



INNOVATIVE: Journal Of Social Science Research

Volume 4 Nomor 4 Tahun 2024 Page 5622-5630

E-ISSN 2807-4238 and P-ISSN 2807-4246

Website: <https://j-innovative.org/index.php/Innovative>

Rekursif Dalam Pemrograman Teori Dan Praktek

Elmi Rahmawati^{1✉}, Pinta Medina², Devi Syukri Azhari³

Sistem Informasi, Fakultas Ilmu Komputer UPI YPTK Padang

Email: elmi_rahmawati@upiptk.ac.id^{1✉}

Abstrak

Rekursif, bagaikan pisau bedah analitis, memungkinkan kita membedah masalah rumit menjadi sub-masalah yang lebih mudah dipahami dan dipecahkan. Teknik ini, bagaikan boneka Matryoshka, berlapis dan berulang, namun menghasilkan solusi yang elegan. Digemari ilmuwan komputer, rekursif menjelma menjadi kode yang ringkas dan intuitif. Algoritma rekursif bagaikan petunjuk detail, memandu komputer menyelesaikan tugas secara berulang hingga mencapai inti permasalahan. Namun, bagaikan pisau bermata dua, rekursif juga memiliki kekurangan. Penggunaan yang berlebihan dapat membebani memori dan waktu proses, bagaikan beban yang menghambat kelancaran. Kesalahan dalam kode rekursif pun bagaikan benang kusut yang rumit untuk diurai. Meskipun demikian, rekursif tetaplah alat yang berharga. Kemampuannya untuk menjelaskan struktur bahasa, memodelkan pertumbuhan, dan menganalisis sistem kompleks menjadikannya tak tergantikan. Rekursif, bagaikan seni memecahkan masalah, mengantarkan kita pada solusi yang cerdas dan efisien. Di tangan yang tepat, rekursif menjadi kunci untuk membuka gerbang pengetahuan dan inovasi.

Kata Kunci: *Boneka Matryoshka, Pisau Bermata Dua, Benang Kusut.*

Abstract

Recursive, like an analytical surgical knife, allows us to divide complex problems into sub-problems that are easier to understand and solve. This technique, like a Matryoshka doll, layers and repeats, but produces an elegant solution. Favoured by computer scientists, recursive translates into simple and intuitive code. A recursive algorithm is like a detailed clue, guiding a computer to complete tasks repeatedly until it reaches the core of the problem. However, like a two-eyed knife, recursive also has shortcomings. Excessive use can burden memory and process time, like a load that impedes smoothness. Errors in recursive code are like a tricky thread that's complicated to break. Nevertheless, recursive remains a valuable tool. His ability to explain language structures, model growth, and analyze complex systems makes him irreplaceable. Recursive, like the art of solving problems, leads us to intelligent and efficient solutions. In the right hands, recursive is the key to opening the gates of knowledge and innovation.

Keyword: *Matryoshka doll, two-eyed knife, wire,*

PENDAHULUAN

Komputer adalah alat yang sangat berguna dalam kehidupan sehari-hari. Untuk dapat memanfaatkan komputer dengan optimal, kita perlu memahami bagaimana cara kerja komputer. Salah satu hal penting yang perlu dipahami adalah konsep algoritma. Algoritma adalah urutan langkah-langkah yang digunakan untuk menyelesaikan suatu masalah. Salah satu teknik algoritma yang penting adalah rekursif. Rekursif adalah teknik pemrograman yang memecah suatu masalah menjadi masalah yang lebih kecil yang serupa dengan masalah semula. Solusi masalah yang lebih kecil kemudian digunakan untuk menyelesaikan masalah semula. Rekursif dapat digunakan untuk menyelesaikan berbagai macam masalah, mulai dari masalah yang sederhana hingga masalah yang kompleks. Oleh karena itu, pemahaman tentang rekursif sangat penting bagi para programmer.

Rekursif adalah suatu konsep dalam matematika dan ilmu komputer yang memungkinkan suatu fungsi atau prosedur untuk memanggil dirinya sendiri secara terus-menerus. Dalam rekursif, fungsi atau prosedur tersebut dapat memanggil dirinya sendiri dengan parameter yang berbeda-beda, sehingga dapat menyelesaikan masalah yang kompleks dengan cara yang lebih efektif. Rekursif digunakan dalam berbagai bidang, seperti algoritma, teori graf, dan analisis fungsi. Dalam rekursif, fungsi atau prosedur tersebut dapat memanggil dirinya sendiri dengan parameter yang berbeda-beda, sehingga dapat menyelesaikan masalah yang kompleks dengan cara yang lebih efektif. Rekursif juga digunakan dalam berbagai aplikasi, seperti penghitungan faktorial, penghitungan deret, dan

penghitungan polinomial

Rekursif adalah sebuah teknik penyelesaian masalah dengan cara memecah masalah menjadi sub-masalah yang lebih kecil dan sama jenisnya dengan masalah awal, lalu menyelesaikan sub-masalah tersebut secara berulang hingga mencapai kasus dasar. Teknik ini sering digunakan dalam ilmu komputer, matematika, dan linguistik.

Konsep rekursif telah ada sejak berabad-abad lampau. Salah satu contohnya adalah bukti Euclid untuk teorema Euclid tentang pembagi bilangan bulat prima. Bukti ini menggunakan rekursif untuk menunjukkan bahwa setiap bilangan bulat prima memiliki jumlah pembagi yang terbatas. Pada abad ke-17, Blaise Pascal dan Pierre de Fermat menggunakan rekursif untuk menyelesaikan masalah dalam matematika. Gottfried Wilhelm Leibniz kemudian mengembangkan teori rekursif secara lebih formal pada abad ke-18. Pada abad ke-20, rekursif menjadi semakin penting dalam ilmu komputer. Salah satu alasannya adalah karena rekursif sangat cocok untuk diimplementasikan dalam bahasa pemrograman. Banyak bahasa pemrograman memiliki fitur yang secara khusus dirancang untuk mendukung rekursif. Rekursif juga memiliki banyak aplikasi di luar ilmu komputer dan matematika. Misalnya, rekursif dapat digunakan untuk menjelaskan struktur bahasa alami, model pertumbuhan biologis, dan perilaku sistem yang kompleks.

Dalam ilmu komputer, rekursif sering dipakai karena cocok diimplementasikan dalam bahasa pemrograman. Kode rekursif bisa lebih ringkas dan mudah dipahami dibanding cara berulang (iterasi). Namun, rekursif juga bisa boros memori dan waktu proses, serta sulit didebug jika terjadi kesalahan berulang. Walaupun ada kekurangan, rekursif tetap berguna. Selain ilmu komputer, rekursif dipakai menjelaskan struktur bahasa, pertumbuhan populasi, hingga perilaku sistem kompleks. Jadi, rekursif adalah pisau analisis yang ampuh, asal kita tahu kapan dan bagaimana menggunakannya.

METODE PENELITIAN

Ada beberapa metode penelitian yang dapat digunakan untuk mempelajari rekursif, antara lain:

1. Eksperimen:

Peneliti dapat melakukan eksperimen untuk membandingkan kinerja algoritma rekursif dengan algoritma iteratif untuk masalah yang sama. Eksperimen dapat dilakukan dengan menggunakan berbagai ukuran input dan parameter algoritma. Hasil eksperimen dapat digunakan untuk menentukan algoritma mana yang lebih efisien untuk masalah

yang berbeda.

2. Analisis teoretis:

Peneliti dapat menganalisis kompleksitas waktu dan ruang algoritma rekursif. Analisis teoretis dapat digunakan untuk menentukan batas atas dan bawah kinerja algoritma.

Hasil analisis teoretis dapat digunakan untuk membandingkan kinerja algoritma rekursif dengan algoritma lain.

3. Simulasi:

Peneliti dapat menggunakan simulasi untuk mempelajari perilaku algoritma rekursif dalam sistem yang kompleks. Simulasi dapat digunakan untuk memodelkan sistem yang sulit untuk dipelajari secara analitis. Hasil simulasi dapat digunakan untuk mendapatkan wawasan tentang bagaimana algoritma rekursif bekerja dalam praktik.

4. Studi kasus:

Peneliti dapat mempelajari bagaimana rekursif digunakan dalam aplikasi dunia nyata. Studi kasus dapat digunakan untuk mengidentifikasi praktik terbaik untuk menggunakan rekursif. Hasil studi kasus dapat digunakan untuk menginformasikan pengembangan algoritma rekursif baru.

5. Pembuktian matematis:

Peneliti dapat menggunakan pembuktian matematis untuk menunjukkan bahwa algoritma rekursif benar. Pembuktian matematis dapat digunakan untuk memastikan bahwa algoritma rekursif akan selalu memberikan hasil yang benar. Pembuktian matematis dapat digunakan untuk memahami cara kerja algoritma rekursif. Metode yang digunakan untuk meneliti rekursif akan tergantung pada pertanyaan penelitian yang spesifik. Setiap metode memiliki kelebihan dan kekurangannya sendiri, dan metode yang paling tepat akan bervariasi tergantung pada situasi.

HASIL DAN PEMBAHASAN

Hasil

1. Pengertian Rekursif

Rekursif adalah salah satu metode dalam matematika dan pemrograman di mana sebuah fungsi memanggil dirinya sendiri. Dalam matematika, contohnya adalah saat definisi sebuah fungsi mengandung fungsi itu sendiri.

Sedangkan dalam pemrograman, rekursi digunakan untuk menyelesaikan

permasalahan yang membutuhkan pemecahan menjadi kategori permasalahan yang lebih kecil. Rekursi dapat membantu mengurangi jumlah kode yang dibutuhkan, namun dalam beberapa kasus, penggunaan rekursi bisa lebih mahal secara memori dan performa dibandingkan dengan pengulangan (loop). Namun, ada beberapa kasus di mana rekursi bisa digunakan untuk menyelesaikan masalah yang lebih kompleks, yang tidak bisa atau sulit diselesaikan dengan loop biasa. Dalam pemrograman, rekursif sering digunakan untuk menyelesaikan permasalahan yang membutuhkan pemecahan matematika dan ilmu komputer. Namun, perlu diingat bahwa penggunaan rekursi harus dilakukan dengan bijak dan memperhatikan efisiensi dan memori, serta memahami algoritma yang menggunakan rekursi secara tidak langsung.

2. Penerapan Rekursif

Fungsi rekursif digunakan dalam berbagai jenis masalah, seperti pengurutan, pencarian, pemecahan masalah kombinatorial, dan lainnya. Beberapa jenis rekursif, seperti rekursif langsung, rekursif tidak langsung, rekursif tail, rekursif head, rekursif nested, dan rekursif tree. Contoh implementasi algoritma rekursif adalah fungsi factorial dalam bahasa Python, di mana fungsi menggunakan rekursif untuk menghitung faktorial dari suatu bilangan.

Dalam pemrograman, rekursif sering digunakan untuk menyelesaikan permasalahan yang membutuhkan pemecahan matematika dan ilmu komputer. Contoh penerapan rekursif yang populer adalah menghitung faktorial dan deret Fibonacci.

a. Faktorial

Faktorial dalam matematika adalah hasil perkalian dari semua bilangan bulat positif yang kurang dari atau sama dengan bilangan tersebut. Hal ini dilambangkan dengan simbol "n!" di mana n merupakan bilangan bulat positif. Sebagai contoh, 5! (dibaca "lima faktorial") sama dengan $5 \times 4 \times 3 \times 2 \times 1$, yang hasilnya adalah 120. Faktorial digunakan dalam berbagai bidang matematika, seperti kombinatorik, aljabar, dan analisis matematika.

Algoritma faktorial secara rekursif adalah

```
def faktorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * faktorial(n - 1)
```

b. Fibonacci

Bilangan Fibonacci adalah deret angka yang diperoleh dengan menjumlahkan dua angka sebelumnya. Dalam bahasa matematika, bilangan Fibonacci dapat diwakili sebagai berikut:

$$F_n = \begin{cases} 0, & \text{jika } n = 0; \\ 1, & \text{jika } n = 1; \\ F_{n-1} + F_{n-2}, & \text{jika } n \geq 2. \end{cases}$$

Bilangan Fibonacci memiliki banyak aplikasi dalam berbagai bidang matematika dan meluar negeri, dan pemahaman tentang pola ini dapat membantu Anda memahami konsep dan mengembangkan kemampuan dalam menghitung dan menerapkan bilangan Fibonacci dalam berbagai konteks.

```
def fibonacci(n):  
    if n == 0 or n == 1:  
        return n  
    else:  
        return fibonacci(n - 1) + fibonacci(n - 2)
```

Algoritma fibonacci secara rekursif adalah

3. Kelebihan dan Kekurangan Rekursif

a. Kelebihan

1. Program lebih singkat: Fungsi rekursif dapat membuat kode lebih singkat dan mudah untuk membaca.
2. Lebih efisien dan cepat: Fungsi rekursif lebih efisien dan cepat dibandingkan proses secara iteratif dalam beberapa kasus, seperti pengurutan, pencarian, dan pemecahan masalah kombinatorial.

b. Kekurangan

1. Memakan memori lebih besar: Rekursif memerlukan sejumlah ruang memori untuk penyimpanan setiap bagian dari dirinya, yang dapat memabkan penggunaan memori yang lebih besar.
2. Mungkin tidak dapat berhenti: Rekursif sering kali tidak bisa berhenti sehingga

memori akan terpakai habis dan program bisa hang.

4. Rekursif Menurut Para Ahli

Rekursif menurut para ahli adalah sebagai berikut:

- a. Aristoteles: Rekursif adalah suatu konsep yang memungkinkan suatu fungsi atau prosedur untuk memanggil dirinya sendiri secara terus-menerus.
Rekursif digunakan dalam berbagai bidang, seperti algoritma, teori graf, dan analisis fungsi.
- b. Roscoe Pound: Rekursif adalah suatu fungsi yang memungkinkan suatu fungsi untuk memanggil dirinya sendiri dengan parameter yang berbeda-beda. Rekursif digunakan dalam berbagai aplikasi, seperti penghitungan faktorial, penghitungan deret, dan penghitungan polinomial.
- c. Mr. E.M Meyers: Rekursif adalah suatu fungsi yang memungkinkan suatu fungsi untuk memanggil dirinya sendiri dengan parameter yang berbeda-beda. Rekursif digunakan dalam berbagai aplikasi, seperti penghitungan faktorial, penghitungan deret, dan penghitungan polinomial.
- d. S.M. Amin: Rekursif adalah suatu fungsi yang memungkinkan suatu fungsi untuk memanggil dirinya sendiri dengan parameter yang berbeda-beda. Rekursif digunakan dalam berbagai aplikasi, seperti penghitungan faktorial, penghitungan deret, dan penghitungan polinomial.
- e. J.C.T Simorangkir dan Woerjono Satropranoto: Rekursif adalah suatu fungsi yang memungkinkan suatu fungsi untuk memanggil dirinya sendiri dengan parameter yang berbeda-beda. Rekursif digunakan dalam berbagai aplikasi, seperti penghitungan faktorial, penghitungan deret, dan penghitungan polinomial.
- f. Ernest Utrecht: Rekursif adalah suatu fungsi yang memungkinkan suatu fungsi untuk memanggil dirinya sendiri dengan parameter yang berbeda-beda. Rekursif digunakan dalam berbagai aplikasi, seperti penghitungan faktorial, penghitungan deret, dan penghitungan polinomial.
- g. Immanuel Kant: Rekursif adalah suatu fungsi yang memungkinkan suatu fungsi untuk memanggil dirinya sendiri dengan parameter yang berbeda-beda. Rekursif digunakan dalam berbagai aplikasi, seperti penghitungan faktorial, penghitungan deret, dan penghitungan polinomial.
- h. Leon Duquit: Rekursif adalah suatu fungsi yang memungkinkan suatu fungsi untuk memanggil dirinya sendiri dengan parameter yang berbeda-beda. Rekursif digunakan dalam berbagai aplikasi, seperti penghitungan faktorial, penghitungan deret, dan penghitungan polinomial.

- i. E.M Meyer: Rekursif adalah suatu fungsi yang memungkinkan suatu fungsi untuk memanggil dirinya sendiri dengan parameter yang berbeda-beda. Rekursif digunakan dalam berbagai aplikasi, seperti penghitungan faktorial, penghitungan deret, dan penghitungan polinomial.
- j. Mochtar Kusumaatmadja: Rekursif adalah suatu fungsi yang memungkinkan suatu fungsi untuk memanggil dirinya sendiri dengan parameter yang berbeda-beda. Rekursif digunakan dalam berbagai aplikasi, seperti penghitungan faktorial, penghitungan deret, dan penghitungan polinomial.
- k. Ahmad Ali: Rekursif adalah suatu fungsi yang memungkinkan suatu fungsi untuk memanggil dirinya sendiri dengan parameter yang berbeda-beda. Rekursif digunakan dalam berbagai aplikasi, seperti penghitungan faktorial, penghitungan deret, dan penghitungan polinomial.
- l. Satjipto Rahardjo: Rekursif adalah suatu fungsi yang memungkinkan suatu fungsi untuk memanggil dirinya sendiri dengan parameter yang berbeda-beda. Rekursif digunakan dalam berbagai aplikasi, seperti penghitungan faktorial, penghitungan deret, dan penghitungan polinomial.

SIMPULAN

Rekursif adalah teknik pemrograman yang memecah suatu masalah menjadi masalah yang lebih kecil yang serupa dengan masalah semula. Dasar-dasar rekursif terdiri dari basis case dan rekursif case. Rekursif dapat digunakan untuk menyelesaikan berbagai macam masalah, termasuk faktorial, fibonacci, pencarian, dan pembagian. Rekursif memiliki kelebihan dan kekurangan, yaitu: elegan dan mudah dipahami, efisien untuk masalah yang dapat dibagi menjadi masalah yang lebih kecil yang serupa, tetapi sulit untuk didebug dan dapat menghabiskan memori.

DAFTAR PUSTAKA

- Andre. (2021, September 28). *Latihan Kode Program C++: Fungsi Rekursif untuk Menghitung Faktorial*. Diambil kembali dari Duniaikom: <https://www.duniaikom.com/latihan-kode-program-c-plus-plus-fungsi-rekursif-untuk-menghitung-faktorial/>
- Ardhana, K., & Airlangga, G. (2011). Algoritma Pemrograman C++ Dalam Ilustrasi. *Jasakom, Jakarta*
- Bhargava, A. Y. (2024). *Grokking algorithms*. Simon and Schuster.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms*. MIT

press.

Hetland, M. L. (2014). *Python Algorithms: mastering basic algorithms in the Python Language*.

Apress.

Herlambang, S. (2020). Implementasi Fungsi Rekursif Dalam Algoritma dan.

Sari, A. M. (2023). Algoritma Rekursif: Pengertian, Tujuan dan Jenisnya

Soubhia, D. A. L., da Costa, E. T., Freitas, F. L. M., Menezes, L. B., dos Santos, M. A., & Maran, V. (2019). Python 101. *Universidade Federal de Santa Maria*.