

Komparasi Stabilitas Eksekusi Kode Bahasa Pemrograman .Net C# Versi 4.0.3019 Dengan Google Golang Versi 1.4.2 Menggunakan Algoritma Bubble Sort dan Insertion Sort

Moh Arif Jauhari¹, Dini Hamidin², Marwanto Rahmatuloh³

^{1,2} Program Studi D3 Teknik Informatika,
Politeknik Pos Indonesia

¹moh.arif.jauhari@gmail.com, ²Dini.Hamidin@gmail.com, ³m_rahmatuloh@yahoo.com

ABSTRAK

PT. Xsis Mitra Utama yang didirikan pada tahun 2005, memiliki salah satu klien yaitu PT. Navcore Nextology yang bergerak dalam industri teknologi informasi dan telekomunikasi. Bahasa pemrograman yang umum digunakan di PT.Navcore Nextology dalam membangun perangkat lunak sistem adalah .Net C# (C Sharp) yang dikembangkan oleh Microsoft. Bahasa pemrograman lain yang akan digunakan untuk membangun perangkat lunak sistem adalah bahasa pemrograman Go yang dibuat oleh Google. Sebelum Go digunakan dalam pembangunan perangkat lunak sistem, akan dilakukan beberapa pengujian terhadap Go dengan tujuan untuk mengetahui stabilitas eksekusi kode dari Go. Pengujian dilakukan dengan menghitung waktu eksekusi kode algoritma bubble sort dan insertion sort dengan parameter yang sama pada C# dan Go. Hasil dari pengujian menunjukkan Go memiliki waktu eksekusi yang lebih cepat dari waktu eksekusi C#.

Kata Kunci : C#, Go, komparasi, bubble sort, insertion sort

I. PENDAHULUAN

A. Latar Belakang

PT. Xsis Mitra Utama didirikan pada tahun 2005, dengan tujuan menyediakan jasa di bidang Teknologi Informasi. PT. Xsis Mitra Utama memiliki banyak klien dimana masing masing klien memiliki manajemen yang telah diatur dan disepakati oleh kedua belah pihak (PT. Xsis Mitra Utama dan Klien). Untuk memperoleh kesepakatan kerja dengan klien dibutuhkan tenaga atau sumber daya manusia atau karyawan yang telah di uji dan di dinilai mampu dalam menyelesaikan tugas yang akan diberikan oleh klien.

Salah satu klien yang dimiliki oleh PT. Xsis Mitra Utama adalah PT. Navcore Nextology yang beralamat di Setiabudi Building 2 Lt.03 JL. HR Rasuna Said Kav. 62 Kunigan Jakarta Selatan. PT. Navcore Nextology bergerak dalam industri teknologi informasi dan telekomunikasi.

Setiap karyawan PT. Xsis Mitra Utama yang telah di tempatkan dalam proyek klien akan mendapatkan tugas dalam bidang pembangunan perangkat lunak sistem. Untuk membangun suatu perangkat lunak sistem dibutuhkan suatu bahasa pemrograman, Darmawan [1] bahasa pemrograman adalah suatu bahasa yang dapat dimengerti oleh manusia dan komputer.

Bahasa pemrograman yang umum digunakan di PT.Navcore Nextology untuk membangun sisi server perangkat lunak sistem adalah .Net C# (C Sharp), Nugroho [2] C# adalah bahasa pemrograman baru yang diciptakan oleh Microsoft (dikembangkan di bawah kepemimpinan Anders Hejlsberg yang notabene juga telah menciptakan berbagai macam bahasa pemrograman

termasuk Borland Turbo C++ dan Borland Delphi). Bahasa C# juga telah distandarisasi secara internasional oleh ECMA.

Bahasa pemrograman lain yang akan digunakan di PT.Navcore Nextology untuk membangun sisi server perangkat lunak sistem adalah bahasa pemrograman Google golang yang biasanya disebut Go. Summerfield [3] Bahasa pemrograman Go sama seperti bahasa pemrograman C, yang menjadi bahasa pemrograman ringan dan efisien dengan fasilitas bahasa pemrograman tingkat rendah yang nyaman seperti pointer, namun Go juga memberikan banyak fitur yang berhubungan dengan bahasa pemrograman tingkat menengah maupun tingkat tinggi, seperti Unicode string, struktur data built-in yang tangguh, duck typing, garbage collection, dan dukungan konkurensi tingkat tinggi yang menggunakan komunikasi data dan penguncian secara bersamaan, selain itu Go juga memiliki library standar yang besar dan luas.

Sebelum bahasa pemrograman Go digunakan dalam pembangunan sisi server perangkat lunak sistem di PT. Navcore Nextology, dilakukanlah beberapa pengujian terhadap bahasa pemrograman Go oleh tim di divisi pengembangan framework PT.Navcore Nextology, untuk mengetahui apakah bahasa pemrograman Go dapat digunakan untuk membangun sisi server perangkat lunak sistem. Salah satu jenis pengujian yang dilakukan adalah komparasi stabilitas eksekusi kode antara bahasa C# sebagai pembanding dari proses komparasi bahasa pemrograman dan bahasa pemrograman Go sebagai bahasa yang akan di uji untuk

mengetahui lama waktu eksekusi kode dari masing masing bahasa pemrograman.

Oleh karena itu dilakukan analisis komparasi stabilitas eksekusi kode bahasa pemrograman .NET C# versi 4.0.3019 dan Google golang versi 1.4.2 untuk mengetahui kecepatan pengeksesian kode dari masing masing bahasa pemrograman dengan mengeksekusi proses algoritma bubble sort dan insertion sort.

B. Identifikasi Masalah

Adapun identifikasi masalah untuk memperjelas tujuan yang akan dicapai dan dirumuskan berdasarkan latar belakang di atas, yaitu:

1. Akan dilakukan proses komparasi bahasa pemrograman C# versi 4.0.3019 (sebagai pembanding) dan bahasa pemrograman Go versi 1.4.2 agar dari hasil komparasi dapat diketahui apakah bahasa pemrograman Go dapat digunakan untuk membangun sisi server perangkat lunak di PT. Navcore Nextology.
2. Bagaimana cara dan proses komparasi bahasa pemrograman dilakukan untuk mengetahui kecepatan pengeksesian kode dari bahasa pemrograman Go yang akan dikomparasi dengan bahasa pemrograman C# yang akan mengeksekusi algoritma bubble sort dan insertion sort.
3. Bagaimana cara menyusun masing masing algoritma dalam bahasa pemrograman C# dan Go serta data pengujian untuk algoritma bubble sort dan insertion sort agar dapat digunakan dalam proses komparasi bahasa pemrograman C# dan Go.

C. Tujuan

Adapun tujuan dari komparasi bahasa pemrograman C# dan Go, yaitu:

1. Komparasi ini bertujuan untuk mengetahui kecepatan pengeksesian kode dalam lingkungan pengembangan bahasa pemrograman Go, agar dapat dijadikan pertimbangan untuk pembangunan sisi server perangkat lunak sistem menggunakan bahasa pemrograman Go di PT. Navcore Nextology.
2. Mengetahui salah satu cara dalam melakukan proses komparasi bahasa pemrograman untuk mengetahui cara menghitung dan membandingkan kecepatan pengeksesian kode suatu bahasa pemrograman.

D. Ruang Lingkup

Untuk mencapai target dan tujuan dari pelaksanaan dalam komparasi Bahasa pemrograman ini, maka terdapat suatu batasan masalah agar tidak keluar dari lingkup tujuan yang ingin dicapai adalah sebagai berikut:

1. Proses komparasi yang akan dilakukan menggunakan bahasa pemrograman .Net C# versi 4.0.3019 dengan google golang versi 1.4.2
2. Bahasa pemrograman C# merupakan bahasa pemrograman yang dipilih oleh PT.Navcore Nextology sebagai pembanding dalam proses komparasi dengan bahasa pemrograman Go karena bahasa pemrograman C# merupakan bahasa

pemrograman sisi server yang sering digunakan dalam pembangunan perangkat lunak sistem di PT. Navcore Nextology.

3. Proses komparasi yang dilakukan adalah pengukuran kecepatan eksekusi kode menggunakan fitur diagnosa penghitung waktu dari masing masing Bahasa pemrograman.
4. Kode yang akan dieksekusi dan di hitung kecepatannya dalam bahasa pemrograman .Net C# versi 4.0.3019 dengan google golang versi 1.4.2 adalah kode dari algoritma pengurutan bubble sort dan insertion sort.
5. Hasil dari proses komparasi adalah waktu pengeksesian kode dari masing masing bahasa pemrograman.
6. Hasil dari proses komparasi kode hanya akan dijadikan sebagai salah satu pertimbangan PT. Navcore Nextology apakah bahasa pemrograman golang dapat digunakan dalam pembangunan sisi server perangkat lunak sistem di PT. Navcore Nextology..

II. LANDASAN TEORI

A. Algoritma

Algoritma sering dipakai dalam metode perhitungan (komputasi) secara umum, menurut Zarlis [4] Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis. Kata logis merupakan kata kunci dalam algoritma. Langkah-langkah dalam algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar. Dalam beberapa konteks, algoritma adalah spesifikasi urutan langkah untuk melakukan pekerjaan tertentu. Ada beberapa pertimbangan menurut Zarlis [4] dalam pemilihan suatu algoritma :

1. Algoritma haruslah benar. Artinya algoritma akan memberikan keluaran yang dikehendaki dari sejumlah masukan yang diberikan. Tidak peduli sebegus apapun algoritma, kalau memberikan keluaran yang salah, pastilah algoritma tersebut bukanlah algoritma yang baik.
2. Mengetahui seberapa baik hasil yang dicapai oleh algoritma tersebut. Hal ini penting terutama pada algoritma untuk menyelesaikan masalah yang memerlukan aproksimasi hasil (hasil yang hanya berupa pendekatan). Algoritma yang baik harus mampu memberikan hasil yang sedekat mungkin dengan nilai yang sebenarnya.
3. Efisiensi algoritma. Efisiensi algoritma dapat ditinjau dari dua hal yaitu efisiensi waktu dan memori. Meskipun algoritma memberikan keluaran yang benar (paling mendekati), tetapi jika kita harus menunggu berjam-jam untuk mendapatkan keluarannya, algoritma tersebut biasanya tidak akan dipakai, setiap orang menginginkan keluaran yang cepat. Begitu juga dengan memori, semakin besar memori yang terpakai maka semakin buruklah algoritma tersebut.

B. Algoritma Pengurutan (*Sorting*)

Pengurutan atau sorting (dalam bahasa Inggris) menurut Sedgewick [5] merupakan suatu proses menata ulang urutan suatu objek, sehingga masing-masing objek tersebut ditempatkan dalam beberapa urutan logis. Untuk melakukan proses pengurutan suatu objek, Sedgewick [5] biasanya menggunakan algoritma pengurutan. Algoritma pengurutan menurut Wajhillah [6] adalah salah satu contoh solusi algoritmik yang kaya dimana satu persoalan (pengurutan), dapat dilakukan dengan puluhan macam metode algoritma. Berikut metode sorting yang akan digunakan untuk proses komparasi antara C# dan Go

C. Metode Pengurutan *Bubble Sort*

Di antara beberapa algoritma pengurutan yang ada, algoritma bubble sort merupakan teknik yang paling sederhana. Orwant [7] metode pengurutan menggunakan bubble sort akan memeriksa masing-masing kumpulan obyek atau array yang akan diurutkan, menukar element yang saling bersebelahan dalam array jika mereka dalam urutan yang salah, sampai tidak ada lagi penukaran yang diperlukan, namun jika memungkinkan hindari penggunaan bubble sort sebagai algoritma pengurutan untuk tujuan umum, karena memiliki kinerja terburuk dalam melakukan pengurutan. Kinerja rata-rata pengurutan yang buruk disebabkan karena proses bubble sort harus memeriksa setiap elemen yang akan diurutkan.

D. Metode Pengurutan *Insertion sort*

Metode pengurutan yang kedua yang digunakan dalam proses komparasi bahasa pemrograman C# dan Go adalah metode pengurutan insertion sort. Menurut Orwant [7] metode pengurutan menggunakan insertion sort akan memeriksa setiap elemen, menemukan yang terkecil atau yang terbesar sesuai kebutuhan jenis pengurutan, dan menyisipkan dalam tempat yang tepat. Ketika setiap tempat yang benar telah ditemukan, elemen yang telah di sorting akan digeser untuk membuat ruang bagi elemen yang akan ditempatkan, Orwant [7] juga menambahkan algoritma pengurutan menggunakan insertion sort memiliki kinerja rata-rata terburuk dalam proses pengurutan data namun sedikit lebih cepat dibandingkan dengan algoritma pengurutan menggunakan bubble sort.

E. Golang

Summerfield [3] Bahasa pemrograman Go sama seperti bahasa pemrograman C, yang menjadi bahasa pemrograman ringan dan efisien dengan fasilitas bahasa pemrograman tingkat rendah yang nyaman seperti pointer, namun Go juga memberikan banyak fitur yang berhubungan dengan bahasa pemrograman tingkat menengah maupun tingkat tinggi, seperti Unicode string, struktur data built-in yang tangguh, duck typing, garbage collection, dan dukungan konkurensi tingkat tinggi yang

menggunakan komunikasi data dan penguncian secara bersamaan, selain itu Go juga memiliki library standar yang besar dan luas. Go memiliki beberapa tipe data built-in, menurut Doxey [12] Go memiliki tiga tipe data built-in antara lain *numbers, string, Booleans*.

F. C#

Nugroho [2] C# (C Sharp) adalah bahasa pemrograman baru yang diciptakan oleh Microsoft (dikembangkan di bawah kepemimpinan Anders Hejlsberg yang notabene juga telah menciptakan berbagai macam bahasa pemrograman termasuk Borland Turbo C++ dan Borland Delphi). Bahasa C# juga telah distandarisasi secara internasional oleh ECMA. C# memiliki empat tipe data built-in Nagel [13] memaparkan empat jenis tipe data tersebut antara lain integer dan floating-point numbers, character, dan Boolean.

III. METODOLOGI PENELITIAN

A. Strategi Metodologi Penelitian

1. Metode Pengumpulan Data:

a. Observasi:

Observasi merupakan teknik pengumpulan data yang mempunyai ciri yang spesifik bila dibandingkan dengan teknik yang lain yaitu wawancara dan kuesioner. Karena observasi tidak selalu dengan obyek manusia tetapi juga obyek-obyek yang lain.

Pengamatan atau observasi ini dilakukan peninjauan dan penelitian langsung di lapangan untuk memperoleh data dan mengumpulkan data yang dibutuhkan. Pengamatan dilakukan pada bulan Februari 2015 sampai dengan bulan Mei 2015 yang bertempat di PT. Navcore Nextology yang beralamat di Setiabudi Building 2 Lt.03 JL. HR Rasuna Said Kav. 62 Kunigan Jakarta Selatan. Pengamatan atau observasi yang dilakukan untuk memperoleh data pengujian yaitu menguji waktu eksekusi algoritma.

Pengujian waktu eksekusi data menggunakan algoritma bubble sort bertujuan untuk mengetahui problem waktu eksekusi menggunakan data tertentu, ketika hasil eksekusi di dapat maka data yang di proses oleh algoritma tersebut akan digunakan untuk komparasi bahasa pemrograman, data yang dibuat sebagai berikut :

Tabel 3.1 Data yang akan digunakan

Reversed
20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
Random
10, 13, 5, 3, 8, 6, 1, 19, 7, 2, 4, 9, 16, 20, 12, 18, 15, 17, 11, 14
Few Unique
20, 10, 15, 5, 20, 10, 15, 5, 20, 10, 15, 5, 20, 10, 15, 5, 20, 10, 15, 5

Nearly Sorted
10, 2, 3, 4, 15, 6, 7, 8, 9, 20, 11, 12, 13, 14, 5, 16, 17, 18, 19, 1

Masing masing jenis kondisi data mempunya panjang array 20, setelah masing masing kondisi tersebut dieksekusi maka akan diperoleh lama waktu eksekusi seperti berikut :

Tabel 3.2 Hasil pengujian data

Jenis Kondisi	Waktu Bubble Sort	Waktu Insertion sort
<i>Reversed</i>	2.0053 ms	3.0023 ms
<i>Random</i>	1.0048 ms	0
<i>Few Unique</i>	1.0043 ms	1.9974 ms
<i>Nearly Sorted</i>	0	999.9µs

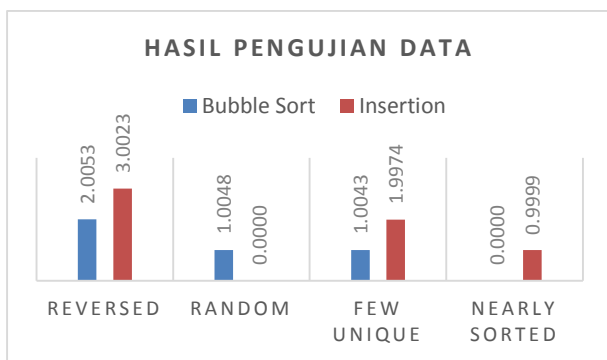


Diagram 3.1 Hasil Pengujian Data

Dari hasil pengujian kondisi data tersebut maka jenis data reversed memiliki waktu eksekusi paling lama, jenis kondisi reversed akan digunakan sebagai data acuan untuk melakukan proses komparasi bahasa pemrograman

b. Wawancara

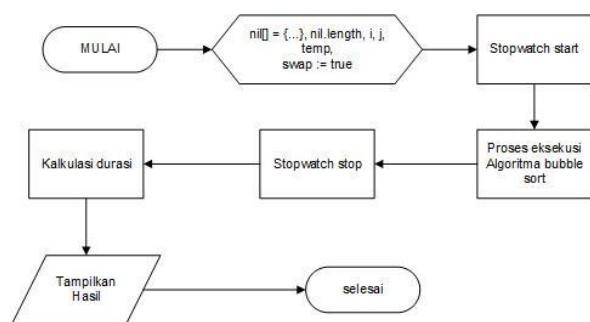
Wawancara digunakan sebagai teknik pengumpulan data apabila peneliti ingin melakukan studi pendahuluan untuk menemukan masalah yang harus diteliti dan juga apabila peneliti ingin mengetahui hal-hal dari responden yang lebih mendalam dan jumlah respondenya sedikit/kecil. Wawancara dapat dilakukan secara terstruktur maupun tidak terstruktur dan dapat dilakukan secara langsung maupun secara tidak langsung.

Wawancara ini dilakukan secara langsung saat melakukan proses kerja sama pembangunan proyek sistem informasi di PT. Navcore Nextology yang beralamat di Setiabudi Building 2 Lt.03 JL. HR Rasuna Said Kav. 62 Kunigan Jakarta Selatan. dengan divisi pengembangan framework yang di datangkan dari pihak Microsoft Indonesia untuk membahas bagaimana proses komparasi bahasa pemrograman C# dan Go harus dilakukan dan apa saja yang dibutuhkan dalam melakukan proses komparasi suatu bahasa pemrograman.

c. Studi Pustaka

Studi pustaka dilakukan dengan mempelajari teori-teori terkait yang diambil dari hasil penelitian sebelumnya agar dapat mendukung penyelesaian masalah penelitian. Pencarian informasi yang didapat terdiri dari beberapa materi dan jurnal online yang berkaitan dengan proses komparasi yang dilakukan menggunakan bahasa pemrograman C#. Jurnal sejenis yang berjudul “Microbenchmarking C++, C#, and Java ” yang disusun oleh Thomas Bruckschlegel pendiri dari ToMMTi Systems dan beberapa jurnal lainnya menjadi studi pustaka yang dijadikan acuan dalam implementasi komparasi stabilitas eksekusi kode pada bahasa pemrograman C# versi 4.0.3019 dan google golang (Go) versi 1.4.2.

B. Diagram Alur Metodologi Penelitian



Gambar 3.1 Diagram Alur Metodologi Penelitian

IV. ANALISIS DAN PERANCANGAN

A. Analisis Sistem

Analisis Sistem merupakan tahap penting yang menjadi proses awal untuk memahami sistem yang akan dibuat. Analisis sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan pada sistem yang akan dibangun. Analisis ini diperlukan sebagai dasar untuk tahap perancangan sistem dan untuk mengamati bagaimana sistem akan berjalan sesuai dengan persyaratan sistem yang mengikuti konsep dasar dari kebutuhan informasi didalamnya. Hasil akhir dari tahap ini nantinya adalah berupa sistem komputerisasi sesuai dengan kebutuhan pengguna.

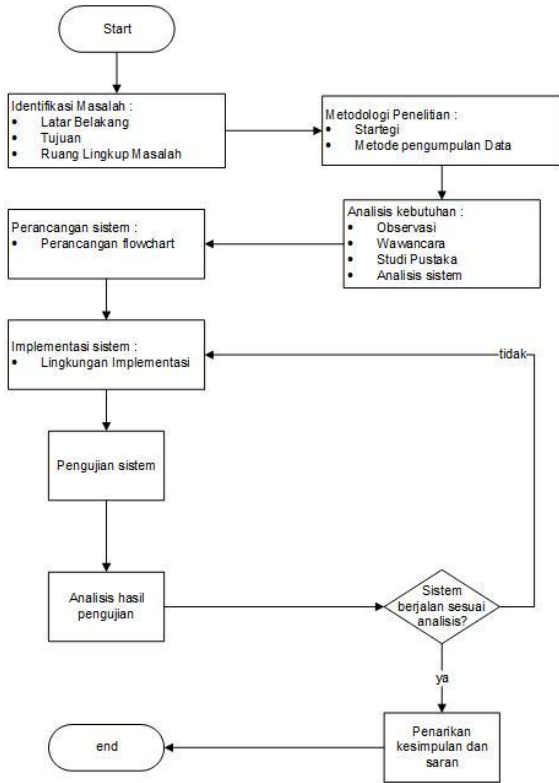
Untuk melakukan proses perhitungan kecepatan dari masing masing bahasa pemrograman, diperlukan fitur dari masing masing bahasa pemrograman untuk mengukur kecepatan dalam pengeksekusian kode.

Dalam bahasa pemrograman C# terdapat namespace system.diagnostic yang menyediakan sekumpulan class yang memungkinkan untuk berinteraksi dengan system proses, log event, dan perhitungan performa. Class yang menyediakan sekumpulan method dan properties yang dapat digunakan untuk melakukan pengukuran waktu secara akurat adalah class stopwatch.

Dalam bahasa pemrograman golang terdapat package time yang menyediakan fungsionalitas untuk mengukur dan menampilkan waktu, dan untuk perhitungan kalender selalu diasumsikan dengan kalender Gregorian. Package time dalam golang memiliki function sub yang akan

menampilkan durasi dari proses pengukuran waktu minimum dan waktu maksimum.

B. Analisis Sistem yang akan dibangun



Gambar 4.1 Flowchart sistem yang akan dibangun

Keterangan :

1. Masing masing dari bahasa pemrograman C# dan golang akan melakuka proses inialisasi awal, proses inialisasi antara lain array, panjang array, variabel iterasi di dalam proses iterasi, dan variabel Boolean ketika terjadi swap.
2. Setelah inialisasi selesai maka fitur penghitung waktu dari masing masing bahasa pemrograman akan dimulai.
3. Proses eksekusi kode algoritma bubble sort dari masing masing bahasa pemrograman akan berjalan.
4. Ketika proses selesai berjalan maka fitur stopwatch dari masing masing bahasa pemrograman akan melakukan proses kalkulasi untuk menghitung lama waktu eksekusi kode.
5. Setelah kalkulasi selesai maka hasil dari lama waktu akan ditampilkan dalam millisecond dan proses selesai

1. Perancangan Algoritma *Bubble Sort* dan *Insertion Sort* Menggunakan C# Versi 4.0.3019

Listing kode dari *algoritma bubble sort* dan *insertion sort* menggunakan bahasa pemrograman .Net C# versi 4.0.3019 beserta penjelasannya adalah sebagai berikut :

```
static void BubbleandInsertion(Int32[] arr)
{
    Stopwatch tBubble = new Stopwatch();
    tBubble.Start();
    for (int i = 0; i < arr.Count(); i++)
    {
        for (int j = 0; j < arr.Count() - 1; j++)
        {
            if (arr[j + 1] < arr[j])
            {
                var t = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = t;
                Console.WriteLine("\nProcessing Bubble sort :
                swap data from {0} to {1}\n", arr[j + 1], arr[j]);
            }
        }
        Console.WriteLine("iteration {1} \n{0}",
```

2. Perancangan Algoritma *Bubble Sort* dan *Insertion Sort* Menggunakan Go Versi 1.4.2

Listing kode dari algoritma *bubble sort* dan *insertion sort* menggunakan bahasa pemrograman Google golang versi 1.4.2 beserta penjelasannya adalah sebagai berikut :

```
func bubbleSortandinsertion(arrayzor []int32) {
    bubTime := time.Now()
    for j := 0; j < len(arrayzor); j++ {
        for i := 0; i < len(arrayzor)-1; i++ {
            if arrayzor[i+1] <
                arrayzor[i] {
                    t := arrayzor[i]
                    arrayzor[i] =
                        arrayzor[i+1]
                    arrayzor[i+1] = t
                }
            }
            fmt.Printf("\nProcessing Bubble sort : swap
            data from %v to %v", arrayzor[i+1], arrayzor[i])
        }
        fmt.Printf("iteration %v \n %v", j,
            arrayzor)
    }
    fmt.Printf("\nAfter sorting bubble
    sorting\n%v\n", arrayzor)
    sBubble = (time.Since(bubTime)).String()
}
```

3. Perancangan Data

Terdapat empat kondisi data dalam proses pengujian waktu eksekusi kode menggunakan algoritma bubble sort dan insertion sort yaitu reversed, random, few

unique, nearly sorted. Setelah dilakukan proses uji coba pada Bab III waktu eksekusi paling lama terdapat dalam jenis data reversed. Jenis data reversed akan digunakan untuk proses komparasi bahasa pemrograman, dan dari jenis data reversed akan di proses data sebagai berikut :

1. Panjang 20 data

20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

2. Panjang 40 data

40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

3. Panjang 100 data

100, 99, 98, 97, 96, 95, 94, 93, 92, 91, 90, 89, 88, 87, 86, 85, 84, 83, 82, 81, 80, 79, 78, 77, 76, 75, 74, 73, 72, 71, 70, 69, 68, 67, 66, 65, 64, 63, 62, 61, 60, 59, 58, 57, 56, 55, 54, 53, 52, 51, 50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

Masing masing data akan dilakukan pengujian berulang sebanyak 10 kali untuk masing masing bahasa pemrograman

V. IMPLEMENTASI DAN PENGUJIAN

A. Lingkungan Implementasi

Lingkungan implementasi meliputi lingkungan perangkat keras dan perangkat lunak.

1. Lingkungan Perangkat Keras

Dalam perancangan dan pembangunan komparasi stabilitas eksekusi kode bahasa pemrograman C# versi 4.0.3019 dengan Google golang versi 1.4.2 menggunakan algoritma bubble sort, lingkungan perangkat kerasnya adalah :

- a. Processor : Intel (R) Core (TM) i3- 2310M CPU @ 2.10 GHz 2.10 GHz
- b. RAM : 4.00 GB
- c. VGA : 256 MB Mobile Intel Graphic Media Accelerator 4500 MHD
- d. Harddisk : 320 GB
- e. Monitor : LCD 14,1 inchi

2. Konfigurasi Perangkat Lunak

Perangkat lunak yang digunakan dalam perancangan dan pembangunan komparasi stabilitas eksekusi kode bahasa pemrograman C# versi 4.0.3019 dengan Google golang versi 1.4.2 menggunakan algoritma bubble sort ini adalah:

- a. Sistem Operasi : Windows 8
- b. Bahasa pemrograman : C# versi 4.0.3019; Golang versi 1.4.2

B. Pengujian

Berikut adalah pengujian dari proses komparasi stabilitas eksekusi kode bahasa pemrograman C# versi 4.0.3019 dengan Google golang versi 1.4.2 menggunakan algoritma bubble sort dan insertion sort, masing masing bahasa pemrograman diberikan daftar elemen yang harus diurutkan sebagai berikut [12] :

1. Panjang 20 data

20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

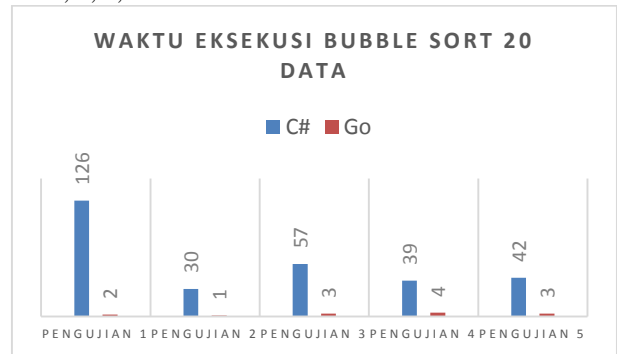


Diagram 5.1 Pengujian 20 data 1 – 5 bubble sort

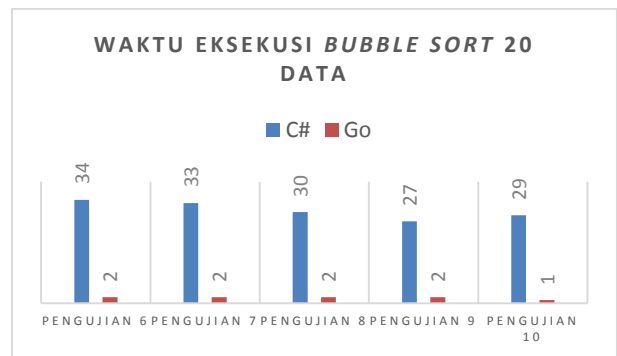


Diagram 5.2 Pengujian 20 data 6 – 10 bubble sort

Tabel 5.1 Hasil pengujian 20 data Bubble sort

Pengujian Ke	C# /ms	Go /ms
1	126,0000	1,9975
2	30,0000	1,0009
3	57,0000	3,0032
4	39,0000	4,0046
5	42,0000	3,0028
6	34,0000	1,9988
7	33,0000	2,0038
8	30,0000	2,0004
9	27,0000	2,0008
10	29,0000	1,0004

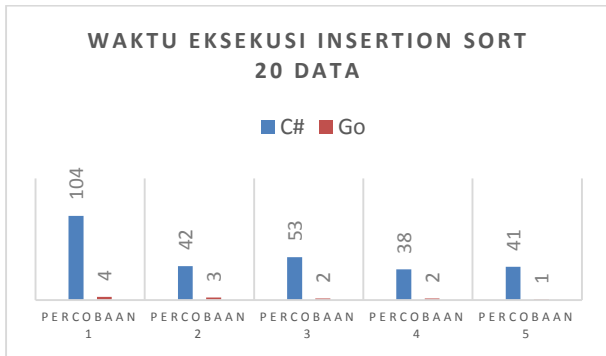


Diagram 5.3 Pengujian 20 data 1 – 5 insertion sort

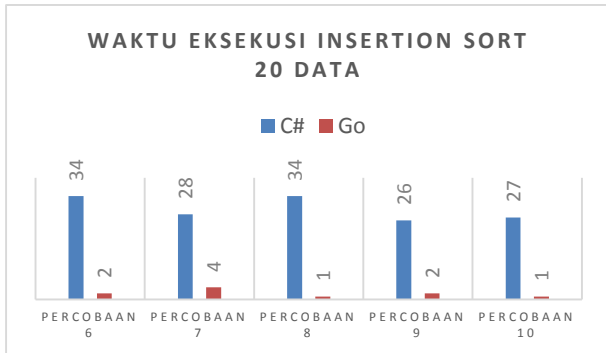


Diagram 5.4 Pengujian 20 data 6 – 10 insertion sort

Tabel 5.2 Hasil pengujian 20 data insertion sort

Pengujian Ke	C# /ms	Go /ms
1	104,0000	3,9982
2	42,0000	3,0018
3	53,0000	2,0008
4	38,0000	2,0038
5	41,0000	0,9965
6	34,0000	2,0018
7	28,0000	4,0012
8	34,0000	1,0009
9	26,0000	2,0009
10	27,0000	1,0009

C. Analisis Hasil Pengujian

Dari hasil pengujian eksekusi kode algoritma bubble sort dan insertion sort menggunakan sekumpulan elemen sebagai bahan uji coba terhadap lama waktu eksekusi kode C# versi 4.0.3019 dengan golang versi 1.4.2 di dapat hasil berikut ini :

Tabel 5.7 Rata rata waktu komparasi menggunakan bubble sort

Bahasa Pemrograman	20 Data	40 Data	100 Data
C#	44,7 ms	137,1 ms	599,4 ms
Go	2,20132 ms	21,83345ms	31,42093 ms

Tabel 5.8 Rata rata waktu komparasi menggunakan insertion sort

Bahasa Pemrograman	20 Data	40 Data	100 Data
C#	42,7 ms	244,1818182 ms	599,4 ms
Go	2,20068 ms	5,10318 ms	31,42093 ms

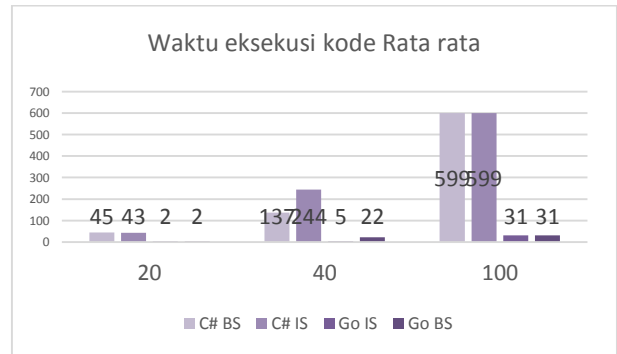


Diagram 5.13 Waktu eksekusi kode C# dan Go

Dari pengujian yang telah dilakukan menggunakan algoritma bubble sort dan insertion sort dapat diketahui waktu eksekusi kode yang tercepat adalah Go, seperti yang di tunjukkan oleh tabel 5.8 dan diagram 5.13 dengan menggunakan jenis kondisi data yang di urutkan adalah reversed, berikut penjelasan dari diagram 5.13 :

- Menunjukkan waktu eksekusi rata rata antara bahasa pemrograman C# dengan Go.
- Algoritma yang digunakan oleh masing masing bahasa pemrograman adalah algoritma bubble sort dan insertion sort dengan menggunakan jenis kondisi data reversed dengan panjang elemen 20, 40, dan 100 elemen.
- Pada bahasa pemrograman C#, lama waktu eksekusi kode menggunakan bubble sort memiliki lama waktu eksekusi kode rata rata 45 ms untuk panjang elemen 20 elemen, 137 ms untuk panjang elemen 40 elemen, dan 599 ms untuk panjang elemen 100 elemen.
- Pada bahasa pemrograman C#, lama waktu eksekusi kode menggunakan insertion sort memiliki lama waktu eksekusi kode rata rata 43 ms untuk panjang elemen 20 elemen, 244 ms untuk panjang elemen 40 elemen, dan 599 ms untuk panjang elemen 100 elemen.
- Pada bahasa pemrograman Go, lama waktu eksekusi kode menggunakan bubble sort memiliki lama waktu eksekusi kode rata rata 2 ms untuk panjang elemen 20 elemen, 5 ms untuk panjang elemen 40

elemen, dan 31 ms untuk panjang elemen 100 elemen.

- f. Pada bahasa pemrograman Go, lama waktu eksekusi kode menggunakan insertion sort memiliki lama waktu eksekusi kode rata rata 2 ms untuk panjang elemen 20 elemen, 22 ms untuk panjang elemen 40 elemen, dan 31 ms untuk panjang elemen 100 elemen.

VI. KESIMPULAN DAN SARAN

A. Kesimpulan

Hasil dari perbandingan bahasa pemrograman pada komparasi stabilitas kode dalam lingkungan pengembangan .Net C# versi 4.0.3019 dengan google golang versi 1.4.2 menggunakan algoritma bubble sort dan insertion sort adalah sebagai berikut:

1. Total waktu paling lama untuk proses eksekusi algoritma bubble sort dan insertion sort adalah menggunakan kondisi elemen reversed.
2. Semakin banyak jumlah elemen yang digunakan untuk melakukan pengujian lama waktu eksekusi kode algoritma bubble sort dan insertion sort maka semakin lama waktu eksekusi pada masing masing algoritma.
3. Semakin banyak jumlah elemen yang digunakan untuk melakukan pengujian lama waktu eksekusi kode pada bahasa pemrograman C# dan Go bukan faktor yang terlalu mempengaruhi lama waktu eksekusi kode dari masing masing bahasa pemrograman.
4. Proses eksekusi kode bahasa pemrograman golang versi 1.4.2 relatif lebih cepat dari pada bahasa pemrograman .net C# versi 4.0.3019.
5. Kecepatan eksekusi kode menggunakan bahasa pemrograman C# 4.0.3019 dapat diketahui dengan menggunakan class stopwatch yang tersedia pada namespace system.diagnostics dan untuk pemrograman golang versi 1.4.2 kecepatan eksekusi kode dapat diketahui menggunakan package time.
6. Hasil dari komparasi suatu bahasa pemrograman tidak digunakan untuk menilai baik atau buruknya kualitas bahasa pemrograman. Tujuan komparasi bahasa pemrograman digunakan untuk meminimalisir kegagalan dalam sistem dan menyempurnakan suatu perangkat lunak sistem.

B. Saran

Saran yang dapat diberikan berdasarkan hasil yang telah dicapai maupun untuk pengembangan proses pengujian pada masa yang akan datang, antara lain :

1. Terdapat banyak sekali proses pengujian untuk membandingkan kinerja dan performa suatu bahasa pemrograman salah satunya adalah prosedur pengujian waktu eksekusi kode.
2. Diharapkan melakukan pengembangan terhadap proses pengujian lain yang dapat dilakukan untuk mengetahui apakah suatu bahasa pemrograman tertentu dapat menunjang kemampuan suatu sistem.
3. Proses pengembangan dari komparasi dapat dilakukan dengan menguji pengaruh sistem operasi terhadap suatu bahasa pemrograman dan faktor pendukung lain seperti kinerja garbage collector pada suatu bahasa pemrograman.

REFERENSI

- [1]. Darmawan, E. 2014. Pemrograman Dasar C-JAVA-C# Yang Susah Jadi Mudah. Bandung: Informatika.
- [2]. Nugroho, A., Chandra, D. W., Prestiliano, J., Beeh, Y.R. 2013. Cara Mudah Dan Cepat Belajar Pemrograman C#.Net. Yogyakarta: ANDI.
- [3]. Summerfield, M. 2012. Programming In Go Creating Applications For The 21st Century. United States of America : Addison-Wesley.
- [4]. Zarlis, M. 2008. Algoritma Dan Pemrograman: Tori Dan Praktik Dalam Pascal. Medan: USU Press.
- [5]. Sedgewick, R., Wayne, K. 2014. Algorithms: Fourth Edition Part I. United States of America : Addison-Wesley.
- [6]. Wajhillah, R., Wibowo, A., Hermalini, E, H. 2011. Visualisasi Logika Algoritma Pengurutan Data Menggunakan Java. Makalah disajikan dalam Seminar Nasional Inovasi Dan Teknologi, LPPM BSI, Jatiwaringin, 18 Maret 2011.
- [7]. Orwant, J., Hietaniemi, J., Macdonald, J. 1999. Mastering Algorithms with Perl. United States of America : O'Reilly.
- [8]. Suprpto. 2008. Bahasa Pemrograman Untuk Sekolah Menengah Kejuruan. Jakarta : Direktorat pembinaan sekolah menengah kejuruan.
- [9]. Lungan, R. 2006. "Aplikasi Statistika dan Hitung Peluang". Yogyakarta : Graha Ilmu.
- [10]. Hasan, I. 2009. "Pokok-Pokok Materi Statistik 1 (Statistik Deskriptif)". Jakarta: PT.Bumi Aksara.
- [11]. Sugiyono. 2012. "Metode Penelitian Kuantitatif, Kualitatif, dan R&D". Bandung: Alfabeta.
- [12]. Doxsey, C. 2012. "An Introduction to Programming in Go". Retrieved 04 april 2015, from <https://www.golang-book.com>
- [13]. Nagel, C., Evjen, B., Glynn, K., Watson, K., Skinner, M. "Professional C# 4 and .NET 4". USA: Wrox