

1. Algoritma Kategori Konstan $O(1)$

Pada percobaan algoritma kategori konstan, kami menggunakan input sebanyak 3 buah variasi yang bernilai 10, 1000, dan 100000. *Code source* yang kami gunakan adalah sebagai berikut.

```
#include <iostream>
#include <chrono>
using namespace std;

int main()
{
    for(int i=0; i<5; i++)
    {
        auto start = chrono::steady_clock::now();

        int a=10;

        cout << a << "\n";

        auto end = chrono::steady_clock::now();
        auto diff = end - start;
        cout << chrono::duration<double, milli>(diff).count() << " ms " << endl
        <<endl;
    }

    return 0;
}
```

Penggunaan chrono adalah untuk mencatat waktu *running time* pada program. Digunakan fungsi loop dengan maksud untuk menghemat waktu dalam menjalankan program sebanyak 5 kali tersebut. Fungsi chrono dimasukkan ke dalam fungsi loop agar waktu yang tercatat bukanlah algoritma kategori linier. Hasil percobaan adalah sebagai berikut.

No.	10	1000	100000
1	0.396 ms	0.618 ms	0.373 ms
2	0.302 ms	0.451 ms	0.291 ms
3	0.272 ms	0.356 ms	0.347 ms
4	0.273 ms	0.416 ms	0.27 ms
5	0.291 ms	0.304 ms	0.331 ms

Dilihat dari data hasil percobaan, terlihat bahwa pada input 10, waktu yang dituliskan mendapatkan nilai yang konstan pada 0,27 ms. Angka pada running pertama seperti 0,396 ms merupakan initial run saja dan membutuhkan waktu yang lebih lama. Setelah initial run waktu running program berjalan normal dan konstan. Untuk input 1000, sama seperti input 10, initial run memerlukan waktu yang lebih lama. Setelahnya, waktu menunjukkan angka stabil di 0,4 ms. Untuk input 100000, angka berjalan stabil di 0,3 ms. Dapat disimpulkan, untuk algoritme kategori konstan, tidak terdapat perbedaan waktu yang signifikan antar input yang bervariasi sehingga waktu tergolong independent dengan nilai input.

2. Algoritma Kategori Linier $O(n)$

Pada percobaan algoritma kategori linier, kami menggunakan iterasi sebanyak 3 buah variasi yang bernilai 1, 3, dan 5. Code source yang kami gunakan adalah sebagai berikut.

```
#include <iostream>
#include <chrono>
using namespace std;

int main()
{
    for(int i=0; i<5; i++)
    {
        auto start = chrono::steady_clock::now();

        int a=10;

        for(int j=0; j<1; j++)
        {
            cout << a << "\n";
        }
        auto end = chrono::steady_clock::now();
        auto diff = end - start;
        cout << chrono::duration<double, milli>(diff).count() << " ms " << endl
<<endl;
```

```

    }

    return 0;
}

```

Penggunaan chrono adalah untuk mencatat waktu *running time* pada program. Digunakan fungsi loop dengan maksud untuk menghemat waktu dalam menjalankan program sebanyak 5 kali tersebut. Fungsi chrono dimasukkan ke dalam fungsi loop agar waktu yang tercatat bukanlah algoritma kategori quadratic. Hasil percobaan adalah sebagai berikut.

No.	1	3	5
1	0.381 ms	0.667 ms	1.016 ms
2	0.261 ms	0.43 ms	0.914 ms
3	0.182 ms	0.707 ms	1.514 ms
4	0.183 ms	0.933 ms	1.497 ms
5	0.315 ms	1.034 ms	1.78 ms

Dari hasil data percobaan algoritma kategori linier diatas, input 1 menghasilkan rata-rata waktu sebanyak 0,264 ms. Untuk input 3, menghasilkan rata-rata *running time* sebanyak 0,754 ms. Untuk input 5, menghasilkan rata-rata *running time* sebanyak 1.344 ms. Rumus umum *running time* algoritme linier adalah $n+1$ dengan notasi Big O adalah n . Jika $n=1$ menghasilkan *running time* sebanyak 0,264 ms, maka untuk $n=3$ akan menghasilkan *running time* sebanyak $3 \times 0,264$ ms yang sama dengan 0,793 ms. Nilai tersebut tidak begitu jauh dengan nilai hasil eksperimen yakni, 0,754 ms. Untuk $n=5$, waktu *running time* adalah $5 \times 0,264$ ms yang sama dengan 1,322 ms. Nilai tersebut juga sangat dekat dengan nilai hasil eksperimen yakni 1,344 ms. Hal ini membuktikan bahwa algoritma diatas berbanding lurus (linier) dengan banyaknya nilai input yang dimasukkan.

3. Algoritma Kategori Konstan $O(n^2)$

Pada percobaan algoritma kategori quadratic, kami menggunakan input sebanyak 3 buah variasi yang bernilai 1, 3, dan 5. Untuk bagian algoritma liniernya, kami menggunakan input hanya $n=1$ agar mempermudah perhitungan. *Code source* yang kami gunakan adalah sebagai berikut.

```
#include <iostream>
#include <chrono>
using namespace std;

int main()
{
    for(int i=0; i<5; i++)
    {
        auto start = chrono::steady_clock::now();

        int a=10;

        for(int j=0; j<3; j++)
        {
            for(int k=0; k<5; k++)
            {
                cout << a << "\n";
            }
        }
        auto end = chrono::steady_clock::now();
        auto diff = end - start;
        cout << chrono::duration<double, milli>(diff).count() << " ms " << endl
<<endl;
    }

    return 0;
}
```

Penggunaan chrono adalah untuk mencatat waktu *running time* pada program. Digunakan fungsi loop dengan maksud untuk menghemat waktu dalam menjalankan program sebanyak 5 kali tersebut. Fungsi chrono dimasukkan ke dalam fungsi loop agar waktu yang tercatat bukanlah algoritma kategori cubic. Hasil percobaan adalah sebagai berikut.

No.	1	3	5
1	0.357 ms	1.13 ms	1.538 ms
2	0.18 ms	0.823 ms	1.475 ms
3	0.155 ms	0.825 ms	1.479 ms
4	0.142 ms	0.996 ms	1.566 ms
5	0.964 ms	1.038 ms	2.031 ms

Dari hasil data percobaan algoritma kategori quadratic diatas, input 1 menghasilkan rata-rata waktu sebanyak 0,3596 ms. Untuk input 3, menghasilkan rata-rata *running time* sebanyak 0,9624 ms. Untuk input 5, menghasilkan rata-rata *running time* sebanyak 1,6178 ms. Rumus umum *running time* algoritme quadratic adalah $n(n+1)*x$ dengan x sebagai konstanta pengoperasian bernilai 0,075 ms (rumus ini didapatkan dari menghitung persamaan algoritma quadratic dengan menggunakan data percobaan eksperimen algoritma kategori linier yakni saat $n=1$, maka $0,264 \text{ ms} + 1(1)x = 0,359$). Jika didekatkan dengan notasi Big O maka rumusnya adalah n^2*x .

Jika $n=1$, maka $1^2 * 0,075 + 0,264 \text{ ms}$ menghasilkan *running time* sebanyak 0,339 ms, maka untuk $n=3$ akan menghasilkan *running time* sebanyak $3^2 * 0,075 + 0,264 \text{ ms}$ yang sama dengan 0,944 ms. Nilai tersebut tidak begitu jauh dengan nilai hasil eksperimen yakni, 0,9624 ms. Untuk $n=5$, waktu *running time* adalah $5^2 * 0,075 + 0,264 \text{ ms}$ yang sama dengan 2,153 ms. Nilai tersebut hampir mendekati dengan nilai hasil eksperimen yakni 1,6178 ms. Hal ini dapat terjadi dikarenakan bervariasinya nilai x sehingga x yang diambil adalah nilai rata-rata dari semua x yang memungkinkan. Hal ini membuktikan bahwa algoritma diatas berbanding kuadrat dengan banyaknya nilai input yang dimasukkan.