

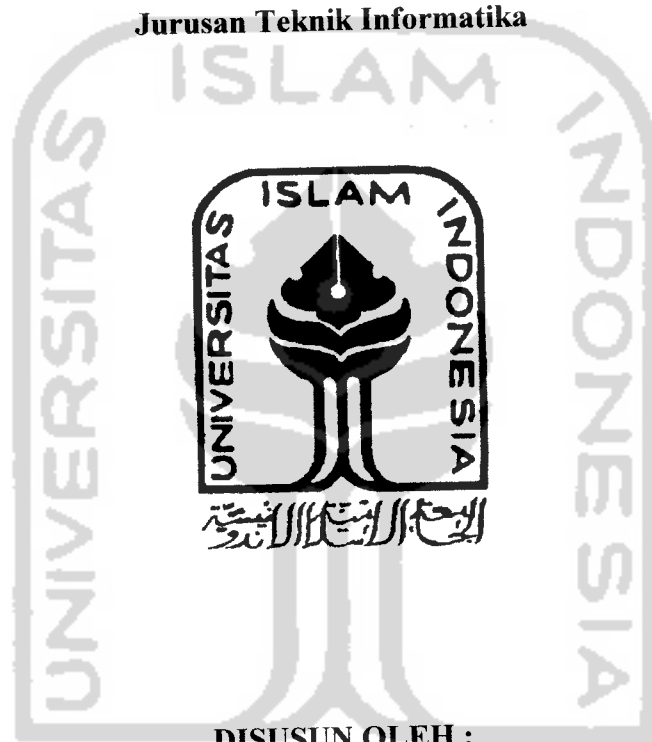
**STUDI BEBERAPA
METODE *SORTING***

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat

Untuk Memperoleh Gelar Sarjana

Jurusan Teknik Informatika



DISUSUN OLEH :

Nama : Susetyo Rusna Wibowo

No.MHS : 98 523 126

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007**

LEMBAR PENGESAHAN PEMBIMBING
STUDI BEBERAPA
METODE *SORTING*

TUGAS AKHIR



Disusun Oleh :

Susetyo Rusna Wibowo
98 523 126

Yogyakarta, 10 Juli 2007

Dosen Pembimbing Tugas Akhir

A handwritten signature in black ink, appearing to be 'Yudi Prayudi', written over a horizontal line.

(Yudi Prayudi, S.Si, M.Kom.)

LEMBAR PENGESAHAN PENGUJI

STUDI BEBERAPA METODE *SORTING*

Tugas Akhir

Disusun Oleh

Nama : SUSETYO RUSNA WIBOWO
No.Mhs : 98 523 126

Telah Dipertabankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 10 Juli 2007

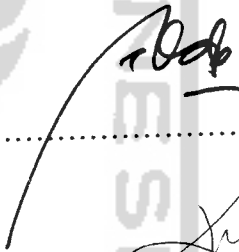


Tim Penguji

Tanda Tangan

Yudi Prayudi, S.Si., M. Kom.


Ami Fauziah, ST., MT.

Syarif Hidayat, S. Kom.


.....

.....

.....

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Teknologi Industri
Universitas Islam Indonesia




(Yudi Prayudi, S.Si, M.Kom)

Persembahan



Kupersembahkan Tugas Akhir ini
Sesembahanku Yang Maha Segalanya ALLAH SWT
Sujud Syukurku kepada-Nya

Untuk Ayah dan Bundaku tercinta
H. Bambang Sumena & Hj. Sri Rusmiaty
Tak akan pernah sebanding kasih sayang dan pengorbananmu
kepada Ananda dengan apa yang Ananda telah berikan

Untuk Adikku tersayang
Alm. Meryana Dwi Lestari (27 hari) dan Nurina Dewi
Kebersamaan dalam persaudaraan adalah kebahagiaan buat
orang-orang yang kita sayangi

Untuk Saudara - saudaraku
Tak pernah ku lupakan jasa, doa serta kebaikan
yang telah aku terima

Untuk Sahabat – sahabatku &
Orang – orang yang pernah ada di hati
Persahabatan, kebersamaan dan kekompakkan
yang kita jalani
hendaknya selalu kita jaga
sampai kapan pun
sehingga silaturahmi tetap terjaga



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Spesial Thanks

1. Om Budi dan Mba Ida yang selalu memberi motivasi, nasehat dan doa.
2. Ade Fany dan Ibu di Bandung, Bowo sudah menganggap Ibu sebagai Ibu Bowo sendiri. Terima kasih atas saran dan masukannya.
3. Untuk motorku **Satria FU150** dan helmku **KYT**, terima kasih telah menemaniku di perjalanan yang penuh polusi. Panas, hujan kita lalui bersama
4. .Saudaraku Ata di Bekasi, thanks ya atas dorongannya.
5. Teman – teman UII Informatika '98, Panji (terima kasih telah banyak membantuku), Galih, Beny n Rio (klo ga ada kalian pendadaranku pasti berantakan, Thank's Guy), Guruh, Nanang, Nyong, Novri, Widi, Romy, Bejo, Metz, Tatah, Gendut, Indra “superman” serta teman – teman yang tidak tersebut, thanks bgt atas dukungannya.
6. Buat teman – temanku yang lain, Wisnu (kamu pasti bisa wis, cayoo..!), Titin (terima kasih atas *printernya* ☺), Boim, Iga, Lusi, Dendi, Ony, Heny, Cris, Ratna, Eta, Poy, Mas Nanang, Buaya, Darma, Diah serta teman – temanku yang ada di luar kota, Cesy, Ima, Ely, Ani, Zeeka, Afif terima kasih atas persahabatannya.
7. Semua pihak yang tidak tersebutkan satu - persatu, terima kasih untuk segalanya.
8. Indonesia Raya.

MOTTO

“Sesungguhnya kesulitan itu selalu disertai oleh kemudahan”

(QS. Al-Baqarah: 185)

“Diantara pekerjaan ibadah adalah memanfaatkan ilmu untuk kemashlahatan orang banyak”

(Baw)

“Kejujuran akan membawa ketenangan, ketidakjujuran akan menimbulkan keragu-raguan”

(Baw)

“Cinta tidak dapat digambarkan tanpa iman tetapi iman pun tidak dapat dibayangkan tanpa cinta”

(Baw)

KATA PENGANTAR



Assalaamu'alaikum Wr. Wb.

Alhamdulillah puji syukur kehadirat Allah SWT, atas limpahan hidayah, taufiq, serta 'inayah-Nya, sehingga penyusunan tugas akhir dengan judul "**STUDI BEBERAPA METODE SORTING**" ini dapat terselesaikan dengan baik.

Sholawat serta salam semoga senantiasa tercurah atas Nabi Muhammad SAW, para kerabat, sahabat, serta pengikut setianya hingga hari kiamat nanti, Amien.

Sehubungan dengan terselesaikannya penyusunan tugas akhir ini, penyusun mengucapkan banyak terima kasih kepada pihak-pihak yang telah memberikan dukungannya baik secara langsung maupun tidak. Dengan penuh rasa syukur kami ucapkan terima kasih kepada :

1. Bapak Yudi Prayudi, S.Si., M.Kom., Selaku Ketua Jurusan Teknik Informatika Universitas Islam Indonesia dan selaku Dosen Pembimbing yang telah memberikan bimbingan dan petunjuk dalam penyusunan tugas akhir ini.
2. Bapak Fathul Wahid, S.T., M.Sc., Selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.

3. Segenap Staf Pengajar Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia yang telah banyak memberikan ilmu pengetahuan dan mendidik penyusun selama menjalani pendidikan sehingga dapat menyelesaikan kuliah dengan baik.
4. Segenap Karyawan dan Karyawati Perpustakaan Fakultas Teknologi Industri Universitas Islam Indonesia yang telah banyak membantu penulis dalam menyelesaikan tugas akhir ini .
5. Semua pihak yang tidak dapat disebutkan satu per satu.

Penyusun menyadari akan banyaknya kekurangan dalam penyusunan tugas akhir ini. Untuk itu kritik dan saran senantiasa kami harapkan.

Harapan kami semoga karya yang sederhana ini dapat menjadi sumbangan yang berarti bagi kampus tercinta Fakultas Teknologi Industri Universitas Islam Indonesia Yogyakarta.

Wassalaamu 'alaikum Wr. Wb.

Yogyakarta, 10 Juli 2007

Penyusun

ABSTRAKSI

Pengurutan adalah subjek klasik dalam ilmu komputer. Ada tiga alasan dalam mempelajari algoritma pengurutan. Pertama, algoritma pengurutan menggambarkan banyak pendekatan yang kreatif dalam pemecahan suatu masalah dan pendekatan ini dapat digunakan memberikan solusi dalam pemecahan masalah yang lain. Kedua, algoritma pengurutan adalah latihan dasar pemrograman yang bagus (terdiri dari pengulangan, pemilihan dan array). Ketiga, algoritma pengurutan adalah contoh yang sangat bagus untuk menunjukkan performa suatu algoritma.

Salah satu cara termudah memahami materi ini adalah dengan cara memvisualisasikan materinya secara langsung. Hal ini tentunya tidaklah mudah, paling tidak didukung oleh beberapa alasan berikut: pertama, seringkali terjadi pertukaran data dari suatu posisi ke posisi lain selama proses pengurutan berlangsung. Kedua, sulitnya membayangkan dan mengingat posisi data yang berpindah dan data yang tidak berpindah. Ketiga, pertukaran dan perpindahan data tergantung kepada metode pengurutan yang digunakan. Algoritma pengurutan yang digunakan adalah *bubble sort*, *bidirectional bubble sort*, *quick sort*, *insertion sort*, *selection sort*, *quick bubble sort*, *fast quick sort*, *enhanced quick sort*.

Keywords: metode pengurutan (bubble sort, bidirectional bubble sort, quick sort, insertion sort, selection sort, quick bubble sort, fast quick sort, enhanced quick sort), visualisasi, java.

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN PEMBIMBING.....	ii
LEMBAR PENGESAHAN PENGUJI.....	iii
HALAMAN PERSEMBAHAN.....	iv
HALAMAN MOTTO.....	vi
KATA PENGANTAR.....	vii
ABSTRAKSI.....	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian.....	3
1.7 Sistematika Penulisan	4
BAB II LANDASAN TEORI	7
2.1 Algoritma Pengurutan.....	7

2.1.1	Algoritma <i>Bubble Sort</i>	7
2.1.2	Algoritma <i>Bidirectional Bubble Sort</i>	10
2.1.3	Algoritma <i>Quick Sort</i>	10
2.1.4	Algoritma <i>Selection Sort</i>	12
2.1.5	Algoritma <i>Fast Quick Sort</i>	13
2.1.6	Algoritma <i>Quick Bubble Sort</i>	13
2.1.7	Algoritma <i>Insertion Sort</i>	14
2.1.8	Algoritma <i>Enhanced Quick Sort</i>	19
BAB III	ANALISIS KEBUTUHAN PERANGKAT LUNAK	21
3.1	Metode Analisis.....	21
3.2	Spesifikasi Yang Akan Dikembangkan	21
3.3	Hasil Analisis Kebutuhan	22
3.3.1	Analisis Kebutuhan Data Masukan.....	22
3.3.2	Analisis Kebutuhan Proses	22
3.3.3	Analisis Kebutuhan Data Keluaran.....	23
3.3.4	Analisis kebutuhan Antar Muka	23
BAB IV	PERANCANGAN PERANGKAT LUNAK	24
4.1	Metode Perancangan.....	24
4.2	Hasil Perancangan	24
4.2.1	Diagram Alir.....	24
4.2.2	Antarmuka	26

BAB V	IMPLEMENTASI PERANGKAT LUNAK	27
5.1	Batasan Implementasi	27
5.1.1	Analisis Kebutuhan Perangkat Lunak	27
5.1.2	Analisis Kebutuhan Perangkat Keras.....	27
5.2	Tampilan Aplikasi Simulasi Visualisasi Pengurutan Data ...	28
5.2.1	Tampilan Menu.....	28
5.2.2	Pemilihan Algoritma Pengurutan.....	29
5.2.3	Proses Pengacakan Data.....	30
5.2.4	Proses Pengurutan Data.....	31
BAB VI	ANALISIS KINERJA PERANGKAT LUNAK	35
6.1	Pengujian Sistem	35
6.1.1	Proses Pemilihan Algoritma.....	35
6.1.2	Proses Pengurutan.....	36
6.2	Kinerja Perangkat Lunak	39
BAB VII	PENUTUP	41
7.1	Kesimpulan	41
7.2	Saran	41

DAFTAR PUSTAKA

DAFTAR GAMBAR

Gambar 2.1	Proses Pengurutan Pemain	8
Gambar 2.2	Posisi Pemain Setelah Proses Pengurutan Pertama	9
Gambar 2.3	Contoh Pola Data yang Akan Diurutkan	11
Gambar 2.4	Proses Pengurutan	11
Gambar 2.5	Data Setelah Diurutkan	12
Gambar 2.6	Data Dasar yang Hendak Diurutkan	15
Gambar 2.7	Langkah Pertama Pengurutan <i>Insertion Sort</i>	15
Gambar 2.8	Langkah Kedua Pengurutan	16
Gambar 2.9	Langkah Ketiga Pengurutan	16
Gambar 2.10	Langkah keempat Pengurutan	17
Gambar 2.11	Langkah Kelima Pengurutan	17
Gambar 2.12	Langkah Keenam Pengurutan	18
Gambar 2.13	Langkah Ketujuh Pengurutan	18
Gambar 2.14	Pengurutan Selesai	19
Gambar 4.1	Diagram Alir Sistem	25
Gambar 4.2	Rancangan Aplikasi	26
Gambar 5.1	Tampilan Menu	29
Gambar 5.2	<i>Combo box</i> Pemilihan Algoritma	30
Gambar 5.3	Visualisasi Data Hasil Proses Pengacakan	31
Gambar 5.4	Hasil Penghitungan Waktu dan Jumlah Langkah	32
Gambar 5.5	<i>Message box</i> Informasi Selesai nya Pengurutan	32

Gambar 5.6	Laporan Perhitungan Waktu dan Jumlah Langkah	33
Gambar 6.1	Pesan Kesalahan Karena Belum Memilih Metode.....	36
Gambar 6.2	Informasi Proses Sedang Berjalan.....	37
Gambar 6.3	Visualisasi Data Sebelum Proses Pengurutan	37
Gambar 6.4	Visualisasi Data Setelah Proses Pengurutan	38
Gambar 6.5	Hasil Proses Simulasi Pengurutan Data.....	39



DAFTAR TABEL

Tabel 2.1	Data dan Proses Pengurutan <i>Bubble Sort</i>	9
Tabel 2.2	Data dan Proses Pengurutan <i>Bidirectional Bubble Sort</i>	10
Tabel 2.3	Data dan Proses Pengurutan <i>Quick Sort</i>	12
Tabel 2.4	Data dan Proses Pengurutan <i>Selection Sort</i>	13
Tabel 2.5	Data dan Proses Pengurutan <i>Fast Quick Sort</i>	13
Tabel 2.6	Data dan Proses Pengurutan <i>Quick Bubble Sort</i>	14
Tabel 2.7	Data dan Proses Pengurutan <i>Insertion Sort</i>	19
Tabel 2.8	Data dan Proses Pengurutan <i>Enhanced Quick Sort</i>	20



BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengurutan adalah subjek klasik dalam ilmu komputer. Ada tiga alasan dalam mempelajari algoritma pengurutan. Pertama, algoritma pengurutan menggambarkan banyak pendekatan yang kreatif dalam pemecahan suatu masalah dan pendekatan ini dapat digunakan memberikan solusi dalam pemecahan masalah yang lain. Kedua, algoritma pengurutan adalah latihan dasar pemrograman yang bagus (terdiri dari pengulangan, pemilihan dan array). Ketiga, algoritma pengurutan adalah contoh yang sangat bagus untuk menunjukkan performa suatu algoritma.

Mata kuliah Struktur Data merupakan salah satu mata kuliah yang diajarkan pada banyak program studi ilmu komputer. Mata kuliah ini mempelajari struktur data dan algoritma, diantaranya array, linked list, stack, queue, table hash, heap, metode pengurutan, metode pencarian, binary tree dan banyak lagi.

Salah satu cara termudah memahami materi mata kuliah ini adalah dengan cara memvisualisasikan materinya secara langsung, misalnya algoritma pengurutan. Tanpa visualisasi, algoritma pengurutan yang dipelajari harus dibayangkan oleh masing-masing orang. Hal ini tentunya tidaklah mudah, paling tidak didukung oleh beberapa alasan berikut: pertama, seringnya terjadi pertukaran data dari suatu posisi ke posisi lain selama proses pengurutan

berlangsung. Kedua, sulitnya membayangkan dan mengingat posisi data yang berpindah dan data yang tidak berpindah. Ketiga, pertukaran dan perpindahan data tergantung kepada metode pengurutan yang digunakan.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka penulis merumuskan permasalahan yang harus dipecahkan yaitu:

Membuat model aplikasi yang dapat menunjukkan peforma bermacam-macam algoritma pengurutan.

1.3 Batasan Masalah

Agar penelitian yang dilakukan lebih terfokus pada pembangunan sistem, maka perlu beberapa batasan masalah, sebagai berikut:

1. Pemrograman menggunakan Applet java
2. Jenis algoritma pengurutan yang digunakan : *bubble sort, bidirectional bubble sort, quick sort, selection sort, fast quick sort, insertion sort, quick bubble sort, enhanced quick sort.*
3. Data yang diurutkan berupa bilangan real yang diambil secara acak oleh computer. Terdiri dari 300 angka yang masing-masing direpresentasikan sebagai garis yang memiliki panjang yang berbeda sesuai dengan nilai dari angka hasil pengacakan.
4. Khusus class `AlgorimDasar`, `AlgorimaPengurutanBowo`, merupakan modifikasi dari kelas `SortItem`, `SortAlgorithm` yang dikembangkan dan

dimodifikasi dari program aslinya yang ditulis oleh James Gosling dari *developer java* pada tahun 1997

1.4 Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah sebagai berikut:

Membandingkan beberapa metode sorting yang sudah ada dan mengukur performa dari masing-masing algoritma pengurutan dalam menyelesaikan suatu masalah.

1.5 Manfaat Penelitian

Manfaat dilakukannya penelitian ini adalah sebagai berikut:

Membantu dalam pemilihan algoritma pengurutan.

1.6 Metodologi Penelitian

Metode yang digunakan dalam penelitian tugas akhir ini terdiri dari metode pengumpulan data dan metode perancangan system.

a. Metode Pengumpulan Data

Metode-metode yang digunakan dalam pengumpulan data ini yaitu:

Studi pustaka, dengan mengumpulkan data-data melalui buku-buku referensi yang berhubungan dengan permasalahan dan teori.

b. Metode Pengembangan Sistem

I. Analisis kebutuhan perangkat lunak

Pada perancangan sistem ini peneliti melakukan observasi terhadap data-data yang diperlukan berdasarkan sumber-sumber yang ada kaitannya dengan penelitian baik itu literature, dokumentasi atau catatan-catatan serta data-data yang berkaitan dengan algoritma pengurutan.

II. Perancangan sistem

Dalam hal ini menentukan perancangan proses, perancangan input dan output serta interface/antarmuka. Perancangan sistem ini dilakukan sesuai dengan sumber-sumber yang ada kaitannya dengan data-data yang diperlukan.

III. Implementasi sistem

Pada implementasi sistem ini adalah praktek langsung pada sistem yang telah dibuat dengan melakukan pengujian. Selain itu, sistem yang dibuat apakah perlu perbaikan atau tidak, dan sesuai dengan kebutuhan yang diperlukan atau tidak.

IV. Analisis Hasil

Analisis hasil diperoleh dari implementasi yang disempurnakan, dan kekurangannya serta apakah sudah layak digunakan.

1.7 Sistematika Penulisan

Agar penulisan ini mudah dimengerti dan memenuhi persyaratan, maka dalam penulisannya dibagi dalam tahapan-tahapan yang antara satu bab dengan bab lain merupakan satu rangkaian yang saling melengkapi yaitu:

- Bab I Membahas masalah umum tentang penyusunan tugas akhir yang meliputi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian serta sistematika penulisan.
- Bab II Menjelaskan landasan teori yang meliputi teknik dasar algoritma pengurutan dan bahasa pemrograman java.
- Bab III Menguraikan tentang hasil analisis kebutuhan sistem yang meliputi: analisis kebutuhan aplikasi, analisis kebutuhan input, analisis kebutuhan output, analisis kebutuhan hardware dan analisis kebutuhan software.
- Bab IV Membahas tahapan perancangan perangkat lunak yang meliputi metode perancangan, perancangan input dan output.
- Bab V Merupakan implementasi sistem yang akan menjabarkan mengenai desain implementasi, cara pengoperasian program, dan hasil program.
- Bab VI Menganalisis implementasi program yang sudah siap pakai dan pengujian terhadap perangkat lunak yang telah dihasilkan. Bagaimana keunggulan dan kekurangannya.

Bab VII Berisikan kesimpulan-kesimpulan atas apa yang telah dikerjakan kemudian diakhiri dengan saran-saran untuk perbaikan dimasa yang akan datang, dan saran atas apa yang telah dikerjakan.



BAB II

LANDASAN TEORI

2.1 Algoritma Pengurutan

Algoritma pengurutan yang digunakan dalam pembuatan perangkat lunak ini adalah :

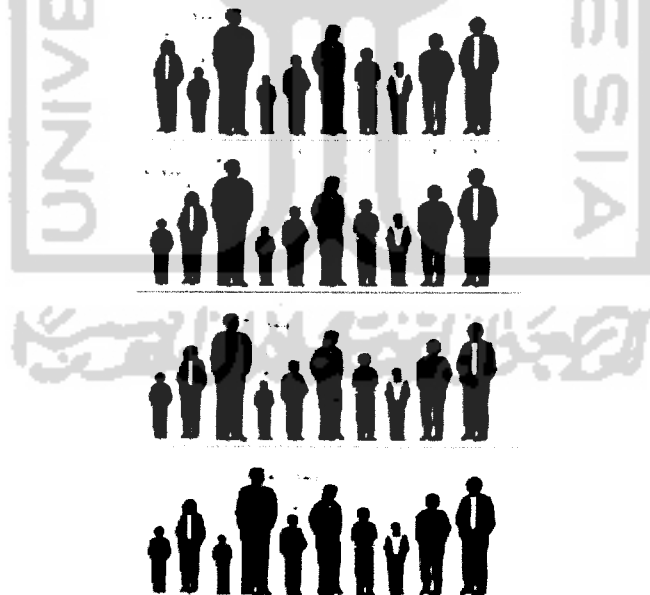
1. *bubble sort*
2. *bidirectional bubble sort*
3. *quick sort*
4. *selection sort*
5. *fast quick sort*
6. *insertion sort*
7. *quick bubble sort*
8. *enhanced quick sort*

2.1.1 Algoritma *bubble sort*

Bubble sort adalah algoritma pengurutan yang sederhana. Untuk menjelaskan kita mengambil contoh kasus pengurutan pemain *baseball* berdasarkan tinggi badan (Data Structures & Algorithms in java),(lihat gambar 2.1). Aturan dasarnya adalah sebagai berikut :

1. Dalam proses pengurutan hanya akan dibandingkan tinggi badan 2 pemain.
2. Jika pemain disebelah kiri pemain disebelah kanan memiliki tinggi badan lebih tinggi, posisi pemain tersebut akan kita tukar.
3. Setelah penukaran posisi, proses penukaran akan bergerak ke pemain berikutnya sampai urutan terakhir.(Gambar 2.2).
4. Setelah sampai pada urutan terakhir, pengurutan kembali ke posisi orang yang pertama. Proses ke satu sampai tiga diulang lagi sampai tidak ada orang/pemain yang dipertukarkan.

Algoritma ini adalah algoritma yang paling sederhana. Efektif digunakan untuk pengurutan data dengan jumlah yang tidak terlalu banyak.



Gambar 2.1 Proses pengurutan Pemain



Gambar 2.2 Posisi pemain setelah proses pengurutan pertama

Contoh data dan proses pengurutan pada tabel 2.1.

31	25	12	22	11	Data Awal
		12	22	11	langkah 1
25			22	11	langkah 2
25	12			11	langkah 3
25	12	22			langkah 4
		22	11	31	langkah 5
12			11	31	langkah 6
12	22			31	langkah 7
12	22	11			langkah 8
		11	25	31	langkah 9
12			25	31	langkah 10
12	11			31	langkah 11
12	11	22			langkah 12
		22	25	31	langkah 13
11			25	31	langkah 14
11	12			31	langkah 15
11	12	22			langkah 16
11	12	22	25	31	Hasil Akhir

Tabel 2.1 Data dan proses pengurutan menggunakan bubble sort

2.1.2 Algoritma *bidirectional bubble sort*

Algoritma ini merupakan pengembangan dari algoritma *bubble sort*. Perbedaannya adalah apabila algoritma *bubble sort* hanya melakukan pengurutan searah hanya dari atas kebawah atau dari kiri kekanan algoritma ini melakukan pengurutan dari dua arah. Misalnya ada 5 buah jenis data yang diurutkan, pengurutan mulai data pertama kemudian dibandingkan nilai data pertama dengan kedua, apabila data pertama lebih besar daripada data pertama maka nilai data ditukarkan. Setelah sampai pada data kelima data yang diurutkan, pengurutan berbalik arah dengan membandingkan data kelima dengan dengan keempat. Apabila data kelima memiliki nilai lebih kecil maka data akan ditukarkan, proses ini diulang sampai data pertama. Contoh data dan pengurutan pada tabel 2.2

31	25	12	22	11	Data Awal
		12	22	11	Langkah 1
25			22	11	Langkah 2
25	12			11	Langkah 3
25	12	22			Langkah 4
25	12	22			Langkah 5
25	12			31	Langkah 6
25			22	31	Langkah 7
		12	22	31	Langkah 8
		12	22	31	Langkah 9
11			22	31	Langkah 10
11	12			31	Langkah 11
11	12	22			Langkah 12
11	12	22	25	31	Hasil Akhir

Tabel 2.2 Data dan proses pengurutan menggunakan bidirectional bubble sort

2.1.3 Algoritma Quick Sort

Quick sort adalah algoritma pengurutan yang diciptakan oleh C.A.R.

Hoare. Algoritma ini berbasis teknik paradigma *divide and conquer*.

- *Divide*

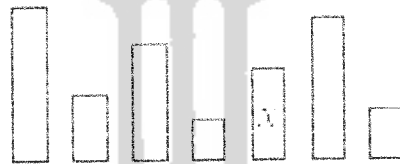
Mengambil salah satu nilai data yang akan diurutkan secara acak misalnya data tersebut kita beri nama x (biasa disebut *pivot* (Gambar 2.3)). Setelah *pivot* ditentukan kemudian kita bagi data yang akan kita urutkan menjadi tiga bagian L, G, E (Gambar 2.4). L adalah bagian yang memiliki nilai kurang dari nilai *pivot*, G adalah bagian yang memiliki nilai lebih besar dari *pivot* dan E adalah nilai *pivot* itu sendiri.

- *Recur*

Proses Kedua adalah kita mengurutkan nilai-nilai yang terdapat pada masing-masing bagian L dan G

- *Conquer*

Gabungkan nilai data pada bagian L, E dan G (Gambar 2.5).



Gambar 2.3 Contoh pola data yang akan diurutkan



Gambar 2.4 Proses Pengurutan



Gambar 2.5 Data Setelah Diurutkan

Contoh data dan proses pengurutan pada tabel 2.3.

31	25	12	22	11	Data Awal
25	31		11	22	Langkah 1
11	12	22	25	31	Hasil Akhir

Tabel 2.3 Data dan proses pengurutan menggunakan quick sort

2.1.4 Algoritma *Selection Sort*

Algoritma *Selection sort* merupakan pengembangan dari algoritma *Bubble sort* dengan mengurangi pertukaran data yang akan diurutkan. Algoritma ini menggunakan metode :

1. Cari nilai paling kecil dari data yang hendak diurutkan
2. Tukar posisi dengan data pada posisi pertama data yang diurutkan
3. Ulangi langkah satu dan dua dengan posisi pertama merupakan posisi setelah posisi data yang ditukar.

Tabel 2.4 (Posisi Awal Data yang Akan Diurutkan) terdiri dari 5 data yang akan diurutkan. Posisi pertama data tersebut ditunjukkan dengan warna abu-abu (angka 31). Langkah pertama adalah dengan mencari data yang paling kecil dari seluruh data yang ditunjukkan dengan warna krem (angka 11). Data dengan warna

abu-abu kemudian ditukarkan dengan data dengan warna krem. Setelah data ditukar, posisi data pertama akan berubah menjadi data setelah data penukaran (data penukaran berwarna biru).

31	25	12	22	11	Data Awal
31	25	12	22	11	Langkah 1
11	25	12	22	31	Langkah 2
11	12	25	22	31	Langkah 3
11	12	22	25	31	Langkah 4
11	12	22	25	31	Hasil Akhir

Tabel 2.4 Data dan proses pengurutan menggunakan selection sort

2.1.5 Algoritma *Fast Quick Sort*

Algoritma ini merupakan gabungan antara algoritma *quick sort* dengan *insertion sort*. Data hasil pembagian pada proses algoritma *quick sort* akan diurutkan menggunakan algoritma *insertion sort*.

Contoh data dan proses pengurutan pada tabel 2.5.

31	25	12	22	11	Data Awal
25	31		11	22	Langkah 1
11	12	22	25	31	Hasil Akhir

Tabel 2.5 Data dan proses pengurutan menggunakan fast quick sort

2.1.6 Algoritma *Quick Bubble Sort*

Algoritma ini merupakan gabungan dari algoritma *bubble sort* dan *quick sort*. Apabila jumlah data pada saat pembagian data pada algoritma kurang dari 6 maka akan dilakukan pengurutan data menggunakan *bubble sort*.

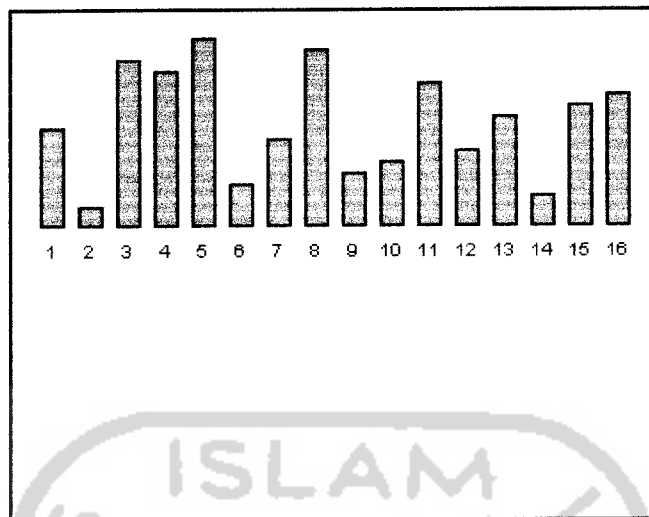
Contoh data dan proses pengurutan quick bubble sort pada tabel 2.6

31	25	12	22	11	Data Awal
25	31		11	22	Langkah 1
11	12	22	25	31	Hasil Akhir

Tabel 2.6 Data dan proses pengurutan menggunakan quick bubble sort

2.1.7 Algoritma *Insertion Sort*

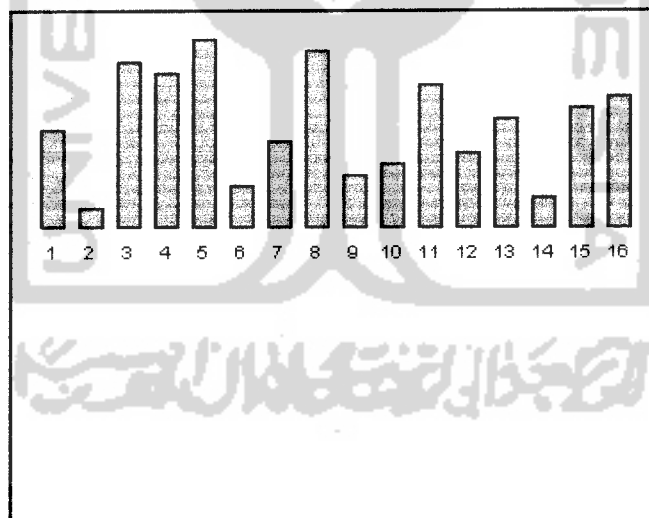
Pengurutan sisip adalah metode pengurutan dengan cara menyisipkan elemen larik pada posisi yang tepat. Pencarian posisi yang tepat dilakukan dengan melakukan pencarian beruntun di dalam larik. Selama pencarian posisi yang tepat dilakukan pergeseran elemen larik. Sering digambarkan seperti dalam mengurutkan angka dalam permainan kartu. Sebagai contoh ada kasus dengan 16 data yang harus diurutkan (Gambar 2.6).



Gambar 2.6 Data Dasar yang Hendak Diurutkan

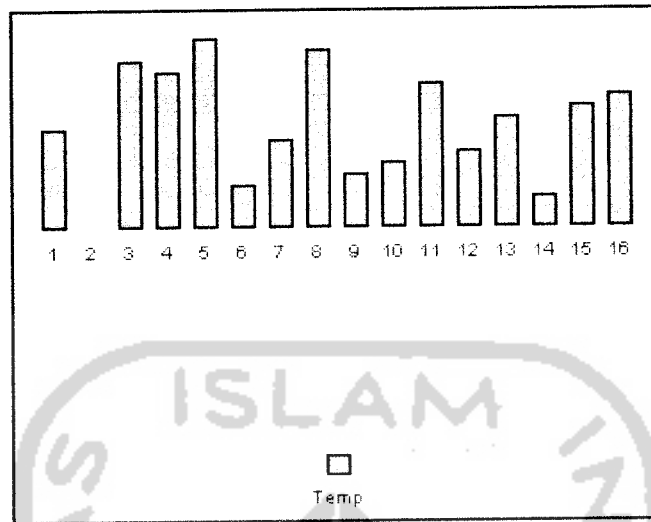
Langkah penyelesaiannya adalah sebagai berikut :

1. Pilih data nomer satu pada daftar yang hendak diurutkan (Gambar 2.7).



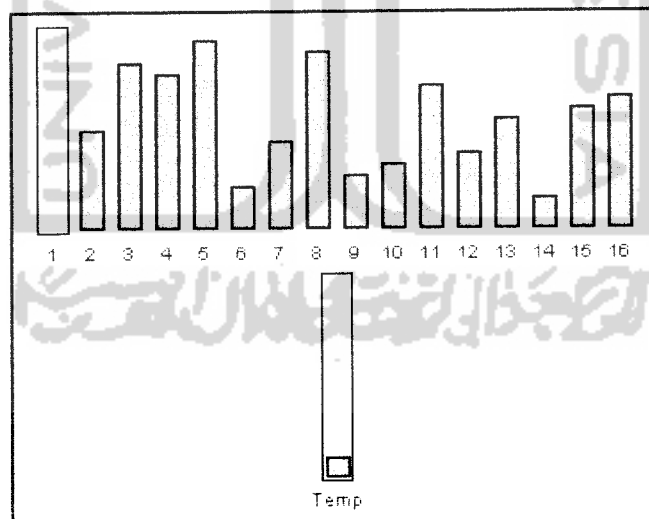
Gambar 2.7 Langkah Pertama Pengurutan *Insertion Sort*

2. Simpan data kedua pada penyimpanan temporary (Gambar 2.8).



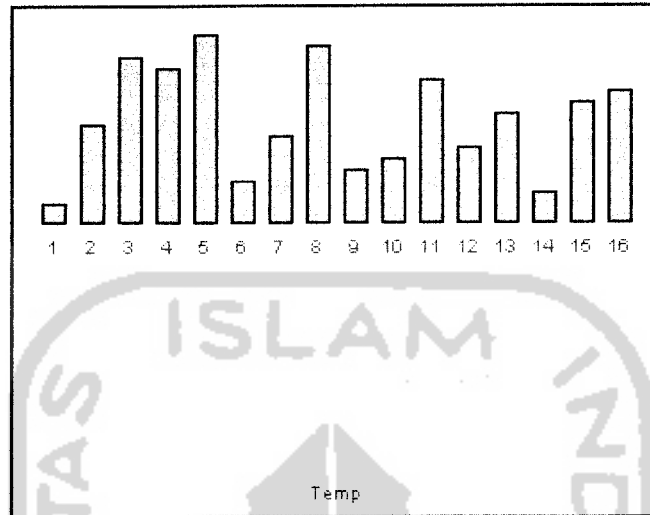
Gambar 2.8 Langkah Kedua Pengurutan

3. Jika data pertama lebih besar dari data kedua, posisi data pertama dipindah ke data kedua (Gambar 2.9).



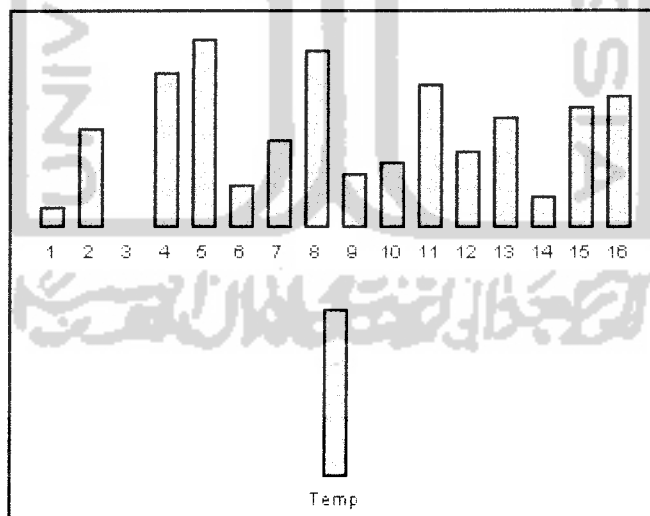
Gambar 2.9 Langkah Ketiga Pengurutan

4. Kelompokkan data nomer satu dan dua pada daftar yang hendak diurutkan (Gambar 2.10).



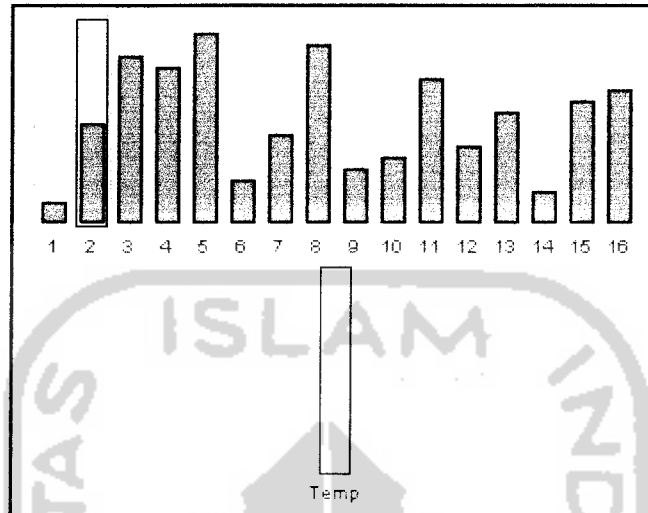
Gambar 2.10 Langkah Keempat Pengurutan

5. Simpan data ketiga pada penyimpanan temporary (Gambar 2.11).



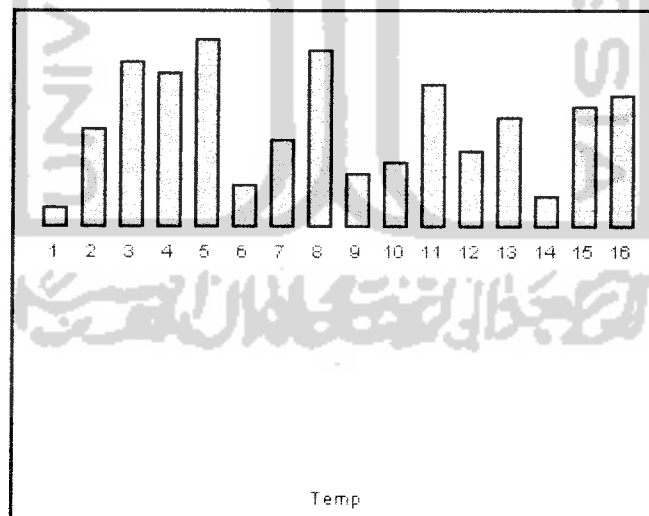
Gambar 2.11 Langkah Kelima Pengurutan

6. Bandingkan data ke tiga dengan data yang ke dua, jika lebih kecil maka tukarkan posisi data kedua dengan ketiga (Gambar 2.12).



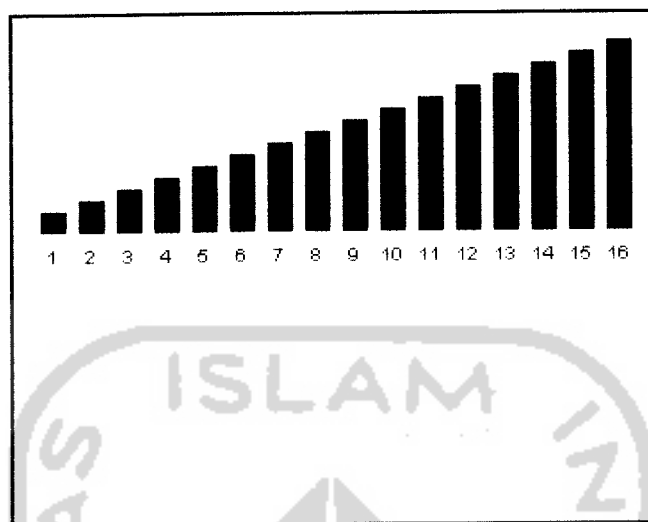
Gambar 2.12 Langkah Keenam Pengurutan

7. Ulangi langkah-langkah tersebut sampai data yang terakhir (Gambar 2.13).



Gambar 2.13 Langkah Ketujuh Pengurutan

8. Data telah terurutkan (Gambar 2.14).



Gambar 2.14 Pengurutan Selesai

Contoh data dan proses pengurutan insertion sort pada tabel 2.7.

31	25	12	22	11	Data Awal
31		12	22	11	Langkah 1
25	31		22	11	Langkah 2
12	25	31		11	Langkah 3
12	22	25	31		Langkah 4
11	12	22	25	31	Hasil Akhir

Tabel 2.7 Data dan proses pengurutan menggunakan insertion sort

2.1.8 Algoritma *Enhanced Quick Sort*

Algoritma ini merupakan pengembangan dari algoritma *Quick Short*.

Dengan cara kerja :

1. Tentukan nilai pivot/posisi tengah dari array yang diurutkan

2. Cari pada bagian sebelah kanan pivot data yang lebih besar. Jika lebih besar dari pivot maka posisi data ditukar dengan pivot.
3. Cari pada bagian sebelah kiri pivot data yang lebih kecil. Jika lebih kecil dari pivot maka posisi data ditukar dengan pivot.
4. Tempatkan nilai median pada bagian tengah list yang akan diurutkan.
5. Urutkan nilai paling kecil atau paling besar disebelah kiri maupun disebelah kanan nilai median.

Contoh data dan proses pengurutan pada tabel 2.8.

31	25	12	22	11	Data Awal
25	31		11	22	Langkah 1
11	12	22	25	31	Hasil Akhir

Tabel 2.8 Data dan proses pengurutan menggunakan enhanced quick sort

BAB III

ANALISIS KEBUTUHAN PERANGKAT LUNAK

3.1 Metode Analisis

Tahapan analisis digunakan untuk mengetahui dan menerjemahkan semua permasalahan serta kebutuhan perangkat lunak dan kebutuhan sistem yang akan dibangun. Tahap ini digunakan untuk mendapatkan data-data yang diperlukan untuk kepentingan membangun sistem. Hasil akhir dari analisis diharapkan akan didapatkan suatu sistem yang strukturnya dapat didefinisikan dengan baik dan jelas.

Dalam analisis kebutuhan ini akan digunakan metode yang berorientasi pada aliran proses (*Flow Chart Methodologies*). Metode ini secara umum didasarkan pada pemecahan dari sistem kedalam modul-modul berdasarkan tipe elemen data dan tingkah laku logika modul tersebut kedalam sistem. Dengan metodologi ini, sistem secara logika dapat digambarkan dari arus proses.

3.2 Spesifikasi Yang Akan Dikembangkan

Target sistem yang ingin dicapai adalah sebagai berikut:

1. Dapat melakukan pemodelan karakteristik masing algoritma pengurutan.
 2. Dapat dioperasikan dengan mudah dan fleksibel untuk dikembangkan.
- Dengan menggunakan proses-proses sistem yang telah dirancang,

diharapkan sistem ini mudah digunakan dan dioperasikan oleh siapa saja. Dan tidak menutup kemungkinan jika diadakan pengembangan yang sifatnya perubahan-perubahan seperti penambahan fasilitas atau menu, modifikasi, serta program ini sangat fleksibel.

3.3 Hasil Analisis Kebutuhan

Hasil analisis yang diperoleh dari sistem aplikasi simulasi pengurutan data ini adalah data masukan (*input*), antarmuka (*interface*), dan keluaran (*output*) dan proses yang dibutuhkan untuk membangun sistem.

3.3.1 Analisis Kebutuhan Data Masukan (*input*)

Dalam proses masukan (*input*) data pada aplikasi simulasi algoritma pengurutan ini terdiri dari:

1. Data integer berjumlah 300 data.
2. Data tersebut didapat dari pengacakan oleh sistem.

3.3.2 Analisis Kebutuhan Proses

Proses-proses yang dibutuhkan dalam aplikasi simulasi algoritma pengurutan data ini adalah :

1. Proses pemilihan algoritma pengurutan.
2. Proses pengacakan data.
3. Proses pengurutan data menggunakan data yang sudah diacak dan algoritma yang sudah dipilih.

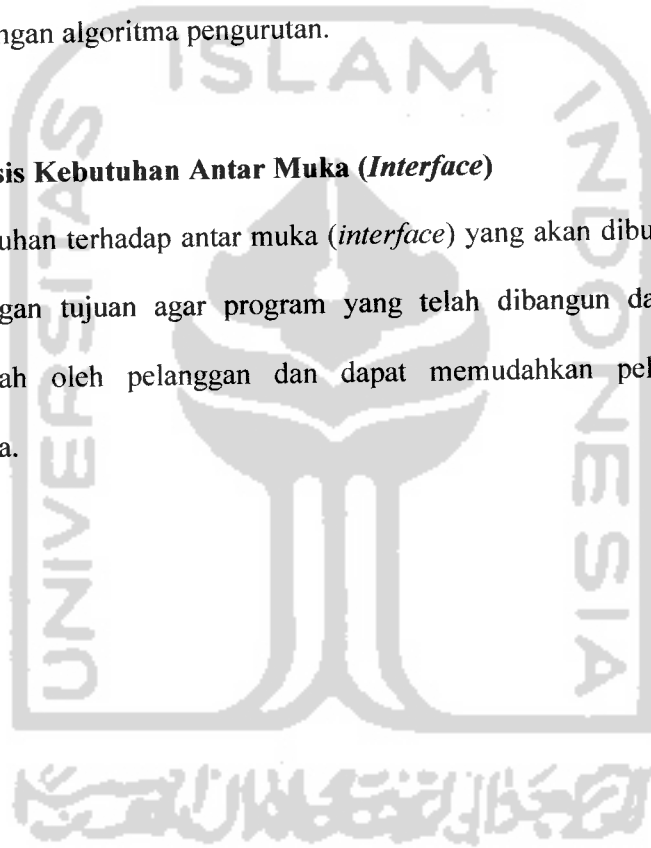
3.3.3 Analisis Kebutuhan Data Keluaran

Keluaran yang dihasilkan oleh sistem simulasi algoritma pengurutan adalah :

1. Memberikan visualisasi proses pengurutan data.
2. Menghitung waktu proses pengurutan data.
3. Menampilkan jumlah langkah dalam menyelesaikan pengurutan dengan algoritma pengurutan.

3.3.4 Analisis Kebutuhan Antar Muka (*Interface*)

Kebutuhan terhadap antar muka (*interface*) yang akan dibuat bersifat user friendly, dengan tujuan agar program yang telah dibangun dapat digunakan dengan mudah oleh pelanggan dan dapat memudahkan pelanggan dalam mengaksesnya.



BAB IV

PERANCANGAN PERANGKAT LUNAK

4.1 Metode Perancangan

Metode perancangan yang digunakan untuk merancang sistem aplikasi pengurutan data menggunakan metode perancangan berbasis bagan alir sistem perancangan terstruktur.

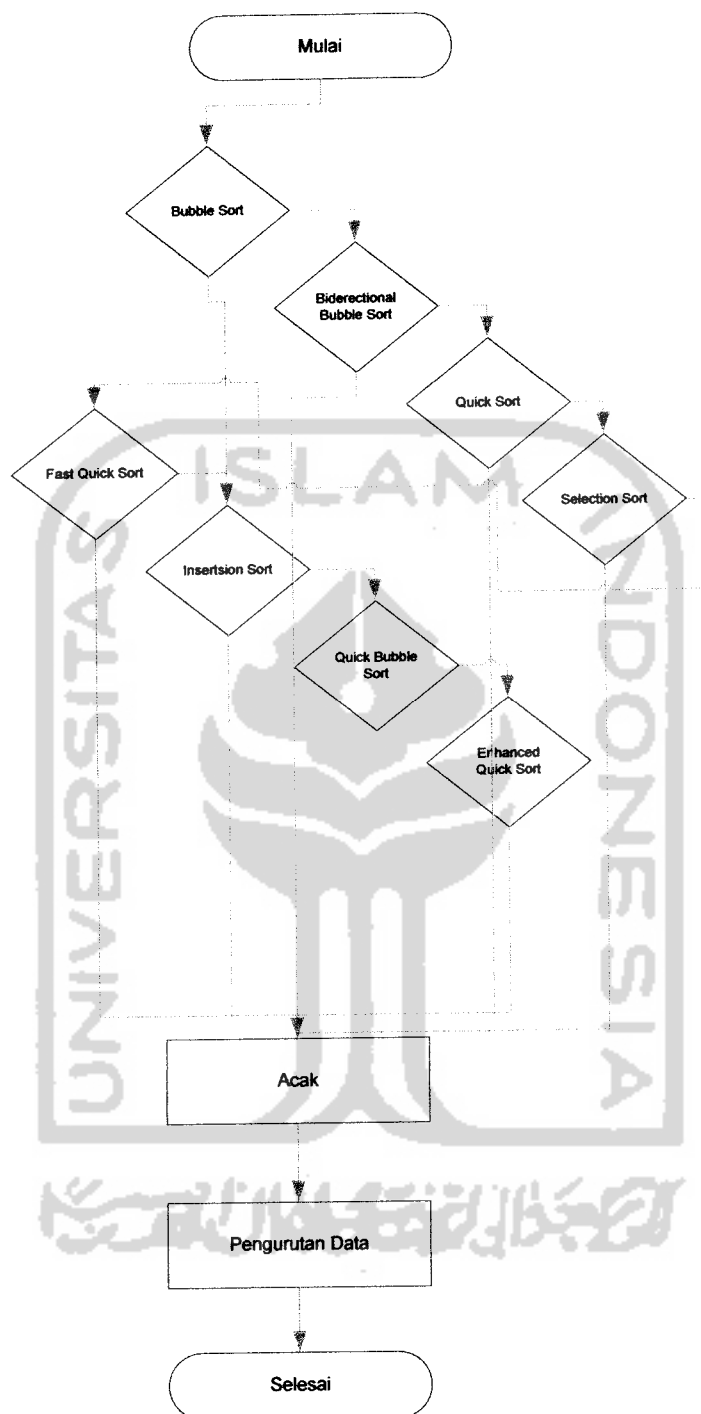
4.2 Hasil Perancangan

Hasil perancangan sistem ini berupa *flow chart* (diagram alir) yang menggambarkan aliran proses yang ada pada aplikasi dan bentuk antarmuka aplikasi.

4.2.1 Diagram Alir (*Flow Chart*)

Diagram konteks digunakan untuk menggambarkan bentuk yang paling global dari sistem yang ada *input* dan *output* (Gambar 4.1). Proses ini meliputi :

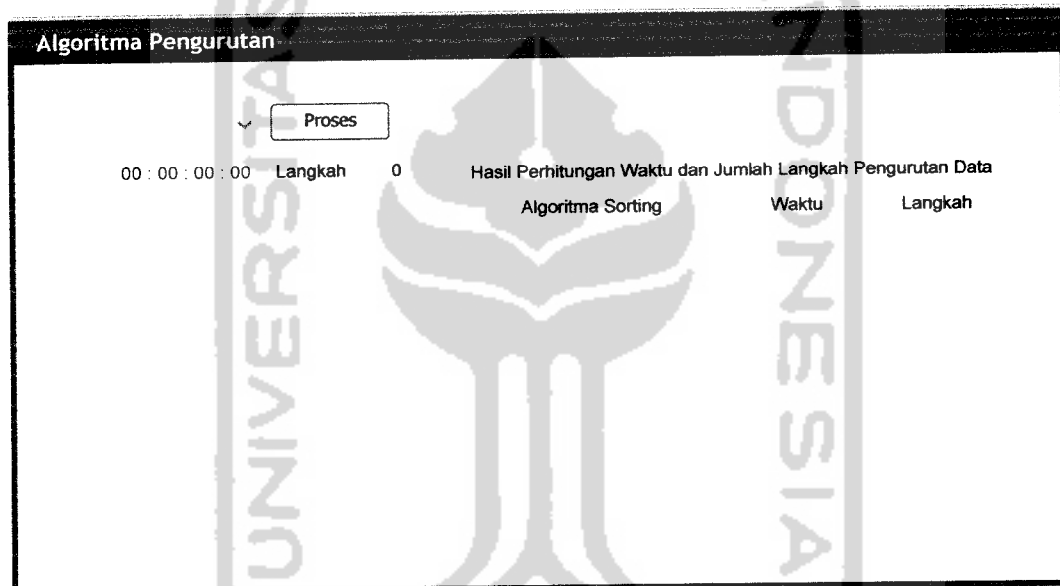
1. Inisialisasi aplikasi
2. Pemilihan Algoritma
3. Pengacakan data
4. Pengurutan data



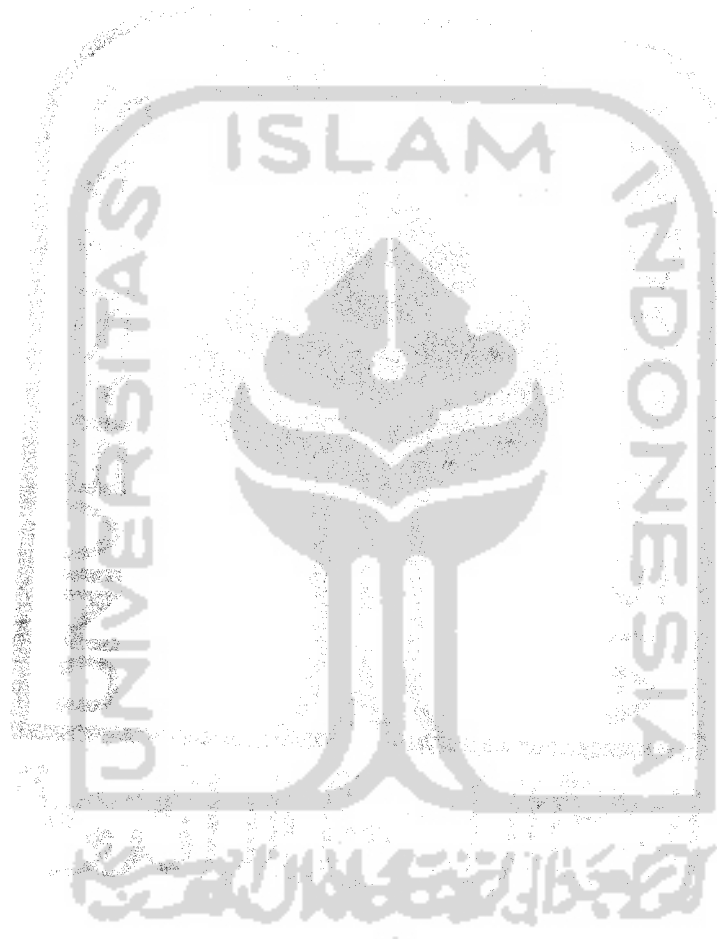
Gambar 4.1 Digram Alir Sistem

4.2.2 Antarmuka (*interface*)

Antarmuka (*interface*) (pada Gambar 4.2) dirancang agar pengguna dapat berinteraksi secara baik dengan aplikasi, sehingga tercipta komunikasi yang mudah dipahami. Komunikasi itu dapat terdiri dari proses memasukkan data yang menampilkan keluaran.



Gambar 4.2 Rancangan Aplikasi.



BAB V

IMPLEMENTASI PERANGKAT LUNAK

5.1 Batasan Implementasi

Batasan implementasi meliputi batasan minimal untuk perangkat keras (*hardware*) atau perangkat lunak (*software*) yang diperlukan agar sistem yang dibuat bisa berjalan dengan baik.

5.1.1 Analisis Kebutuhan Perangkat Lunak

Sistem aplikasi simulasi algoritma pengurutan ini memiliki beberapa kriteria perangkat lunak yang akan digunakan agar sistem berjalan dengan baik, yaitu :

- a. Microsoft Windows98 atau Xp yaitu sistem operasi yang digunakan.
- b. Netbeans 5.5 digunakan sebagai alat untuk mendevlope perangkat lunak.
- c. Java bahasa pemrograman yang digunakan untuk membangun aplikasi.
- d. *Web browser* yang mendukung applet java.

5.1.2 Analisis Kebutuhan Perangkat Keras

Kebutuhan sistem yang akan digunakan pada PC untuk dapat menjalankan sistem aplikasi ini, memiliki spesifikasi minimal sebagai berikut :

- a. Processor Intel Pentium 233 atau lebih tinggi sejenisnya

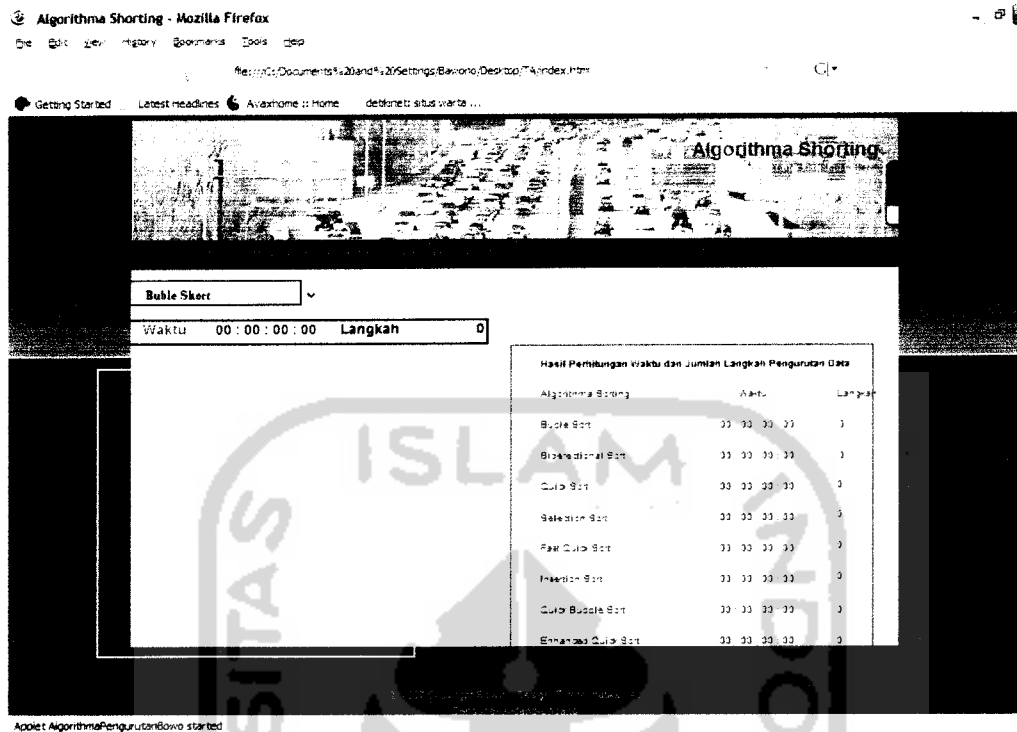
- b. RAM 64 MB
- c. Kartu VGA

5.2 Tampilan Aplikasi Simulasi dan Visualisasi Pengurutan Data.

5.2.1 Tampilan Menu

Halaman web simulasi algoritma pengurutan merupakan tampilan yang pertama kali muncul ketika pelanggan mengakses halaman web, bentuk dari tampilan ini terlihat pada gambar 5.1. Ada 5 bagian penting dari tampilan applet yaitu :

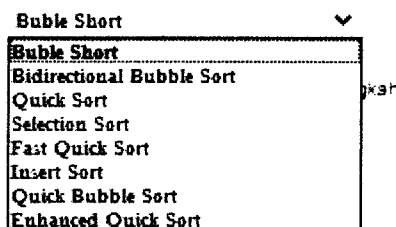
1. Kotak warna merah, *combobox* yang berfungsi untuk memilih metode pengurutan data.
2. Kotak warna abu-abu, untuk mengacak data yang akan diurutkan.
3. Kotak kuning, visualisasi data yang akan diurutkan
4. Kotak biru, daftar hasil pengurutan
5. Kotak berwarna hitam, waktu dan jumlah langkah pengurutan.



Gambar 5.1 Tampilan Menu

5.2.2 Pemilihan Algoritma Pengurutan

Sembelum digunakan untuk melakukan simulasi, harus dilakukan pemilihan algoritma yang akan digunakan untuk pengurutan data. Dalam proses pemilihan algoritma ini sekaligus melakukan pengacakan data yang akan digunakan dalam simulasi. Pemilihan algoritma dilakukan dengan memilih daftar algoritma yang ada pada *combo box* aplikasi (Gambar 5.2).



Gambar 5.2 *Combo Box* yang Pemilihan Algoritma.

5.2.3 Proses Pengacakan Data

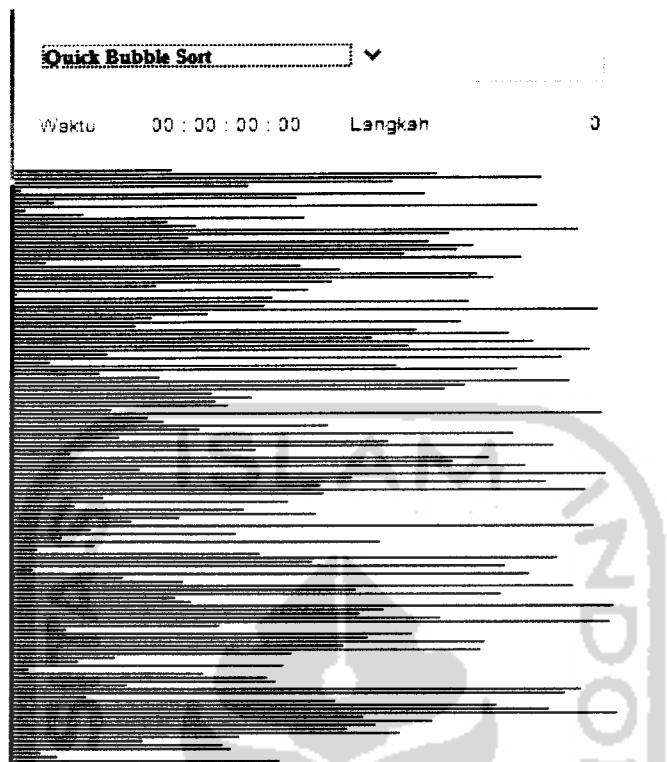
Setelah aplikasi dijalankan, pada waktu pemilihan metode pengurutan otomatis data langsung diacak. Setelah proses pemilihan metode pengurutan dan menjalankan simulasi, tombol “Proses” akan menjadi aktif dan apabila kita memilih metode yang berbeda pada *combo box* pengacakan tidak dilakukan (Gambar 5.3).

```

/* Prosedur Pengacakan angka yang akan diurutkan
*/
void acak() {
    pilih = true;
    initialSize = getSize();
    a = new int[(initialSize.height-80) / 2];
    double f = (initialSize.width-466) / (double) a.length;
    for (int i = a.length; --i >= 0; ) {
        a[i] = (int)(i * f);
    }
    for (int i = a.length; --i >= 0; ) {
        int j = (int)(i * Math.random());
        int t = a[i];
        a[i] = a[j];
        a[j] = t;
    }
}

```

Proses pengacakan data akan dilakukan apabila kita menekan tombol “Proses”.



Gambar 5.3 Visualisasi Data Hasil Proses Pengacakan dan Kondisi Tombol Proses yang Non-Aktif Sebelum Proses Pengurutan.

5.2.4 Proses Pengurutan Data

Untuk memilih metode pengurutan dapat memilih metode pada *Combobox* disamping kiri tombol proses (Gambar 5.4). Proses pengurutan data dilakukan dengan melakukan klik pada visualisasi data yang akan diacak. Kemudian aplikasi secara otomatis akan melakukan proses pengurutan data dan menghitung waktu dan jumlah langkah proses pengurutan data. Proses pengurutan ini berakhir dengan ditandai dengan *message box* yang berisi informasi bahwa proses pengurutan telah selesai (lihat Gambar 5.5).

Fast Quick Sort	▼	Proses
Waktu	00 : 00 : 05 : 18	Langkah
		245

Gambar 5.4 Hasil Penghitungan Waktu dan Jumlah Langkah Algoritma Pengurutan



Gambar 5.5 Message box Informasi Selesainya Algoritma Pengurutan.

Setelah proses pengurutan dilakukan, hasil perhitungan waktu dan langkah ditampilkan pada daftar disamping kanan pada gambar 5.6. Daftar hasil pengurutan data terdiri dari nama algoritma, waktu dan langkah. Perhitungan waktu berisi lama proses pengurutan dan langkah merupakan perhitungan jumlah langkah proses pengurutan data. Daftar ini terdiri dari delapan algoritma pengurutan data yang difasilitasi oleh aplikasi.

Hasil Perhitungan Waktu dan Jumlah Langkah Pengurutan Data

Algoritma Sorting	Waktu	Langkah
Buble Sort	00 : 03 : 48 : 32	10797
Bidirectional Sort	00 : 03 : 12 : 88	9072
Quick Sort	00 : 00 : 38 : 17	1704
Selection Sort	00 : 04 : 00 : 60	11328
Fast Quick Sort	00 : 00 : 05 : 69	284
Insertion Sort	00 : 01 : 58 : 27	5669
Quick Bubble Sort	00 : 00 : 07 : 15	337
Enhanced Quick Sort	00 : 00 : 08 : 29	298

Gambar 5.6 Laporan Waktu Proses Pengurutan Data dan Jumlah Langkah.

Prosedur yang digunakan dalam metode *QuickSort*

```
private void QuickSort(int a[], int l, int r) throws Exception
```

```
{
    int M = 4;
    int i;
    int j;
    int v;
    if ((r-l)>M)
    {
        i = (r+l)/2;
        if (a[l]>a[i]) swap(a,l,i);
        if (a[l]>a[r]) swap(a,l,r);
        if (a[i]>a[r]) swap(a,i,r);

        j = r-1;
        swap(a,i,j);
        i = l;
        v = a[j];
        for(;;)
        {
            while(a[++i]<v);
            while(a[--j]>v);
            if (j<i) break;
            swap (a,i,j);
            pause(i,j);
            if (stopRequested) {
```

```
        return;  
    }  
    }  
    swap(a,i,r-1);  
    pause(i);  
    QuickSort(a,l,j);  
    QuickSort(a,i+1,r);  
} }
```



BAB VI

ANALISIS KINERJA PERANGKAT LUNAK

6.1 Pengujian Sistem

Dalam tahap ini akan dijelaskan mengenai pengujian program Aplikasi yang digunakan pada sistem aplikasi simulasi algoritma pengurutan. Dengan pengujian ini diharapkan dapat diketahui kekurangan-kekurangan dari sistem untuk kemudian diperbaiki sehingga kesalahan dari sistem dapat diminimalisasi atau bahkan dihilangkan.

Pengujian kinerja sistem dilakukan dengan cara memberikan input kedalam sistem. Sistem dikatakan baik apabila sistem dapat menangani/mengantisipasi input data yang tidak benar. Sebaliknya, sistem dikatakan tidak baik bila tidak dapat menangani/mengantisipasi input data yang tidak benar. Penanganan kesalahan pada sistem ini dilakukan dengan memberikan dalam bentuk pesan peringatan yang berisikan informasi tentang keharusan untuk mengisikan data tertentu.

Dalam hal ini pengujian kinerja untuk mendeteksi kesalahan pada sistem terdiri dari pengujian normal dan pengujian tidak normal.

6.1.1 Proses Pemilihan Algoritma

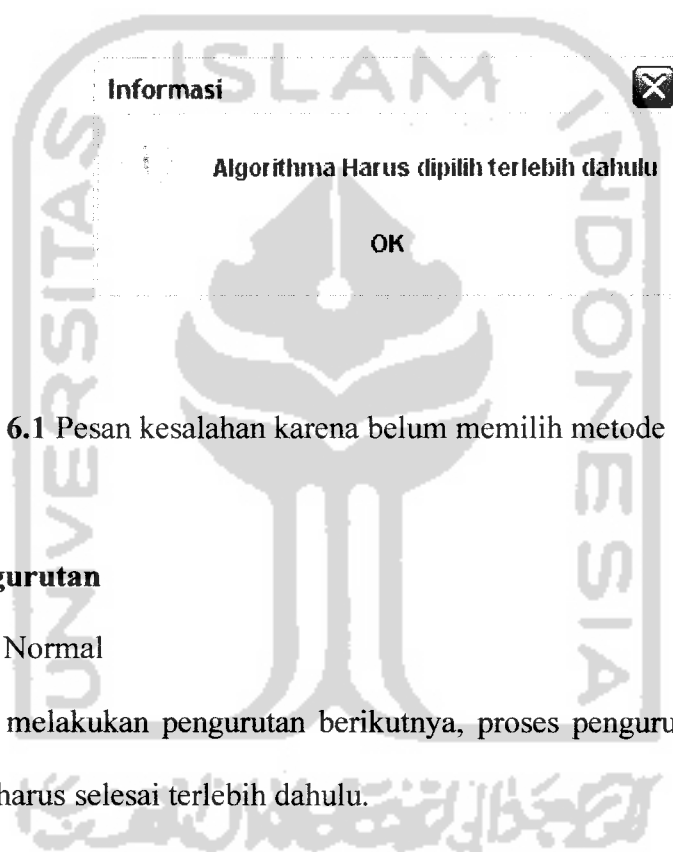
pengujian dari proses login adalah sebagai berikut :

1. Prosedur Normal

Pada *Combo box* pemilihan metode pengurutan haruslah dipilih terlebih dahulu. Setelah dipilih maka akan tampak gambar data yang akan diurutkan.

2. Prosedur Tidak Normal

Pada *Combo box* tidak dipilih terlebih dahulu, sehingga memunculkan pesan kesalahan (gambar 6.1).



Gambar 6.1 Pesan kesalahan karena belum memilih metode pengurutan

6.1.2 Proses Pegurutan

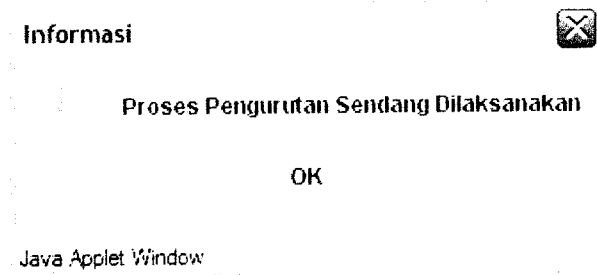
1. Prosedur Normal

Sebelum melakukan pengurutan berikutnya, proses pengurutan yang sedang berjalan harus selesai terlebih dahulu.

2. Prosedur Tidak Normal

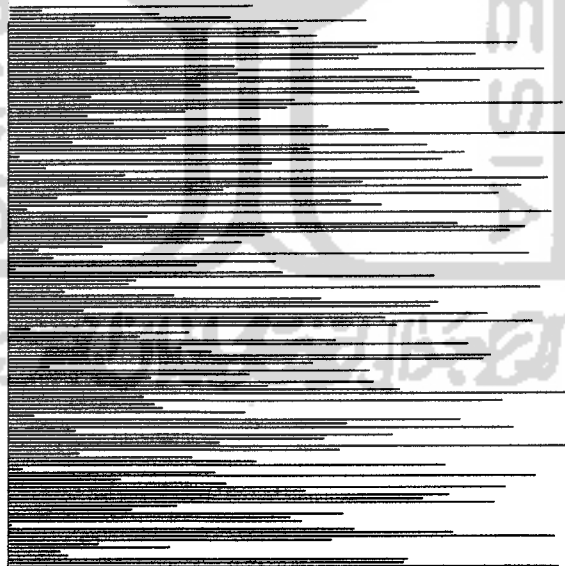
Sebelum proses pengurutan selesai dilakukan proses pengurutan berikutnya.

Ini akan memunculkan *message box* yang berisi informasi yang menyatakan bahwa proses pengurutan belum selesai (Gambar 6.2).

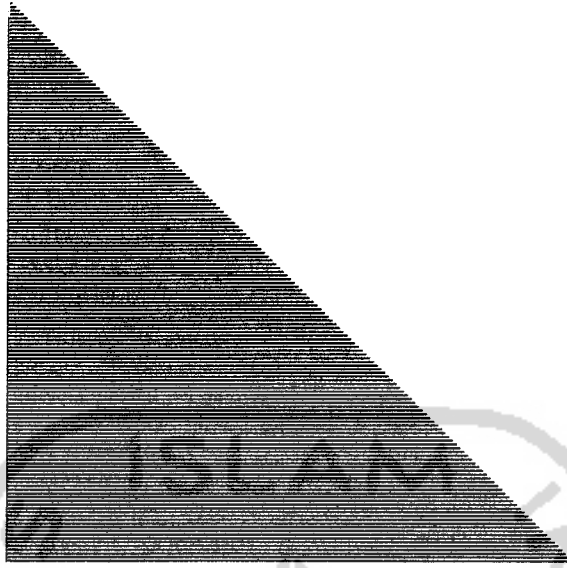


Gambar 6.2 Pesan kesalahan proses pengurutan sedang berjalan

Data yang digunakan dalam proses pengurutan data, diambil secara acak untuk kemudian divisualisasikan dengan garis sesuai dengan nilai data. Untuk data yang bernilai kecil divisualisasikan dengan garis pendek sedangkan data yang bernilai lebih besar divisualisasikan dengan garis yang lebih panjang (gambar 6.3).



Gambar 6.3 Visualisasi Data Sebelum Proses Pengurutan Data.



Gambar 6.4 Visualisasi Data Setelah Proses Pengurutan Data.

Setelah proses pengurutan selesai posisi visualisasi data menjadi seperti gambar 6.4. Hasil perhitungan waktu dan jumlah langkah ditampilkan pada gambar 6.5. Terdiri dari dua jenis perhitungan, waktu dan langkah. Perhitungan waktu dihitung berdasarkan waktu mulai sampai dengan waktu selesainya proses pengurutan data. Perhitungah dihitung berdasarkan banyaknya langkah dalam proses pengurutan data. Untuk kasus ini algoritma *fast quick sort*, merupakan algoritma yang paling cepat dan paling sedikit jumlah langkah penyelesaiannya dengan waktu 5, 59 detik dan jumlah langkah 264. Sedangkan algoritma *selection sort* merupakan algoritma paling lama dan paling banyak jumlah langkah dengan waktu 4 menit 50 milidetik dengan jumlah langkah 11325.

Hasil Perhitungan Waktu dan Jumlah Langkah Pengurutan Data

Algoritma Sorting	Waktu	Langkah
Bubble Sort	00 : 02 : 49 : 32	10797
Bidirectional Sort	00 : 02 : 12 : 88	9072
Quick Sort	00 : 00 : 28 : 17	1704
Selection Sort	00 : 04 : 00 : 60	11226
Fast Quick Sort	00 : 00 : 06 : 69	284
Insertion Sort	00 : 01 : 59 : 27	6669
Quick Bubble Sort	00 : 00 : 07 : 15	227
Enhanced Quick Sort	00 : 00 : 06 : 29	282

Gambar 6.5 Hasil Proses Simulasi Pengurutan Data

6.2 Kinerja Perangkat Lunak

Perangkat lunak ini memodelkan data yang diurutkan dengan menggambarkan dengan panjang garis yang berbeda-beda untuk nilai yang berbeda. Data yang digunakan dalam proses simulasi adalah data yang diambil secara acak oleh komputer sehingga kita tidak perlu melakukan pemasukan data secara manual. Data yang digunakan sepanjang 300 data, yang disimpan dalam bentuk array. Perhitungan waktu sebagai salah satu hasil dari perangkat lunak didasarkan pada waktu dimulai proses pengurutan sampai dengan waktu selesainya proses pengurutan data. Sedangkan proses perhitungan jumlah langkah didasarkan pada jumlah langkah dalam menyelesaikan proses pengurutan.

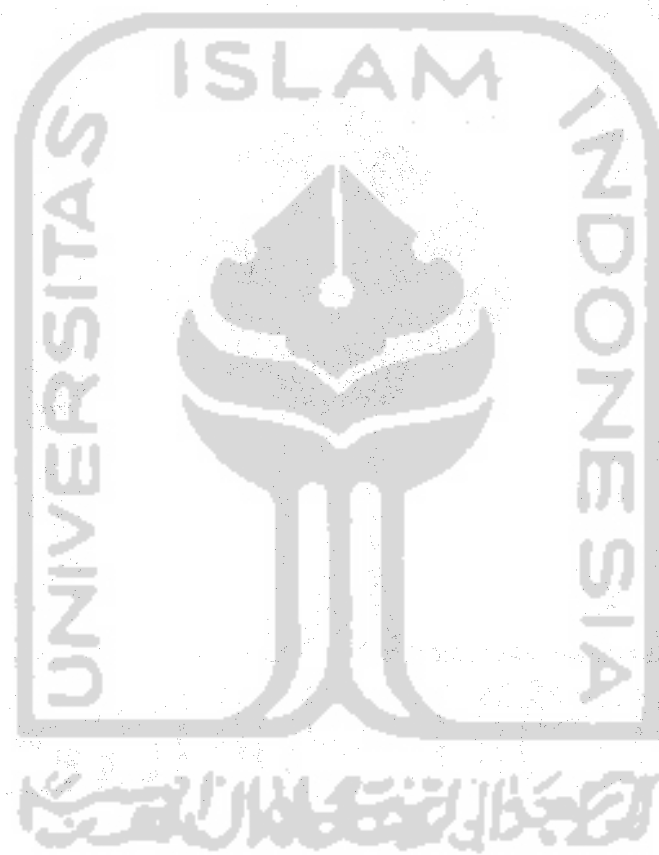
Kelebihan perangkat lunak

1. Input data merupakan hasil pengacakan data secara random
2. Data yang akan diurutkan divisualisasikan dengan garis yang memiliki panjang yang berbeda sehingga dapat dilihat proses pengurutan datanya.
3. Dapat menghitung jumlah waktu dan langkah dalam proses pengurutan.

Kekurangan perangkat lunak

1. Tidak dapat melihat nilai sebenarnya data dalam proses pengurutan
2. Tidak dapat melakukan simulasi pengurutan secara bertahap.





BAB VII

PENUTUP

7.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian yang berjudul “STUDI BEBERAPA METODE SORTING”:

1. Simulasi dan visualisasi algoritma pengurutan berbasis applet java dapat digunakan untuk memodelkan beberapa metode sorting.
2. Hasil perhitungan waktu dan jumlah efisiensi langkah dengan menggunakan data yang diacak sebanyak 300 data menunjukkan algoritma fast quick sort memiliki waktu dan jumlah langkah yang paling efisien.
3. Untuk data yang besar selisih waktu dan proses perpindahan data menjadi sangat besar. Sedangkan untuk data berukuran kecil misalnya 5 data, proses pengurutan memiliki waktu yang hampir sama.

7.2 Saran

Saran yang diberikan berdasarkan hasil penelitian adalah sebagai berikut:

1. Perlu adanya peningkatan dan pengembangan lebih lanjut terhadap aplikasi Simulasi dan visualisasi algoritma pengurutan misalnya dengan menambahkan algoritma atau dengan memberikan visualisasi nilai data yang sedang diurutkan pada proses pengurutan data.

2. Perlu adanya penambahan fitur sehingga jumlah data yang diurutkan bisa lebih fleksibel.
3. Penambahan fitur untuk menampilkan nilai data yang akan diurutkan dan fasilitas untuk merubah nilai data yang diurutkan.



DAFTAR PUSTAKA

- [DEI05] Deitel H. M., *Java How to Program*, Prentice Hall, Sixth Edition, 2005.
- [HAR05] Harrison, 2005 [http: // www.cs.ubc.ca/~harrison/Java/sorting-demo.html](http://www.cs.ubc.ca/~harrison/Java/sorting-demo.html).
- [PAT05] Patrick Keegan, et al., *NetBeans 5.0 IDE Field Guide*, Prentice Hall, First Edition, 2005.
- [PAT06] Patrick Keegan, et al., *NetBeans 5.0 IDE Field Guide*, Prentice Hall, Second Edition, 2006.
- [SAN95] Santoso, P. Insap, Ir., Msc., *Struktur Data Menggunakan Turbo Pascal 6.0*, Yogyakarta : Andi Offset, 1995.
- [SCH99] Schild, Herbert. *Java 2 The Complete Reference*, New Yorks McGraw-Hill, 1999.
- [SRI07] Sri Hartati, B. Herry Suharto, M. Soesilo Wijono. *Pemrograman GUI Swing Java dengan NetBeans 5*, Yogyakarta : Andi Yogyakarta, 2007.