**Analisis Kompleksitas Binery Search dalam**

**Meningkatkan Profitabilitas Perkebunan Sawit Melalui Penerapan Teknik Budidaya yang Efektif**

Dosen Pengampu :

Selly Meliana, M.Kom

****

Disusun Oleh :

Aziz Syanjaya 103012300300

Brian Julio Arsibald 103012300388

Program Studi S1 Informatika

Fakultas Informatika

Telkom University

2023/2024

# Daftar Isi

[1. Daftar Isi 2](#_Toc185684107)

[2. Pendahuluan 3](#_Toc185684108)

[2.1. Latar Belakang 3](#_Toc185684109)

[2.2. Tujuan Penelitian 3](#_Toc185684110)

[2.3. Ruang Lingkup 3](#_Toc185684111)

[3. Deskripsi Studi Kasus Permasalahan 3](#_Toc185684112)

[3.1. Penjelasan masalah 3](#_Toc185684113)

[3.2. Konteks dan Relevasi dalam 3](#_Toc185684114)

[4. Deskripsi Algoritma 4](#_Toc185684115)

[4.1. Algoritma Binary Search 4](#_Toc185684116)

[a. Penjelasan dan Cara Kerja 4](#_Toc185684117)

[b. Kelebihan dan Kekurangan 4](#_Toc185684118)

[4.2. Binary Search Iteratif 4](#_Toc185684119)

[a. Penjelasan dan Cara Kerja 4](#_Toc185684120)

[b. Kelebihan dan Kekurangan 4](#_Toc185684121)

[5. Grafik Perbandingan Running Time 5](#_Toc185684122)

[5.1. Metodologi Pengujian 5](#_Toc185684123)

[5.2. Hasil Pengujian 5](#_Toc185684124)

[5.3. Analisis Grafik 5](#_Toc185684125)

[6. Analisi Perbandingan Kedua Algoritma 5](#_Toc185684126)

[6.1. Kriteria Perbandingan 5](#_Toc185684127)

[6.2. Hasil Analisis 5](#_Toc185684128)

[6.3. Kesimpulan dari Analisis 5](#_Toc185684129)

[7. Kesimpulan 5](#_Toc185684130)

[7.1. Ringkasan Temuan 5](#_Toc185684131)

[7.2. Saran untuk Penelitian Selanjutnya 5](#_Toc185684132)

[8. Referensi 5](#_Toc185684133)

# Pendahuluan

## Latar Belakang

Perkebunan sawit di Indonesia, khususnya di Kalimantan dan Sumatera, memiliki peran penting dalam perekonomian nasional. Namun, sektor ini menghadapi tantangan signifikan, termasuk rendahnya produktivitas. Banyak perkebunan sawit di Indonesia yang tidak dapat mencapai potensi maksimalnya akibat teknik budidaya yang digunakan masih kurang efektif dan pengelolaan sumber daya yang tidak efisien. Hal ini menyebabkan profitabilitas yang rendah, yang berdampak pada kesejahteraan petani dan pelaku usaha di sektor ini.

## Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengidentifikasi teknik budidaya yang optimal dan dapat meningkatkan profitabilitas perkebunan sawit. Secara spesifik, penelitian ini bertujuan untuk :

* Memaksimalkan penggunaan lahan untuk penanaman kelapa sawit.
* Memberikan rekomendasi praktis bagi para pelaku usaha perkebunan sawit untuk meningkatkan efisiensi dan efektivitas teknik budidaya yang diterapkan.

## Ruang Lingkup

Ruang lingkup penelitian ini mencakup analisis teknik budidaya sawit yang diterapkan di pulau Kalimantan dan Sumatera. Penelitian ini akan menggunakan algoritma optimasi teknik budidaya sawit, yang tersedia dalam dua versi, yaitu iteratif dan rekursif. Kedua versi algoritma ini akan digunakan untuk menganalisis data yang berkaitan dengan teknik budidaya. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi pada pengembangan ilmu pengetahuan dan teknologi di bidang perkebunan sawit serta meningkatkan profitabilitas usahan perkebunan.

# Deskripsi Studi Kasus Permasalahan

## Penjelasan masalah

Permasalahan utama yang dihadapi oleh perkebunan sawit di Indonesia adalah rendahnya produktivitas. Banyak perkebunan yang tidak dapat mencapai potensi maksimalnya karena teknik budidaya yang kurang efektif. Hal ini menyebabkan profitabilitas yang rendah, yang berdampak negatif pada kesejahteraan petani dan pelaku usaha di sektor ini.

## Konteks dan Relevasi

Konteks dari penelitian ini berfokus pada industri perkebunan sawit di Kalimantan dan Sumatera, yang merupakan dua daerah utama penghasil sawit di Indonesia. Dengan meningkatnya permintaan global terhadap minyak sawit, penting bagi para pelaku usaha untuk meningkatkan produktivitas dan efisiensi. Penelitian ini relevan karena dapat memberikan wawasan dan rekomendasi praktis bagi petani dan pengelola perkebunan dalam menerapkan teknik budidaya yang lebih baik, serta berkontribusi pada peningkatan kesejahteraan ekonomi masyarakat di daerah tersebut.

# Deskripsi Algoritma

## Algoritma Binary Search Iteratif

### Penjelasan dan Cara Kerja

### Pendekatan iteratif menggunakan loop (dalam hal ini, *while*) untuk menanam pohon sawit. Setiap iterasi menambah jumlah pohon yang ditanam sebanyak *TREES\_PER\_DAY* hingga mencapai jumlah total yang diinginkan *JumSawit*.

### Kelebihan dan Kekurangan

Kelebihan :

* Efisiensi Memori : Pendekatan ini lebih efisien dalam penggunaan memori karena tidak memerlukan stack tambahan untuk setiap panggilan fungsi, sehingga mengurangi risiko stack overflow.
* Kecepatan Eksekusi : Dalam banyak kasus, pendekatan iteratif dapat lebih cepat karena tidak ada overhead dari pemanggilan fungsi berulang.

Kekurangan :

* Keterbacaan : Kode iteratif bisa jadi kurang intuitif dibandingkan dengan kode rekursif, terutama untuk masalah yang secara alami lebih cocok untuk pendekatan rekursif.
* Penggunaan Variabel : Memerlukan lebih banyak variabel untuk melacak status seperti *SawitDitanam*.

## Binary Search Rekursif

### Penjelasan dan Cara Kerja

### Pendekatan rekursif memanggil fungsi *tanamPohonRekursif* secara berulang, di mana setiap panggilan menambah jumlah pohon yang ditanam hingga mencapai target.

### Kelebihan dan Kekurangan

Kelebihan :

* Keterbacaan : Kode rekursif sering kali lebih mudah dibaca dan dipahami, terutama untuk masalah yang memiliki struktur berulang.
* Sederhana : Mengurangi kompleksitas logika dengan memecah masalah menjadi sub-masalah yang lebih kecil.

Kekurangan :

* Efisiensi Memori : Setiap panggilan fungsi menambah frame baru ke stack, yang dapat menyebabkan penggunaan memori yang lebih tinggi dan berpotensi menyebabkan stack overflow jika jumlah pohon yang ditanam sangat besar.
* Kecepatan Eksekusi : Ada overhead dari pemanggilan fungsi yang dapat membuat eksekusi lebih lambat dibandingkan dengan pendekatan iteratif.

### Langkah- langkah Cara Kerja Kode

1. Inisialisasi Konstanta dan Variabel :

* Kode dimulai dengan mendeklarasikan konstanta *TREES\_PER\_DAY* yang diatur ke 60, yang menunjukkan jumlah pohon yang dapat ditanam per hari.
* Variabel *hektar, JumSawit,*  dan  *SawitDitanam* diinisialisasikan. *SawitDitanam*  diatur ke 0 sebagai jumlah pohon yang telah ditanam.

1. Input dari Pengguna :

* Program meminta pengguna untuk memasukkan jumlah hektar tanah yang ingin ditanami. Nilai ini disimpan dalam variable *hektar.*

1. Menghitung Jumlah total Pohon :

* Jumlah total pohon sawit yang akan ditanam dihitung dengan mengalikan jumlah hektar dengan 150 (asumsi jumlah pohon per hektar) dan disimpan dalam variabel *JumSawit.*
* Program menampilkan total pohon yang akan ditanam.

1. Pendekatan Iteratif :

* Program mencatat waktu mulai untuk eksekusi iteratif menggunakan *chrono.*
* Menggunakan loop *while*, program terus menambah jumlah pohon yang ditanam (*SawitDitanam*) sebanyak *TREES\_PER\_DAYS* hingga jumlah pohon yang ditanam mencapai atau melebihi *JumSawit*.
* Setiap kali pohon ditanam, program menampilkan jumlah pohon yang telah ditanam.
* Setelah loop selesai, program mencatat waktu akhir dan menghitung durasi waktu eksekusi untuk pendekatan iteratif, kemudian menampilkannya.

1. Pendekatan Rekursif :

* Program mencatat waktu mulai untuk eksekusi iteratif menggunakan *chrono.*
* Fungsi *tanamPohonRekursif* dipanggil dengan parameter *JumSawit* dan 0 (jumlah pohon yang telah ditanam).
* Didalam fungsi rekursif :

1. Base case : Jika jumlah pohon yang ditanam (*SawitDitanam*) sudah mencapai atau melebihi *JumSawit*, program menampilkan pesan bahwa semua pohon telah ditanam dan fungsi kembali.
2. Jika belum, jumlah pohon yang ditanam ditambah *TREES\_PER\_DAYS*, dan program menampilkan jumlah pohon yang telah ditanam.
3. Fungsi *tanamPohonRekursif* dipanggil kembali dengan parameter yang diperbarui.

* Setelah semua pohon ditanam, program mencatat waktu akhir dan menghitung durasi waktu eksekusi untuk pendekatan rekursif, kemudian menampilkannya.

1. Akhir Program :

* Program selesai dan mengembalikan nilai 0, menandakan bahwa eksekusi program berhasil.

# Grafik Perbandingan Running Time

## Metodologi Pengujian

Pengujian dilakukan dengan :

1. Perbandingan Waktu Eksekusi
2. Efisiensi Memori

## Hasil Pengujian

A screenshot of a computer

Description automatically generated

Menunjukkan bahwa untuk masalah ini, pendekatan iteratif lebih disukai dalam hal efisiensi waktu dan memori. Namun, pendekatan rekursif tetap memiliki nilai dalam hal keterbacaan dan pemecahan masalah yang lebih kompleks.

## Analisis Grafik

A graph with a red line

Description automatically generated

Grafik ini menunjukkan bahwa pendekatan iteratif lebih efisien dalam hal waktu eksekusi dibandingkan dengan pendekatan rekursif, terutama saat jumlah hektar yang ditanami meningkat. Ini mendukung temuan sebelumnya bahwa meskipun pendekatan rekursif lebih mudah dibaca, pendekatan iteratif lebih cepat dan lebih efisien dalam penggunaan sumber daya. Maka kompleksitas waktunya atau T(n) nya adalah O(n).

# Analisi Perbandingan Kedua Algoritma

## Kriteria Perbandingan

Perbandingan dilakukan berdasarkan :

1. Waktu Eksekusi : Mengukur berapa lama masing-masing pendekatan (iteratif dan rekursif) memerlukan waktu untuk menyelesaikan proses menanam pohon.
2. Penggunaan Memori:  Menilai seberapa mudah kode dapat dipahami dan diikuti oleh pengembang lainnya.
3. Kemudahan Implementasi: Menilai seberapa mudah masing-masing pendekatan dapat diimplementasikan dalam konteks yang lebih besar.

## Hasil Analisis

Waktu Eksekusi: Pendekatan iteratif menunjukkan waktu eksekusi yang lebih cepat dibandingkan dengan pendekatan rekursif. Ini disebabkan oleh overhead yang lebih rendah dalam pemanggilan fungsi pada pendekatan iteratif.

Penggunaan Memori: Pendekatan iteratif lebih efisien dalam penggunaan memori karena tidak memerlukan stack tambahan untuk setiap panggilan fungsi. Sebaliknya, pendekatan rekursif dapat menyebabkan penggunaan memori yang lebih tinggi, terutama jika jumlah pohon yang ditanam sangat besar.

Kemudahan Implementasi: Kedua pendekatan dapat diimplementasikan dengan relatif mudah, tetapi pendekatan iteratif mungkin lebih disarankan untuk masalah yang lebih sederhana.

## Kesimpulan dari Analisis

Hasil analisis menunjukkan bahwa untuk masalah menanam pohon sawit, pendekatan iteratif lebih efisien dalam hal waktu eksekusi dan penggunaan memori dibandingkan dengan pendekatan rekursif.

# Kesimpulan

## Ringkasan Temuan

Berdasarkan analisis yang dilakukan terhadap kode program yang mengimplementasikan dua pendekatan untuk menanam pohon sawit, yaitu pendekatan iteratif dan rekursif, dapat disimpulkan sebagai berikut :

* Efisiensi Waktu : Pendekatan iteratif menunjukkan waktu eksekusi yang lebih cepat dibandingkan dengan pendekatan rekursif. Hal ini disebabkan oleh kurangnya overhead dari pemanggilan fungsi berulang pada pendekatan iteratif.
* Penggunaan Memori : Pendekatan iteratif lebih efisien dalam penggunaan memori, karena tidak memerlukan stack tambahan untuk setiap panggilan fungsi. Sebaliknya, pendekatan rekursif dapat menyebabkan penggunaan memori yang lebih tinggi, terutama ketika jumlah pohon yang ditanam sangat besar.
* Keterbacaan Kode : Meskipun kode rekursif lebih mudah dibaca dan dipahami, terutama untuk masalah yang memiliki struktur berulang, pendekatan iteratif tetap cukup jelas dan mudah diimplementasikan untuk masalah sederhana ini.

Secara keseluruhan, pemilihan antara pendekatan iteratif dan rekursif harus mempertimbangkan konteks spesifik dari aplikasi yang sedang dikembangkan, serta kebutuhan akan efisiensi waktu dan penggunaan memori.

# Referensi

* Adhynugraha Santo, POTENSI DAN PERMASALAHAN PENGEMBANGAN PERKEBUNAN KELAPA SAWIT SKALA BESAR DI KALIMANTAN TIMUR