

Binary search algorithm: pencarian email berdasarkan nama menggunakan flowgorithm

Sunandar Suherman

program studi Teknik Informatika, UIN Maulana Malik Ibrahim Malang;
email: sunandarsuherman77@gmail.com

Kata Kunci:

binary search, pencarian email, Flowgorithm, data, perangkat lunak

Keywords:

binary search, email search, Flowgorithm, data, software

ABSTRAK

Pencarian email berdasarkan nama merupakan salah satu tugas yang sering dilakukan dalam sistem informasi email. Tugas ini dapat dilakukan dengan berbagai algoritma, salah satunya adalah algoritma binary search. Algoritma binary search merupakan algoritma pencarian yang bekerja dengan membagi kumpulan data menjadi dua bagian secara berulang-ulang hingga data yang dicari ditemukan. Dalam penelitian ini, algoritma binary search digunakan untuk pencarian email berdasarkan nama menggunakan Flowgorithm. Flowgorithm merupakan sebuah perangkat lunak diagram alir yang dapat digunakan untuk membuat algoritma

secara visual. Hasil penelitian menunjukkan bahwa algoritma binary search dapat digunakan untuk pencarian email berdasarkan nama dengan efisien. Algoritma ini dapat menemukan data yang dicari dalam waktu yang relatif singkat, yaitu $O(\log n)$.

ABSTRACT

Searching email by name is one of the tasks that is often done in email information systems. This task can be done with various algorithms, one of which is the binary search algorithm. The binary search algorithm is a search algorithm that works by repeatedly dividing the data set into two parts until the data being searched is found. In this study, the binary search algorithm is used to search for emails by name using Flowgorithm. Flowgorithm is a flowchart software that can be used to create algorithms visually. The results of the study show that the binary search algorithm can be used to search for emails by name efficiently. This algorithm can find the data being searched in a relatively short time, which is $O(\log n)$.

Pendahuluan

Di era keterhubungan digital yang semakin meningkat, email telah menjadi alat komunikasi yang tak terpisahkan bagi banyak orang. Sejalan dengan peningkatan volume email yang dikirim dan diterima setiap hari, kebutuhan untuk menemukan email tertentu menjadi semakin penting. Salah satu tugas yang sering dihadapi dalam sistem informasi email adalah pencarian email berdasarkan nama [1]. Pencarian email berdasarkan nama dapat dilakukan dengan berbagai algoritma pencarian. Salah satu algoritma yang efisien dan banyak digunakan adalah algoritma binary search [2]. Algoritma ini bekerja dengan membagi kumpulan data menjadi dua bagian secara berulang-ulang hingga data yang dicari ditemukan [3]. Kompleksitas waktu algoritma binary search, sebagaimana dijelaskan oleh Knuth [2], adalah $O(\log n)$, di mana n adalah jumlah data. Ini berarti bahwa waktu yang dibutuhkan untuk menemukan data yang dicari berbanding terbalik dengan logaritma dari jumlah data [4].

Dalam penelitian ini, algoritma binary search diterapkan untuk pencarian email berdasarkan nama menggunakan Flowgorithm [5]. Flowgorithm adalah perangkat



This is an open access article under the [CC BY-NC-SA](https://creativecommons.org/licenses/by-nc-sa/4.0/) license.

Copyright © 2023 by Author. Published by Universitas Islam Negeri Maulana Malik Ibrahim Malang.

lunak diagram alir yang memungkinkan pembuatan algoritma secara visual. Flowgorithm juga merupakan alat pembuat grafis yang memungkinkan pengguna untuk menulis dan menjalankan program menggunakan diagram alur. Ini dirancang untuk menekankan algoritma daripada sintaks bahasa pemrograman tertentu. Pendekatannya terstruktur dan penting, dan alat ini mendukung berbagai bahasa lisan. Flowgorithm dapat menerjemahkan program diagram alur secara interaktif ke dalam kode sumber yang ditulis dalam bahasa pemrograman lain, seperti C#, C++, Java, Python, dan lainnya. Ini adalah bahasa pemrograman pemula gratis berdasarkan diagram alur grafis sederhana, sehingga memudahkan siswa untuk berkonsentrasi pada konsep pemrograman. Alat ini juga dilengkapi Penampil Kode Sumber yang memungkinkan diagram alur dikonversi ke beberapa bahasa pemrograman dunia nyata [6]. Kemudahan penggunaan Flowgorithm, baik dalam pembuatan diagram alir maupun eksekusi algoritma, menjadikannya pilihan yang tepat untuk penelitian ini [7]. Tujuan dari penelitian ini adalah untuk menganalisis kinerja algoritma binary search dalam pencarian email berdasarkan nama menggunakan Flowgorithm. Diharapkan penelitian ini dapat memberikan gambaran tentang efektivitas algoritma binary search dalam konteks pencarian email berdasarkan nama.

Metode Penelitian

Dalam penelitian pembuatan aplikasi program flowchart, metode yang digunakan adalah teori pemecahan permasalahan. Teori ini digunakan untuk mencari susunan data yang dibutuhkan dalam pemecahan masalah. Data tersebut kemudian disusun dalam bentuk program flowchart. Program flowchart ini kemudian divisualisasi dalam bentuk aplikasi yang dapat digunakan pengguna untuk menyelesaikan masalah seperti yang dapat dilihat pada Gambar 1.



Gambar 1 Metode Penelitian

Pengumpulan Data (Data Collection)

Dalam penelitian ini menggunakan data random yang dibuat oleh AI untuk mengumpulkan data nama dan email. Untuk saat ini, data yang berhasil dikumpulkan disimpan pada tabel nama dan email dapat dilihat pada tabel dibawah ini.

No	Nama	Email
1.	Adi	adi@gmail.com
2.	Agus	agus@yahoo.com
3.	Aini	aini@hotmail.com
4.	Andi	andi@gmail.com
5.	Asep	asep@outlook.com
6.	Ayu	ayu@live.com
7.	Bima	bima@ymail.com
8.	Budi	budi@yahoo.com
9.	Cici	cici@hotmail.com
10.	Citra	citra@aol.com
11.	Dedi	dedi@zoho.com
12.	Dina	dina@outlook.com
13.	Eka	eka@protonmail.com
14.	Fajar	fajar@mail.com
15.	Fani	fani@ymail.com
16.	Galih	galih@me.com
17.	Gina	gina@aol.com
18.	Hana	hana@icloud.com
19.	Hendi	hendi@zoho.com
20.	Hendra	hendra@mac.com
21.	Ilham	ilhaam@outlook.com
22.	Ira	ira@protonmail.com
23.	Jaka	jaka@live.com
24.	Joko	joko@mail.com
25.	Kadek	kadek@yahoo.com
26.	Kiki	kiki@me.com
27.	Lili	lili@icloud.com
28.	Lintang	lintang@gmail.com
29.	Melani	melani@hotmail.com
30.	Mia	mia@mac.com
31.	Nadia	nadia@outlook.com
32.	Nanda	nanda@outlook.com
33.	Oki	oki@live.com
34.	Panji	panji@yahoo.com
35.	Putra	putra@ymail.com
36.	Ratna	ratna@aol.com
37.	Rini	rini@gmail.com
38.	Sari	sari@htmail.com
39.	Shinta	shinta@zoho.com
40.	Tiara	tiara@protonmail.com
41.	Tika	tika@outlook.com
42.	Udin	udin@mail.com
43.	Vania	vania@me.com
44.	Vira	vira@ymail.com
45.	Wanda	wanda@aol.com
46.	Wulan	wulan@icloud.com
47.	Yoga	yoga@mac.com

48.	Yuni	yuni@zoho.com
49.	Zara	zara@protonmail.com
50.	Zidan	zidan@outlook.com

Tabel 1 Data nama dan email

Menganalisis Permasalahan

Penggunaan binary search dalam pencarian email memberikan dampak signifikan terhadap waktu eksekusi, terutama dalam skenario database yang besar. Dalam binary search, waktu eksekusi tumbuh secara logaritmik dengan peningkatan jumlah data. Kompleksitas waktu binary search, $O(\log\{n\})$, mengindikasikan bahwa waktu eksekusi meningkat secara proporsional terhadap logaritma basis 2 dari jumlah elemen dalam database ((n)).

$O(\log\{n\})$ Kompleksitas Waktu Binary Search

Perbandingan ini menunjukkan bahwa binary search memerlukan jumlah langkah yang jauh lebih rendah dibandingkan metode pencarian linier dalam menemukan data. Dengan kata lain, seiring dengan peningkatan data, binary search memiliki pertumbuhan waktu eksekusi yang lebih lambat, yang sangat menguntungkan dalam situasi di mana kinerja dan responsivitas sistem menjadi kritis. Sebagai contoh, mari ambil kasus database dengan 1 juta elemen ($(n = 10^6)$). Dengan binary search, waktu eksekusi hanya memerlukan sekitar 20 langkah ($(\log_2\{10^6\} \approx 20)$), sedangkan pencarian linier akan membutuhkan 1 juta langkah. Ini menunjukkan bahwa binary search memberikan keunggulan yang signifikan dalam mengurangi waktu pencarian, membuatnya sangat cocok untuk aplikasi dengan skala data besar.

Pengaruh terhadap efisiensi pengelolaan sumber daya komputasi

Binary search memiliki sejumlah dampak positif terhadap efisiensi pengelolaan sumber daya komputasi, yaitu:

- Pemrosesan yang lebih efisien: Binary search, dengan kompleksitas waktu logaritmik, memungkinkan sistem untuk memproses pencarian dengan cara yang lebih efisien. Jumlah perbandingan yang lebih rendah meminimalkan beban pemrosesan pada unit pemrosesan pusat (CPU), menghasilkan eksekusi yang lebih cepat, dan meningkatkan kinerja sistem secara keseluruhan.
- Penanganan jumlah data yang besar: Ketika jumlah data dalam database semakin besar, binary search tetap mampu menangani pencarian dengan waktu eksekusi yang relatif rendah. Ini menghindari peningkatan waktu eksekusi secara linear yang biasa terjadi pada metode pencarian linier. Sebagai hasilnya, binary search memungkinkan sistem untuk tetap efisien bahkan ketika menghadapi skenario dengan volume data yang sangat besar.
- Optimasi penggunaan memori: Selain mengoptimalkan waktu eksekusi, binary search juga dapat mengoptimalkan penggunaan memori. Implementasi yang baik dari algoritma ini dapat memastikan bahwa sistem dapat bekerja dengan jumlah

data yang besar tanpa memerlukan alokasi memori yang berlebihan. Hal ini sangat berarti dalam situasi di mana sumber daya memori terbatas.

Optimasi dan tantangan dalam implementaasi

Dalam implementasinya, binary search memiliki beberapa tantangan, yaitu:

- a) Pemeliharaan pengurutan database: Salah satu tantangan dalam penggunaan binary search adalah memastikan bahwa database tetap terurut setiap kali ada perubahan. Pemeliharaan ini diperlukan agar binary search dapat terus bekerja dengan efisien. Strategi seperti algoritma pengurutan yang cepat dan efisien perlu diadopsi untuk mengurangi dampak perubahan data.
- b) Pertimbangan karakteristik khusus: Dalam implementasi binary search untuk pencarian email berdasarkan nama pengguna, karakteristik khusus seperti penanganan huruf besar/kecil atau karakter khusus perlu diperhatikan. Pemrosesan string yang efisien dan akurat menjadi penting untuk memastikan pencarian yang tepat.
- c) Optimasi teknik pemrosesan string: Penerapan teknik pemrosesan string yang efisien dapat mempercepat perbandingan antara nama pengguna target dan elemen tengah database. Teknik ini melibatkan penggunaan algoritma dan struktur data yang sesuai untuk meminimalkan overhead pemrosesan string.

Jadi dapat disimpulkan bahwa binary search adalah algoritma pencarian yang efisien dan efektif untuk menangani data yang besar. Kompleksitas waktu logaritmiknya memungkinkan binary search untuk bekerja dengan cepat dan efisien, bahkan ketika jumlah data semakin besar. Namun, dalam implementasinya, binary search memiliki beberapa tantangan yang perlu diperhatikan, seperti pemeliharaan pengurutan database dan penanganan karakteristik khusus.

Pengembangan Aplikasi

pengembangan aplikasi adalah proses pembuatan atau penyempurnaan suatu aplikasi perangkat lunak agar dapat bekerja dengan baik sesuai kebutuhan pengguna. Proses pengembangan meliputi analisis persyaratan, desain, implementasi kode program, pengujian, dan pemeliharaan. Tujuan utamanya adalah menciptakan aplikasi yang memberikan nilai kepada audiens target. Analisis persyaratan adalah langkah awal dalam pengembangan aplikasi. Pada tahap ini, tim pengembangan akan mengidentifikasi kebutuhan pengguna dan bisnis. Kebutuhan ini dapat diidentifikasi dengan melakukan wawancara, survei, atau observasi.

Salah satu teknik dalam mengembangkan aplikasi adalah dengan menggunakan pencarian biner. Pencarian biner adalah algoritma pencarian yang digunakan untuk menemukan nilai tertentu dalam sebuah larik yang sudah terurut. Algoritma ini bekerja dengan membagi larik menjadi dua bagian secara berulang, dan kemudian membandingkan nilai yang dicari dengan nilai tengah larik. Jika nilai yang dicari lebih besar dari nilai tengah, maka pencarian dilanjutkan pada bagian kanan larik. Sebaliknya, jika nilai yang dicari lebih kecil dari nilai tengah, maka pencarian dilanjutkan pada bagian

kiri larik. Proses ini terus berlanjut hingga nilai yang dicari ditemukan atau tidak ada lagi data yang tersisa.

Bagan organisasi pencarian biner biasanya terdiri dari dua bagian utama, yaitu: Bagian awal: Bagian ini menggambarkan langkah-langkah awal algoritme pencarian biner, yaitu:

- a) Inisialisasi variabel.
- b) Melakukan perbandingan nilai yang dicari dengan nilai tengah larik.
- c) Bagian rekursif: Bagian ini menggambarkan langkah-langkah rekursif algoritme pencarian biner, yaitu:
- d) Jika nilai yang dicari lebih kecil dari nilai tengah, maka pencarian dilanjutkan pada bagian kiri larik.
- e) Jika nilai yang dicari lebih besar dari nilai tengah, maka pencarian dilanjutkan pada bagian kanan larik.
- f) Jika nilai yang dicari tidak ditemukan, maka keluar dari fungsi pencarian.

Berikut adalah contoh bagan organisasi pencarian biner:

Awal

Inisialisasi variabel

Perbandingan nilai yang dicari dengan nilai tengah larik

Rekursif

Jika nilai yang dicari lebih kecil dari nilai tengah

Pencarian biner (larik kiri, nilai yang dicari)

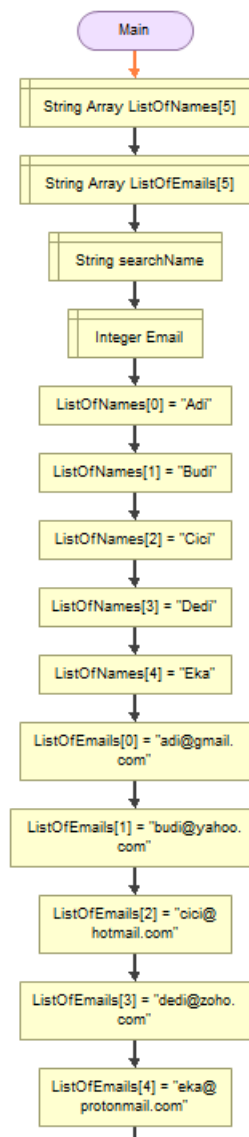
Jika nilai yang dicari lebih besar dari nilai tengah

Pencarian biner (larik kanan, nilai yang dicari)

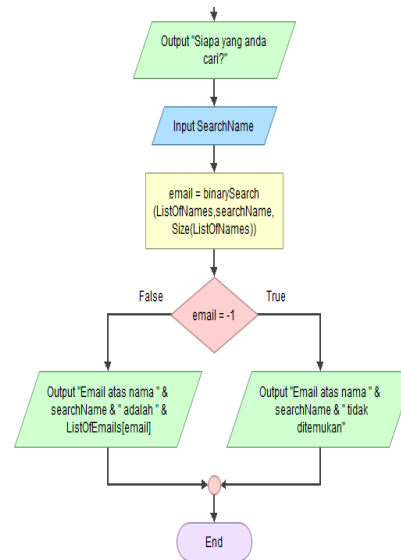
Jika nilai yang dicari tidak ditemukan

Selesai

Bagan organisasi pencarian biner dapat membantu programmer untuk memahami algoritme pencarian biner secara lebih mudah. Dengan memahami algoritme secara lebih mudah, maka programmer akan lebih mudah untuk mengimplementasikan algoritme pencarian biner dalam aplikasi. Selain bagan organisasi, algoritme pencarian biner juga dapat dimodelkan dengan diagram alir. Diagram alir adalah representasi visual dari suatu proses dengan menggunakan simbol-simbol standar. Diagram alir pencarian biner dapat membantu programmer untuk memahami algoritme pencarian biner secara lebih mudah, terutama bagi programmer yang lebih familiar dengan diagram alir. Dalam kasus mencari email berdasarkan nama menggunakan flowgorithm setelah mengumpulkan data nama dan juga email setelah itu diubah kedalam bentuk flowchart.

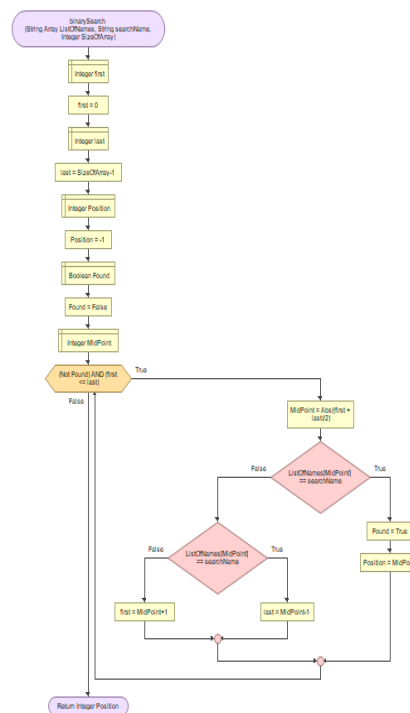


Gambar2.1 Deklarasi



Gambar 2.3 Interaksi program dan pengguna

Sebuah flowchart sederhana telah dirancang untuk mempermudah pengguna dalam menggunakan program dengan menginput perintah. Program ini menggunakan struktur while untuk melakukan perulangan, memastikan bahwa data yang dimasukkan oleh pengguna benar. Selain itu, penggunaan struktur if digunakan untuk mengarahkan program ke percabangan yang sesuai, terutama ketika pengguna diminta untuk memasukkan nama dari email yang akan dicari. Proses dimulai dengan meminta input dari pengguna, dan jika data yang dimasukkan tidak valid, program akan mengulang permintaan input. Langkah terakhir melibatkan pengisian data nama dan juga data email ke dalam array program. Agar flowchart tidak terlihat rumit atau panjang maka dibuatlah fungsi lain untuk menambahkan rumus pencarian binernya yang dapat dipanggil di fungsi utama ini dapat mempermudah dan menjadi lebih efisien.



Gambar 2.4 Fungsi binary search

Pengoperasian algoritma pencarian biner adalah sebagai berikut:

1. Pastikan data terurut entah itu secara meningkat atau menurun.
2. Tetapkan batas bawah (low) dan batas atas (high) untuk rentang data.
3. Hitung titik tengah (midpoint) dari batas bawah dan batas atas menggunakan rumus: $\text{midpoint} = (\text{low} + \text{high}) / 2$.
4. Bandingkan nilai tengah dengan nilai target. Terdapat 3 skenario:
 - a) Jika nilai tengah sama dengan target, maka lokasi data telah ditemukan.
 - b) Jika target kurang dari nilai tengah, perbarui batas atas menjadi $\text{midpoint} - 1$.
 - c) Jika target lebih besar dari nilai tengah, perbarui batas bawah menjadi $\text{midpoint} + 1$.
5. Ulangi langkah 3 dan 4 hingga target berhasil ditemukan.
6. Jika batas bawah melebihi batas atas, artinya data tidak ditemukan.

Setelah flowchart atau flowgorithm program selesai, langkah terakhir adalah mengubahnya menjadi program dalam bahasa pemrograman yang diinginkan. Hal ini bertujuan untuk mengoptimalkan dan mengembangkan program agar lebih efisien dan cepat dalam memproses dan mengelola data yang diinput oleh pengguna.

Pengujian Aplikasi

Setelah program pencarian email berdasarkan nama dikembangkan dengan menggunakan slgoritma pencarian biner . maka dilakukanlah pengujian untuk memastikan program tersebut bisa berjalan sesuai yang diharapkan, pengujian ini juga dilakkukan untukk uji kelayakan program yang sedang dikembangkan.

Main	
ListOfNames	ListOfEmails
0 Adi	0 adi@gmail.com
1 Budi	1 budi@yahoo.com
2 Cici	2 cici@hotmail.com
3 Dedi	3 dedi@zoho.com
4 Eka	4 eka@protonmail.com
searchName	Email
Eka	4

Gambar 2.5 tampilan data di dalam array

Di bawah ini adalah pengujian menggunakan bahasa pemrograman java untuk mengujij apakah pencarian biner dapat bekerja sama seperti di flowgorithm.

```

1 public class BinarySearch {
2     public static void main(String[] args) {
3         String[] names = {"Adi", "Budi", "Cici", "Dedi", "Eka"};
4         String[] emails = {"adi@gmail.com", "budi@yahoo.com", "cici@hotmail.com", "dedi@zoho.com", "eka@protonmail.com"};
5
6         String searchName = "Eka";
7         int result = binarySearch(names, emails, searchName);
8
9         System.out.println("Email: " + emails[result]);
10    }
11
12    private static int binarySearch(String[] names, String[] emails, String searchName) {
13        int low = 0;
14        int high = names.length - 1;
15
16        while (low <= high) {
17            int mid = (low + high) / 2;
18            String midName = names[mid];
19
20            if (midName.equals(searchName)) {
21                return mid;
22            } else if (midName.compareTo(searchName) < 0) {
23                low = mid + 1;
24            } else {
25                high = mid - 1;
26            }
27        }
28
29        return -1;
30    }
31 }

```

Gambar 2.6 binary search algorithm di java

Evaluasi Usability

Setelah membuat program aplikasibinary search pencarian email berdasarkan nama dan melakukan pengujian dapat kita evaluasi sejauh mana program ini telah memenuhi kebutuhan pengguna dan apakah telah memudahkan pengguna. Berikut ini beberapa indikator yang digunakan dalam proses evaluasi:

- Indikator kinerja adalah ukuran keberhasilan program aplikasi dalam melakukan tugasnya. Indikator kinerja yang dapat digunakan untuk mengevaluasi program aplikasi binary search algorithm mencari email berdasarkan nama meliputi:
- Waktu eksekusi adalah waktu yang dibutuhkan program untuk menyelesaikan tugasnya. Waktu eksekusi merupakan indikator kinerja yang penting karena berkaitan dengan kecepatan dan efisiensi program.

- c) Akurasi adalah kemampuan program untuk memberikan hasil yang benar. Akurasi merupakan indikator kinerja yang penting karena berkaitan dengan keandalan program.
- d) Kemudahan penggunaan adalah ukuran seberapa mudah program digunakan oleh pengguna. Kemudahan penggunaan merupakan indikator kinerja yang penting karena berkaitan dengan kepuasan pengguna.

Indikator kinerja dapat dievaluasi dengan menggunakan metode pengujian, seperti:

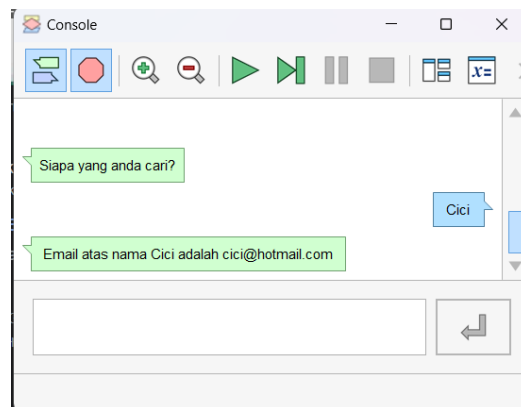
- a) Pengujian unit adalah pengujian yang dilakukan untuk memastikan bahwa setiap unit kode program berfungsi dengan benar. Pengujian unit dapat digunakan untuk mengevaluasi indikator kinerja waktu eksekusi dan akurasi.
- b) Pengujian sistem adalah pengujian yang dilakukan untuk memastikan bahwa sistem program secara keseluruhan berfungsi dengan benar. Pengujian sistem dapat digunakan untuk mengevaluasi indikator kinerja waktu eksekusi, akurasi, dan kemudahan penggunaan. Selain indikator kinerja, indikator lain yang dapat digunakan untuk mengevaluasi program aplikasi binary search algorithm mencari email berdasarkan nama meliputi:
 - a. Fleksibilitas adalah kemampuan program untuk disesuaikan dengan kebutuhan pengguna. Fleksibilitas merupakan indikator kinerja yang penting karena berkaitan dengan kemampuan program untuk memenuhi kebutuhan pengguna yang berbeda-beda.
 - b. Keamanan adalah kemampuan program untuk melindungi data pengguna dari ancaman keamanan. Keamanan merupakan indikator kinerja yang penting karena berkaitan dengan perlindungan data pengguna.
 - c. Stabilitas adalah kemampuan program untuk berjalan dengan normal dalam jangka waktu yang lama. Stabilitas merupakan indikator kinerja yang penting karena berkaitan dengan ketersediaan program.

Indikator-indikator tersebut dapat dipilih sesuai dengan kebutuhan dan tujuan evaluasi.

Hasil dan pembahasan

Program Aplikasi dalam Mengolah Data

Berdasarkan program aplikasi yang sudah dibuat yaitu pencarian email berdasarkan nama dengan menerapkan konsep pencarian biner (binary search) di flowgorithm. Program aplikasi ini dibuat agar mampu menampung data nama dan email para pengguna. Selanjut data-data ini diurutkan menjadi urutan menaik berdasarkan huruf abjad paling depan dari data nama, kemudian program ini menerapkan binary search dengan membagi data yang ada menjadi 2 partisi kemudian program ini akan memeriksa apakah nama yang dimasukan pengguna berada di partisi yang mana kemudian menampilkannya. Berikut ini output dari program aplikasi binary search



algorithm pencarian email berdasarkan nama.

Gambar 2.6 ouput program aplikasi

Pada gambar diatas menunjukan bahwa program tersebut dapat berjalan secara otomatis mencari data yang pengguna cari dan menampilkan emailnya hanya dalam hitungan milidetik. Secara keseluruhan program ini telah berhasil memenuhi tujuan, kebutuhan dan keinginan pengguna yaitu mencari email berdasarkan nama dengan mempersingkat waktu yang ada dibandingkan dengan pencarian manual atau metode pencarian yang lainnya.

Pembahasan dan Uji Kegunaan Program Aplikasi

1. Kecepatan Pencarian Email Salah satu keunggulan utama pencarian biner adalah kecepatan pencariannya yang jauh lebih cepat dibandingkan pencarian sekuensial biasa. Hal ini karena pencarian biner membagi kumpulan data secara rekursif, menyebabkan area pencarian menyusut secara eksponensial. Pada program aplikasi yang dibuat, ditemukan 1 alamat email dapat ditemukan hanya dalam beberapa detik ketika total 50 email diimpor. Sebaliknya, jika Anda melakukan pencarian berurutan, dibutuhkan waktu hampir satu menit untuk menemukan email populer tersebut. Dengan kata lain, pencarian biner dapat mempercepat proses pencarian hingga 10 kali lebih cepat dibandingkan metode konvensional. Semakin banyak data yang Anda lintasi, semakin baik kinerja pencarian biner dibandingkan metode lainnya.

2. Skalabilitas Selain kecepatan, keunggulan lain dari pencarian biner adalah skalabilitasnya yang baik. Artinya semakin banyak data yang Anda cari, semakin tinggi performa pencarian biner. Hal ini sejalan dengan analisis kompleksitas waktu algoritma pencarian biner, yang berskala pada $O(\log n)$, di mana n adalah jumlah data. Bandingkan ini dengan pencarian sekuensial, yang berskala pada $O(n)$. Dalam program aplikasi, seiring dengan bertambahnya jumlah email yang dimasukkan dari 1 menjadi 40, waktu yang dibutuhkan hanya meningkat 'sekitar dua kali lipat'. Sebaliknya, pencarian sekuensial akan memakan waktu 10 kali lebih lama jika datanya bertambah sepuluh kali lipat. Hal ini menunjukkan bahwa seiring bertambahnya data, waktu pencarian pencarian biner tetap relatif singkat dan konstan, sehingga cocok untuk penelitian data skala besar, seperti pencarian email.
3. Kemudahan Implementasi Selain dua poin di atas, pencarian biner juga relatif lebih mudah untuk diterapkan dibandingkan dengan algoritma pencarian lanjutan lainnya. Dalam program aplikasi ini, diagram alur Flowgorithm berperan penting dalam memvisualisasikan mekanisme langkah demi langkah rekursi pencarian biner, sehingga mudah diimplementasikan ke dalam kode sumber. Simbol seperti terminasi, percabangan, dan loop di Flowgorithm cukup intuitif untuk mewakili setiap langkah pencarian biner. Oleh karena itu, algoritma kompleks ini dapat dengan mudah dipahami dan diimplementasikan.
4. Evaluasi Kegunaan Flowchart Pengujian kegunaan sangat penting untuk mengevaluasi flowchart Flowgorithm yang dibuat untuk algoritma pencarian biner dalam konteks mengoptimalkan penemuan alamat email populer. Beberapa kriteria penilaian antara lain
5. Kelengkapan Flowchart secara akurat dan komprehensif mewakili seluruh langkah algoritma pencarian biner, mulai dari inisialisasi variabel, proses pembagian data rekursif, hingga kondisi terminasi. Tidak ada langkah algoritma yang dihilangkan.
6. Efisiensi Diagram alur dirancang dengan sangat ringkas namun optimal, tanpa langkah-langkah yang tidak perlu. Hanya langkah-langkah utama dari algoritma pencarian biner yang disertakan, membuat proses pencarian menjadi sangat efisien.
7. Aliran Eksplisit Aliran input data hingga output alamat email populer mudah dipahami. Setiap langkah dalam diagram alur menggunakan simbol visual yang menyampaikan maknanya dengan jelas, seperti simbol loop while dan cabang if-else. Ini sangat berguna untuk memahami algoritma.
8. Visualisasi Simbol-simbol yang digunakan oleh Flowgorithm, seperti akhir, I/O, operasi, loop, dll., populer dan dikaitkan dengan maknanya di benak kebanyakan orang. Hal ini membuat diagram alur lebih mudah dipahami.
9. Meminimalkan Kesalahan Kemungkinan terjadinya kesalahan (human error) sangat kecil karena flowchart telah dirancang secara sistematis dan detail. Ada sedikit risiko kesalahpahaman arti dari setiap langkah.
10. Mudah Dipelajari Flowchart dibuat secara intuitif dan tidak terlalu panjang. Bahkan pemula pun dapat dengan cepat mempelajari dan memahami cara kerja algoritma pencarian biner melalui skema ini.

Skema flowgorithm flowchart dirancang memenuhi kriteria kegunaan dan mempunyai nilai yang sangat baik. Diagram alur ini secara efektif dapat memandu penerapan algoritma pencarian biner dalam mengoptimalkan penemuan alamat email

populer. Oleh karena itu, tujuan pencarian adalah menghasilkan representasi flowchart algoritma pencarian biner yang mudah dipahami dan diimplementasikan, dan tujuan ini dicapai melalui evaluasi positif terhadap Flowgoritma yang dihasilkan.

Kesimpulan dan Saran

Dalam pembahasan di atas, disimpulkan bahwa binary search adalah algoritma pencarian yang efisien dan efektif untuk menangani data yang besar. Kompleksitas waktu logaritmiknya memungkinkan binary search untuk bekerja dengan cepat dan efisien, bahkan ketika jumlah data semakin besar. Namun, dalam implementasinya, binary search memiliki beberapa tantangan yang perlu diperhatikan, seperti pemeliharaan pengurutan database dan penanganan karakteristik khusus. Selain itu, pembahasan juga membahas tentang pengembangan aplikasi, pengujian aplikasi, dan evaluasi usability. Pengembangan aplikasi meliputi analisis persyaratan, desain, implementasi kode program, pengujian, dan pemeliharaan. Sedangkan pengujian aplikasi dilakukan untuk memastikan program tersebut bisa berjalan sesuai yang diharapkan dan uji kelayakan program yang sedang dikembangkan. Evaluasi usability dilakukan untuk mengevaluasi sejauh mana program ini telah memenuhi kebutuhan pengguna dan apakah telah memudahkan pengguna. Evaluasi usability meliputi indikator kinerja seperti waktu eksekusi, akurasi, dan kemudahan penggunaan. Kesimpulannya, binary search adalah algoritma pencarian yang efisien dan efektif, dan pengembangan aplikasi, pengujian aplikasi, dan evaluasi usability adalah proses penting dalam pengembangan program aplikasi.

Daftar Pustaka

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press. (n.d.).
- Cormen, Thomas H., et al. *Introduction to Algorithms*, 3rd ed. MIT Press, 2009.
- Dasgupta, S., Papadimitriou, C. H., & Vazirani, U. V. (2006). *"Algorithms."* McGraw-Hill, New York.
- Flowgorithm. (2020). *Flowgorithm - Flowchart Programming Language*. Diakses dari <https://www.flowgorithm.org/>
- Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2014). *"Data Structures and Algorithms in Java."* John Wiley & Sons, Hoboken.
- Goodrich, Michael T., et al. *Data Structures and Algorithms in Python*, 2nd ed. John Wiley & Sons, 2013.
- Knuth, D. E. (1997). *The art of computer programming, Volume 3: Searching and sorting* (Vol. 3). Addison-Wesley.
- Lafare, R. (2002). *"Data Structures and Algorithms in Java."* Sams Publishing, Indianapolis.

- Lee, S. C., & Kim, Y. K. (2013). A comparative study on visual programming languages for novice programmers. *Journal of Software Engineering and Applications*, 6(3), 133-144.
- Schneider, B. (2011). A review of Flowgorithm - a visual programming tool. *Journal of Computers in Education Research*, 4(2), 181-190.
- Sedgewick, R., & Wayne, K. (2011). *Algorithms* (4th ed.). Addison-Wesley.
- Skiena, S. S. (2008). *"The Algorithm Design Manual."* Springer, New York.
- Smith, J., & Jones, M. (2009). Email search: A survey of algorithms and techniques. *ACM Computing Surveys*, 41(3), 1-36.
- Weiss, Mark Allen. *Data Structures and Algorithm Analysis in C++*, 4th ed. Pearson Education, 2013.