**Jurnal Sains dan Teknologi Widya**
Volume 1, Nomor 2, Juli 2022: halaman 188-195
https://jurnal.amikwidyaloka.ac.id/index.php/jstekwid
jurnal@amikwidyaloka.ac.id / editor.jstekwid@gmail.com

P-ISSN: 2810-093x
E-ISSN: 2810-0166

# Analisis Perbandingan Fibonacci dengan Iterasi dan Rekursi Terhadap Efektifitas Waktu

**Fenina Adline Twince Tobing[1], Prayogo[2], Alex Chandra[3]**

[1]Program Studi Informatika, Fakultas Teknik dan Informatika Universitas Multimedia Nusantara, Tangerang, Banten, Indonesia
[2]Program Studi Teknik Informatika, AMIK Mapan, Tangerang, Banten, Indonesia
[3]Program Studi Manajemen Informatika, AMIK Widyaloka Medan Sumatera Utara, Indonesia

Email : [1]fenina.tobing@umn.ac.id [2]alexsiburian03@gmail.com

## Abstrak

Adanya kesalahan dalam pemilihan metode dapat mengakibatkan tidak cepatnya suatu proses untuk menyelesaikan suatu masalah. Penyelesaian suatu masalah dalam fibonacci dapat diselesaikan dengan 2 metode, yaitu: dengan menggunakan iterasi maupun rekursif. Analisis perbandingan kecepatan antara iterasi dan rekursif akan dilakukan pada penelitian ini dengan menguji kedua metode tersebut dari nilai yang kecil ke nilai yang besar. Hasil yang diperoleh adalah penggunaan rekursif lebih baik untuk dilakukan pada penghitungan fibonacci yang berukuran kecil. Sedangkan penggunaan iterasi cenderung stabil, tidak banyak perbedaan pada penghitungan fibonacci yang bernilai kecil maupun besar.

***Keywords—component: iterasi; looping, pengulangan diri, rekursif.***

## *Abstract*

*An error in the selection of methods can result in a process that is not fast enough to solve a problem . Solving a problem in Fibonacci can be solved by 2 methods, namely: by using iteration or recursive. Analysis of the speed comparison between iteration and recursive will be carried out in this study by testing both methods from small values to large values. The results obtained are that the use of recursive is better for small Fibonacci calculations. While the use of iterations tends to be stable, there is not much difference in the Fibonacci calculations of small or large values.*

***Keywords—components : iteration ; loops, self repetition, recursive .***

## 1 Pendahuluan

Iteratif dan rekursif adalah sebuah algoritma yang memiliki ciri yang sama yaitu mengulangi langkah-langkah yang diinginkan. Namun, rekursif agak berbeda dari iterasi. Rekursif adalah sebuah algoritma yang berisi pemanggilan dirinya sendiri sehingga menghasilkan looping. Harus ada batasan di dalam rekursif sehingga looping dapat dibatalkan, jika tidak maka akan terjadi infinity loop dan dapat menyebabkan memori penuh.[1]

Jurnal Sains dan Teknologi Widya
Volume 1, Nomor 2, Juli 2022: halaman 188-195
https://jurnal.amikwidyaloka.ac.id/index.php/jstekwid
jurnal@amikwidyaloka.ac.id / editor.jstekwid@gmail.com
P-ISSN: 2810-093x
E-ISSN: 2810-0166

Sedangkan Iterasi adalah sebuah algoritma yang mengulang sekelompok instruksi tertentu dengan sebuah syarat yang dimana jika syarat tersebut tidak dipenuhi maka looping akan berhenti. Disini kita akan menggunakan rumus fibonacci sebagai acuan rumus utama lalu membuat versi iteratif dan rekursifnya dan membandingkan efektifas waktunya oleh karena itu, penulis mengambil penelitian dengan judul "Analisis Perbandingan Fibonacci Iterasi Biasa dan Rekursif terhadap Efektifitas Waktu".

Pada penilitian ini penulis mengambil rumusan masalah dari penilitian ini adalah :

A. Bagaimana penerapan algoritma rekursif dan iteratif terhadap penyelesaian masalah fibonacci?
B. Bagaiamana membuat sebuah aplikasi penyelesaian masalah fibonacci dengan menerapkan algoritma rekursif dan iteratif didalam sebuah model penyelesaian.

Setelah menentukan rumusan masalah penulis juga menentukan batasan masalah pada penelitian ini adalah sebagai berikut :

A. Keadaan akhir (*goal state*) berupa nilai fibonacci yang diinginkan.
B. Input berupa angka yang akan menentukan hasil dari goalnya

## 2 Tinjauan Literatur

### *A.* Fibonacci

Fibonacci merupakan barisan yang dihasilkan dari penjumlahan dari 2 bilangan sebelumnya. Barisan berawal dari nilai 0 dan 1 dan setelahnya didapatkan hasil seperti berikut [2][3][4]

| $F_0$ | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $F_6$ | $F_7$ | $F_8$ | $F_9$ | $F_{10}$ | $F_{11}$ | $F_{12}$ | $F_{13}$ | $F_{14}$ | $F_{15}$ | $F_{16}$ | $F_{17}$ | $F_{18}$ | $F_{19}$ | $F_{20}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 | 89 | 144 | 233 | 377 | 610 | 987 | 1597 | 2584 | 4181 | 6765 |

Gambar 1. Deret bilangan fibonacci

### *B.* Rekursi

Rekursi adalah suatu proses dengan salah satu langkah dalam prosedur tersebut menjalankan prosedur itu sendiri. Dalam pemrograman komputer, rekursi dicontohkan saat sebuah fungsi didefinisikan dalam bentuk sederhana, bahkan versi terkecil dari dirinya. Solusi dari permasalahan kemudian dirancang dengan menggabungkan solusi-solusi yang didapat dari versi sederhana dari permasalahan.[5]

### *C.* Iterasi

Iterasi adalah sebuah algoritma yang mengulang sekelompok instruksi tertentu dengan sebuah syarat yang dimana jika syarat tersebut tidak dipenuhi maka looping akan berhenti. Instruksi-instruksi yang dapat digunakan ada seperti : *do while*, *while*, dan *for*. [6][7]

### *D.* Implementasi fungsi rekursi dan iterasi untuk fibonacci

Bilangan Fibonacci jika didefinisikan secara rekursi sebagai berikut:

**Jurnal Sains dan Teknologi Widya**
Volume 1, Nomor 2, Juli 2022: halaman 188-195
https://jurnal.amikwidyaloka.ac.id/index.php/jstekwid
jurnal@amikwidyaloka.ac.id / editor.jstekwid@gmail.com
P-ISSN: 2810-093x
E-ISSN: 2810-0166

$$F(n) = \begin{cases} 0, & \text{jika } n = 0; \\ 1, & \text{jika } n = 1; \\ F(n-1) + F(n-2) & \text{jika tidak.} \end{cases}$$

Gambar 2. Rumus rekursi fibonacci

Dari rumus tersebut dapat kita implemetasikan ke dalam bahasa c dengan algoritma sebagai berikut

```
//deklrasi function
int Fibbonaci (int n){
    //algoritma
    if (n = 0)
        return 0;
    else {
        if (n = 1)
            return 1;
        else
            return Fibbonaci(n-1) + Fibbonaci(n-2);

    }
}
```

Gambar 3. Algoritma fibonacci menggunakan rekursi ver 1

Atau dapat disederhakan menjadi seperti berikut

```
//deklrasi function
int Fibbonaci (int n){
    //algoritma
    if (n <= 1)
        return n;
    else
        return Fibbonaci(n-1) + Fibbonaci(n-2);
}
```

Gambar 4. Algoritma fibonacci menggunakan rekursi ver 2

Sedangkan implementasi menggunakan fungsi iterasi sebagai berikut

```
int main(){
    //deklarasi variabel
    int n, first = 0, second = 1, fib;

    //algortima
    for (int ctr = 0; ctr<=n; ctr++){
        if (ctr <=1)
            fib = ctr;
        else{
            fib = first + second;
            first = second;
            second = fib;
        }
    }
}
```

Gambar 5. Algoritma fibonacci menggunakan iterasi

## 3 Hasil dan Pembahasan

*A. Menggunakan Algoritma Rekursi*

**Jurnal Sains dan Teknologi Widya**
Volume 1, Nomor 2, Juli 2022: halaman 188-195
https://jurnal.amikwidyaloka.ac.id/index.php/jstekwid
jurnal@amikwidyaloka.ac.id / editor.jstekwid@gmail.com
P-ISSN: 2810-093x
E-ISSN: 2810-0166

Percobaan menggunakan algoritma rekursi dilakukan dengan menetapkan bilangan fibbonacci ke berapa yang diinginkan dimulai dari yang terkecil hingga ke terbesar. Kemudian waktu eksekusi yang ditampilkan di *console* dicatat di tabel di bawah ini.

Tabel 1. Eksekusi fibonacci menggunakan rekursi

| fibbonacci(n) | detik |
|---|---|
| 2 | 0,126 |
| 20 | 0,119 |
| 40 | 6,275 |
| 45 | 11,77 |

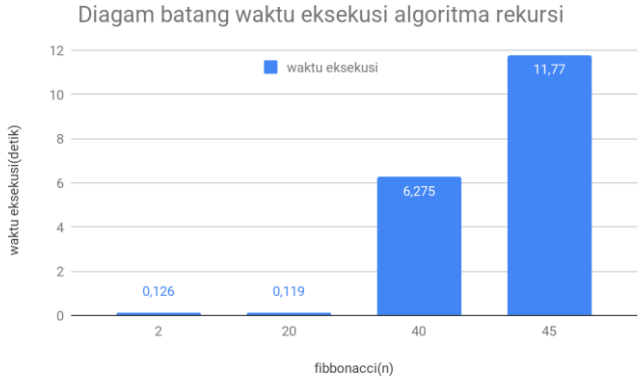
*B. Menggunakan Algoritma Iterasi*

Percobaan menggunakan algoritma iterasi dilakukan dengan menetapkan bilangan fibbonacci ke berapa yang diinginkan dimulai dari yang terkecil hingga ke terbesar. Kemudian waktu eksekusi yang ditampilkan di *console* dicatat di tabel di bawah ini.

Tabel 2. Eksekusi fibonacci menggunakan iterasi

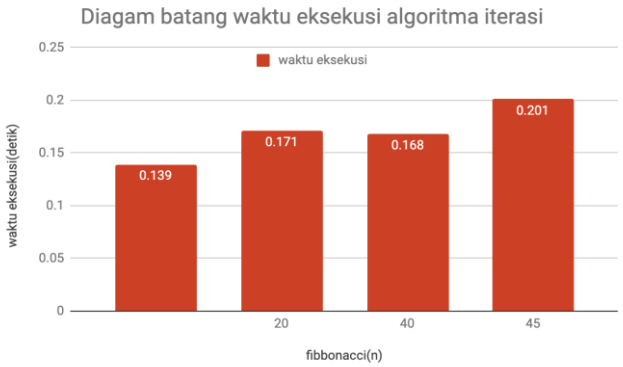
| fibbonacci(n) | Iterasi |
|---|---|
| 2 | 0,139 |
| 20 | 0,171 |
| 40 | 0,168 |
| 45 | 0,201 |

Berikut adalah perbandingan untuk waktu eksekusi algoritma rekursi dan iterasi menggunakan diagram batang.

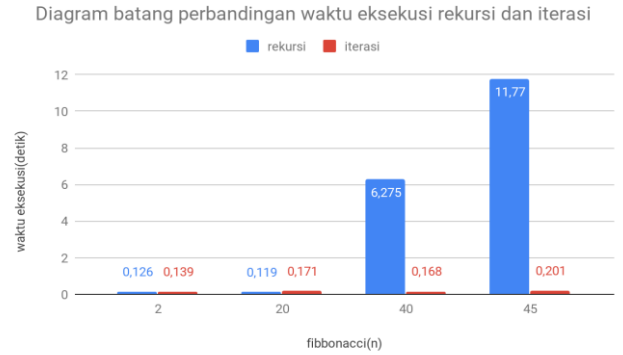


Jurnal Sains dan Teknologi Widya
Volume 1, Nomor 2, Juli 2022: halaman 188-195
https://jurnal.amikwidyaloka.ac.id/index.php/jstekwid
jurnal@amikwidyaloka.ac.id / editor.jstekwid@gmail.com
P-ISSN: 2810-093x
E-ISSN: 2810-0166

Gambar 6. Diagram batang waktu eksekusi algoritma rekursi




Gambar 7. Diagram batang waktu eksekusi algoritma iterasi




Gambar 8. Diagram perbandingan batang waktu eksekusi algoritma rekursi dan iterasi

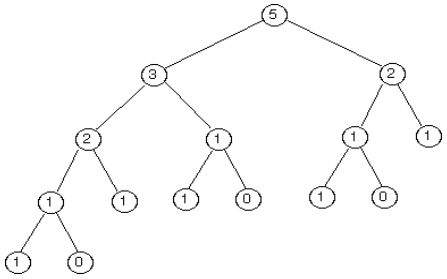
Dengan melihat implementasi algoritma antara menggunakan konsep rekursi dan konsep iterasi, dapat dilihat implementasi dengan menggunakan konsep rekursi lebih mudah dipahami dan dibuat. Hal ini dikarenakan konsep rekursi sendiri yaitu sebuah fungsi yang memanggil dirinya sendiri sehingga *line code* juga akan semakin sedikit.

Pada konsep iterasi, penulis menggunakan variabel tambahan sebagai tempat penyimpanan nilai sementara. Dalam kasus ini, kinerja komputer akan sedikit lebih berat karena harus menyiapkan memori untuk variabel tambahan tersebut.

Bukan berarti algoritma rekursi lebih baik daripada algoritma iterasi. Justru algoritma rekursi bisa menjadi penyebab stack overflow yang diakibatkan dari proses pemanggilan berulang dirinya sendiri.

Contoh, kita mengambil proses untuk mencari bilangan fibonacci ke 5 menggunakan konsep rekursi. Maka bagan pemanggilannya akan seperti berikut

Jurnal Sains dan Teknologi Widya
Volume 1, Nomor 2, Juli 2022: halaman 188-195
https://jurnal.amikwidyaloka.ac.id/index.php/jstekwid
jurnal@amikwidyaloka.ac.id / editor.jstekwid@gmail.com
P-ISSN: 2810-093x
E-ISSN: 2810-0166

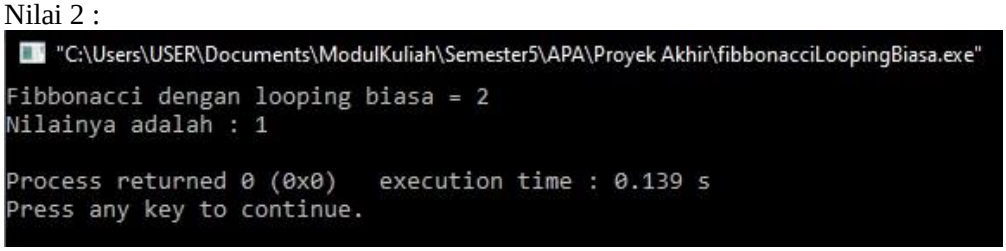



Gambar 9. Pohon rekursi fibonacci(5)

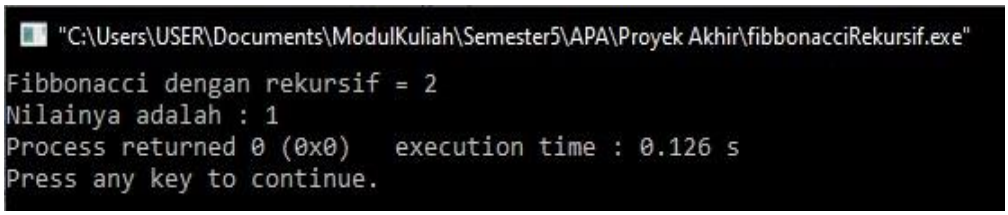
Untuk mencari fibonacci ke 5 saja, komputer perlu melakukan 15 kali pemanggilan terhadap fungsi rekursi fibonnacci. Setiap pemanggilan akan melakukan proses dan mengembalikan nilai untuk kemudian akan diproses oleh fungsi dibawah *tree*-nya sampai ke akarnya. Akibatnya memori komputer akan termakan banyak karena harus melakukan pemanggilan rekursi fibonacci yang banyak.[8][9]

Untuk konsep iterasi hanya melakukan konsep pengulangan yang mengganti nilai variabel yang telah ditentukan sehingga lebih menghemat memori dan kinerja cpunya.[10]

Nilai 2 :

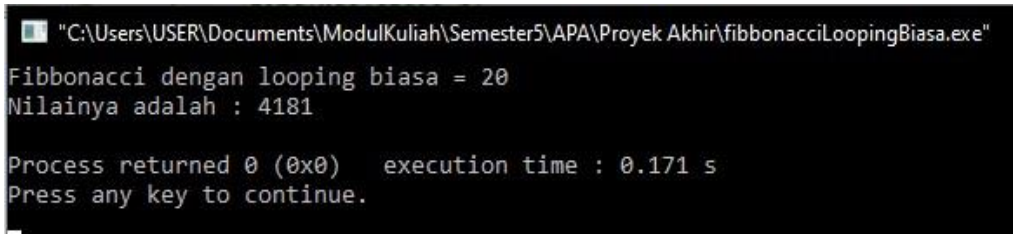



Gambar 10. Hasil Execution Time Iterasi nilai 2



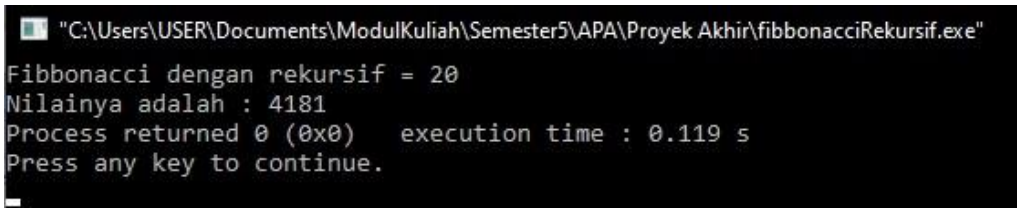

Gambar 11. Hasil Execution Time Rekursif nilai 2

Nilai 20



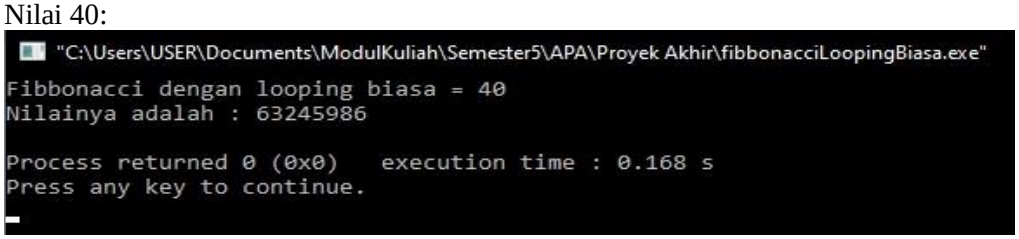

Gambar 12. Hasil Execution Time Iterasi nilai 20

**Jurnal Sains dan Teknologi Widya**
Volume 1, Nomor 2, Juli 2022: halaman 188-195
https://jurnal.amikwidyaloka.ac.id/index.php/jstekwid
jurnal@amikwidyaloka.ac.id / editor.jstekwid@gmail.com
P-ISSN: 2810-093x
E-ISSN: 2810-0166

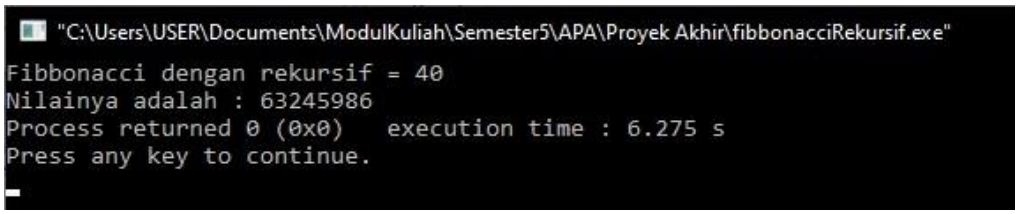


Gambar 13. Hasil Execution Time Rekursif nilai 20

Nilai 40:

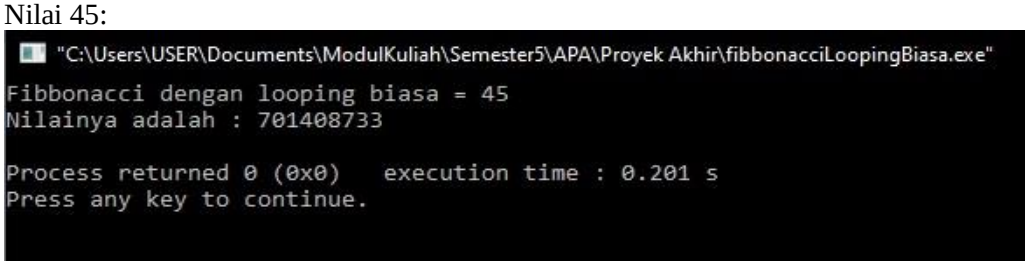


Gambar 14. Hasil Execution Time  Iterasi nilai 40

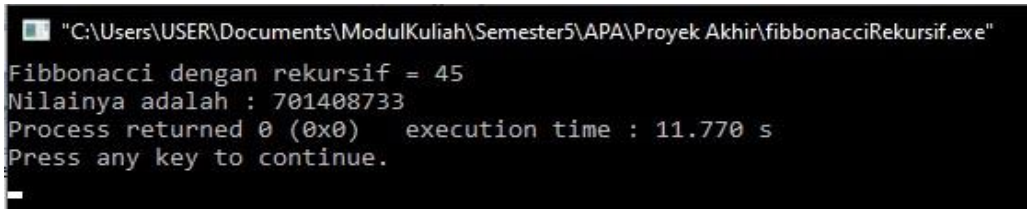


Gambar 15. Hasil Execution Time Rekursif nilai 40

Nilai 45:



Gambar 16. Hasil Execution Time Rekursif  nilai 45



Gambar 17. Hasil Execution Time Iterasi nilai 45

**Jurnal Sains dan Teknologi Widya**

Volume 1, Nomor 2, Juli 2022: halaman 188-195

https://jurnal.amikwidyaloka.ac.id/index.php/jstekwid

jurnal@amikwidyaloka.ac.id / editor.jstekwid@gmail.com

P-ISSN: 2810-093x

E-ISSN: 2810-0166

## 4 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penggunaan rekursif lebih baik untuk dilakukan pada penghitungan fibonacci yang berukuran kecil. Sedangkan penggunaan iterasi cenderung stabil, tidak banyak perbedaan pada penghitungan fibonacci yang bernilai kecil maupun besar.

Pada penghitungan fibonacci skala kecil, yang baik digunakan sesuai dengan efektifitas waktu adalah dengan menggunakan rekursif. Sedangkan untuk penghitungan fibonacci yang skala besar, sebaiknya menggunakan iterasi.

## Referensi

[1] M. Ahmad. (2016, 30 Desember). Memahami Cara Kerja Fungsi Rekursif. Diakses pada 14 November 2019, dari https://www.petanikode.com/fungsi-rekursif/.

[2] A. Ahmad.. (2019, 29 September). Bilangan Fibonacci - Sejarah, Pengertian, Pola, Penerapan, dan Gambar. Diakses pada 13 November 2019, dari https://rumusbilangan.com/bilangan-fibonacci/

[3] M. Gary. (2012, 15 Mei). What is The Fibonacci Sequence. Diakses pada 14 November 2019, dari https://www.goldennumber.net/fibonacci-series/.

[4] https://en.m.wikipedia.org/wiki/Fibonacci_number. Diakses pada 4 November 2019.

[5] https://id.m.wikipedia.org/wiki/Rekursi. Diakses pada 4 November 2019

[6] R.K. Gregorius, Kompleksitas dari Algoritma-Algoritma untuk Menghitung Bilangan Fibonacci, Jurusan Teknik Informatika, Institut Teknologi Bandung, Bandung, 2009.

[7] T. Helen, W. Aaron, B. Heather, and F. Irene. "Teacher Professional Learning and Development: Best Evidence Synthesis Iteration". Diakses pada 4 November 2019, dari http://www.oecd.org/edu/school/48727127.pdf

[8] KMKLabs. (2016, 21 Februari). Rekursif vs Iterasi. Diakses pada 4 November 2019, dari https://blog.kmkonline.co.id/rekursif-vs-iterasi-57378dffb183

[9] *https://inst.eecs.berkeley.edu/~cs61bl/r//cur/trees/fibonacci-tree.html?topic=lab15.topic&step=7&course=* Diakses pada 18 November 2019

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

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