yang lebih besar dibandingkan *parent node*-nya yaitu 85,7 (S5). Oleh karena itu, *parent node* pada langkah ini berubah menjadi 85,73 (S10) dan *parent node* sebelumnya yaitu 85,7 (S5) menjadi *child node*-nya. Pertukaran posisi kedua nilai tersebut dapat dilihat pada Gambar 10 di bawah ini.

 https://doi.org/10.56211/helloworld.v1i3.144



Gambar 10. *Heap Sort* Langkah 2

Dari Gambar 10, jika dilihat dalam bentuk tabel maka dapat dilihat pada Tabel 4.

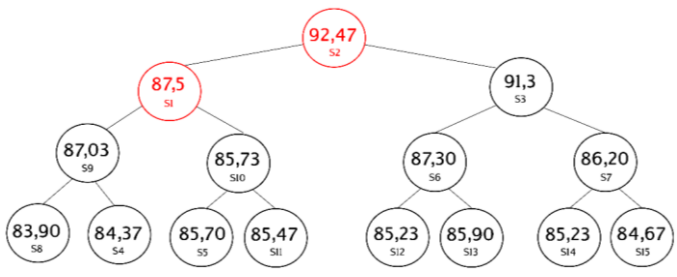
Tabel 4. Representasi *Array Heap Sort* Langkah 2

Indeks	NRS
1	87,5 (S1)
2	92,47 (S2)
3	91,3 (S3)
4	87,03 (S9)
5	85,73 (S10)
6	87,30 (S6)
7	86,20 (S7)
8	83,90 (S8)
9	84,37 (S4)
10	85,7 (S5)
11	85,47 (S11)
12	85,23 (S12)
13	85,90 (S13)
14	85,23 (S14)
15	84,67 (S15)

Sama halnya dengan Tabel 3, Tabel 4 di atas pertukaran posisi ditandai dengan latar berwarna.

Selanjutnya, hal yang sama dilakukan seperti pada langkah 1 dan 2 hanya saja, pada langkah ini ditemukan *parent node* dengan nilai 87,3 (S6) dan 86,2 (S7) sudah sesuai dengan ketentuan *max heap* yaitu nilai *parent node* selalu lebih besar dari *child node*-nya. Oleh karena itu, *parent node* yang akan dilakukan perbandingan selanjutnya adalah level yang berada di tengah yaitu *parent node* dengan nilai 92,47 (S2) dan 91,3 (S3). Namun, *parent node* dengan nilai 92,47 (S2) dan 91,3 (S3) sudah memiliki nilai yang terbesar diantara masing-masing *child node*-nya. Maka, perbandingan yang akan dilakukan selanjutnya adalah *node root* yang berada pada level tertinggi.

Node root (87,5 (S1)) memiliki *child node* [92,47 (S2), 91,3 (S3)] dimana jika dilakukan perbandingan, nilai terbesar berada pada salah satu *child node*-nya yaitu 92,47 (S2). Maka, *node root* berubah menjadi 92,47 (S2) dengan *child node* [87,5 (S1), 91,3 (S3)]. Proses ini dapat dilihat dengan cara membandingkan Gambar 9 dengan Gambar 10.



Gambar 11. *Heap Sort* Langkah 3

Berdasarkan Gambar 11, berikut merupakan penggambaran *heap sort* pada langkah 3:

Tabel 5. Representasi *Array Heap Sort* Langkah 3

Indeks	NRS
1	92,47 (S2)
2	87,5 (S1)
3	91,3 (S3)
4	87,03 (S9)
5	85,73 (S10)
6	87,30 (S6)
7	86,20 (S7)
8	83,90 (S8)
9	84,37 (S4)
10	85,7 (S5)
11	85,47 (S11)
12	85,23 (S12)
13	85,90 (S13)
14	85,23 (S14)
15	84,67 (S15)

Ketiga langkah yang telah dilakukan diulangi terus-menerus sampai tersisa 1 *node*. Hasil akhir pengurutan *max-heap* dapat dilihat pada Gambar 12 di bawah ini.

Gambar 12. *Heap Sort* Langkah 40

Representasi *array* dari Gambar 12 dapat dilihat pada Tabel 6 di bawah ini.

Tabel 6. Representasi *Array Heap Sort* Langkah 40

Indeks	NRS
1	83,9 (S8)
2	84,37 (S4)
3	84,67 (S15)
4	85,23 (S14)
5	85,23 (S12)
6	85,47 (S11)
7	85,7 (S5)
8	85,73 (S10)
9	85,9 (S13)
10	86,2 (S7)
11	87,03 (S9)
12	87,3 (S6)
13	87,5 (S1)
14	91,3 (S3)
15	92,47 (S2)

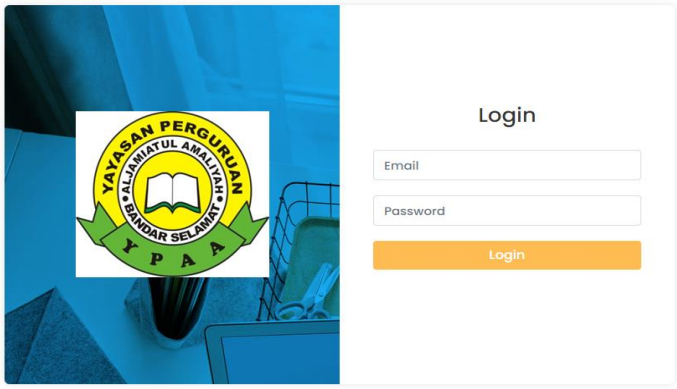
Dari pengurutan nilai rata-rata semester yang dilakukan pada 15 siswa dengan menggunakan algoritma *heap sort* diperoleh hasil pengurutan secara menaik (*ascending*) yang dapat dilihat pada Tabel 7.

Tabel 7. Hasil Pengurutan *Heap Sort* Terhadap 15 Data Sampel

Kode Siswa	Nilai Rata-Rata Semester
S8	83,9
S4	84,37
S15	84,67
S14	85,23
S12	85,23
S11	85,47
S5	85,7
S10	85,73
S13	85,9
S7	86,2
S9	87,03
S6	87,3
S1	87,5
S3	91,3
S2	92,47

Implementasi

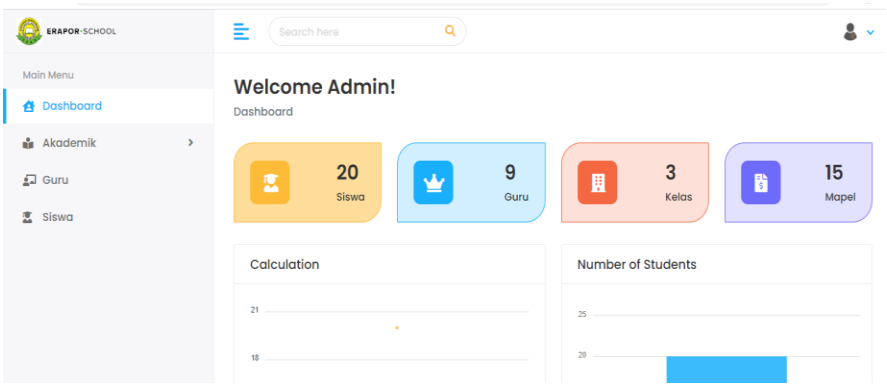
Tampilan halaman *login* dapat dilihat pada Gambar 13 di bawah ini.



Gambar 13. Halaman *Login*

Gambar 13 di atas merupakan tampilan halaman *login* memiliki 2 *field input* yang dapat diisi oleh *user* agar dapat masuk ke aplikasi. Dua *field input* tersebut merupakan *field input* untuk tipe *email* dan *password*. Tampilan halaman *login* ini berlaku untuk semua *user* (admin, guru dan siswa).

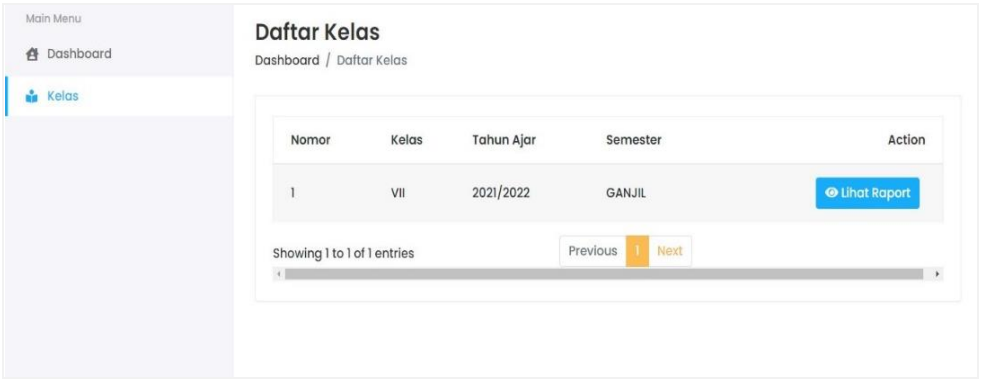
Tampilan halaman *dashboard* admin dapat dilihat pada Gambar 14 di bawah ini.



Gambar 14. Halaman *Dashboard* Admin

Gambar 14 merupakan tampilan halaman *dashboard* admin yang menampilkan ringkasan informasi data yang terdapat pada aplikasi.

Tampilan halaman kelas pada akun siswa dapat dilihat pada Gambar 15 di bawah ini.



Gambar 15. Halaman Kelas Pada Akun Siswa

Gambar 15 merupakan tampilan halaman kelas pada akun siswa yang akan menampilkan data kelas baik yang sedang berlangsung maupun yang telah berlangsung. Pada halaman ini masing-masing dari data kelas dapat melihat detail raport dengan cara menekan tombol “Lihat Raport”.

KESIMPULAN DAN SARAN

Pada aplikasi e-raport siswa Mts S Aljamiatul Amaliyah Desa Bandar Selamat Labuhanbatu Utara diperoleh nilai siswa secara berurutan adalah sebagai berikut Nur Hidayat (83,90), Aji Prasetio (84,37), Murlas Ady Putra (84,67), Muhammad Sahputra Simatupang (85,13), Adil Firmansyah Nasution (85,23), M. Hasan Basri Mangunsong (85,61), Isra Auliya (85,70), Nabila Az-Zahra Sinaga (85,73), Salsabila Anazra (85,90), Nadila A-Zahra Sinaga (86,20), Ayun Fadilah Amanda (87,03), Dania Simamora (87,30), Fahmi Saka Falah (87,50), Sri Wahyuni (91,30) dan Aura Perdana Putri (92,77). Algoritma *heap sort* memiliki kelebihan yaitu, penggunaan memori yang minim karena selain untuk menampung daftar awal *item* yang akan disortir, tidak memerlukan ruang memori tambahan untuk bekerja serta termasuk algoritma pengurutan yang sangat mudah dipahami. Sedangkan kekurangannya yaitu, membutuhkan waktu yang lebih banyak karena bekerja dengan cara membandingkan nilai pada setiap *parent node*-nya lalu dieksekusi dan tidak efektif digunakan pada data yang memiliki kompleksitas yang tinggi atau kumpulan data yang besar. Penelitian ini hanya fokus pada pembahasan pembuatan aplikasi e-raport dengan algoritma *heap sort*. Oleh karena itu, untuk penelitian selanjutnya diharapkan dapat melakukan perbandingan metode pengurutan *heap sort* dengan metode pengurutan lainnya

DAFTAR PUSTAKA

[1] H. Situmorang, “Simulasi Pengurutan Data dengan Algoritma Heap Sort,” *Jurnal Mahajana Informasi*, vol. 1, no. 2, 2016.
[2] A. Wijaya and N. Feter, “Aplikasi Simulasi Pengurutan Data Menggunakan Algoritma Heap Sort,” *Jurnal Pseudocode*, vol. 2, no. 2, pp. 81–88, 2015, [Online]. Available: www.ejournal.unib.ac.id
[3] C. al Akhyati, A. Johar, and B. Susilo, “Perangkat Lunak Pendukung Pembelajaran Algoritma Heapsort,” *Jurnal Rekursif*, vol. 2, no. 1, pp. 37–44, 2014.
[4] H. Abdurahman and A. R. Riswaya, “Aplikasi Pinjaman Pembayaran Secara Kredit Pada Bank Yudha Bhakti,” *Jurnal Computech & Bisnis*, vol. 8, no. 2, pp. 61–69, Dec. 2014.
[5] Rinaldi, “Rancang Aplikasi Untuk Pengurutan Data Dengan Metode Heap Sort,” Skripsi, Universitas Pembangunan Panca Budi, Medan, 2019.
[6] R. Abdulloh, *Easy & Simple Web Programming*. Jakarta: PT Elex Media Komputindo, 2016.
[7] J. Enterprise, *Otodidak Pemrograman Javascript*. Jakarta: PT Elex Media Komputindo, 2017.
[8] C. A. Pamungkas, *Dasar Pemrograman Web dengan PHP*. Yogyakarta: CV Budi Utama, 2017.
[9] R. Abdullah, *7 in 1 Pemrograman Web Tingkat Lanjut*. Jakarta: PT Elex Media Komputindo, 2018.
[10] B. Nugroho, *Aplikasi Pemrograman Web Dinamis dengan PHP dari MySQL*, Cetakan I. Yogyakarta: Gava Media, 2019.

BIODATA PENULIS

Mindi Wahyuni, S.T.
Lahir di Bukit Rata pada tanggal 03 Oktober 1999, menyelesaikan pendidikan S-1 di Universitas Islam Sumatera Utara, Fakultas Teknik Program Studi Teknik Informatika.