

Aplikasi Algoritma Decrease and Conquer dalam Pencarian Nilai Z pada Tabel Normal Standar

Rifqi Rifaldi Utomo - 13516098

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung 40132, Indonesia

rrifaldu@students.itb.ac.id

Abstrak—Makalah ini membahas mengenai cara untuk melakukan pencarian nilai z pada tabel normal standar menggunakan algoritma *decrease and conquer*. Terdapat dua pendekatan yang digunakan, yakni pencarian biner dan pencarian interpolasi. Makalah ini juga membandingkan kedua pendekatan tersebut dalam aplikasi pencarian nilai z.

Kata kunci—*decrease and conquer*; *pencarian interpolasi*; *pencarian biner*; *statistika*

I. PENDAHULUAN

Statistika merupakan salah satu cabang ilmu dari matematika. Menurut KBBI, statistika merupakan ilmu tentang cara mengumpulkan, menabulasi, menggolong-golongkan, menganalisis, dan mencari keterangan yang berarti dari data yang berupa angka [1].

Terdapat banyak sekali kegunaan ilmu statistika. Salah satu kegunaannya adalah melakukan prediksi. Dalam beberapa kasus prediksi, kita perlu mengetahui nilai-z dengan persentase luas kurva yang sudah diketahui. Untuk mengetahui nilai-z tersebut, kita merujuk pada tabel distribusi normal standar. Agar kita mendapat nilai yang lebih akurat, kita akan merujuk pada tabel dengan elemen yang lebih banyak. Akan tetapi, hal tersebut juga menyulitkan kita untuk mencari nilai-z. Hal itu disebabkan kita seringkali mencari nilai-z pada tabel secara acak. Pencarian itu tentu saja tidak efisien.

Z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9725	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9924	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.6	0.9953	0.9955	0.9956	0.9957	0.9958	0.9960	0.9961	0.9962	0.9963	0.9964
2.7	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.8	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
2.9	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986

Gambar 1. Tabel distribusi normal standar

Sumber: <https://statistics.laerd.com/statistical-guides/normal-distribution-calculations.php>

(akses: 12 Mei 2018)

Oleh karena itu, kita perlu melakukan pencarian yang lebih efisien sehingga tidak banyak waktu yang digunakan untuk mencari nilai-z. Terdapat banyak metode pencarian yang dapat digunakan, seperti pencarian biner dan pencarian interpolasi.

Pada makalah ini, akan dibahas pencarian nilai-z pada tabel normal standar menggunakan pencarian interpolasi dan pencarian biner.

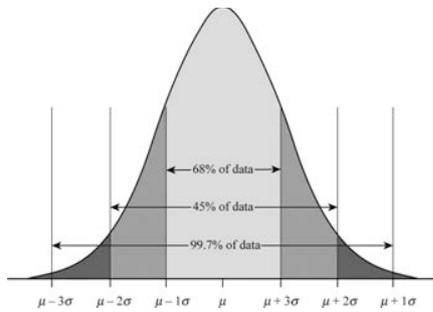
II. LANDASAN TEORI

A. Distribusi Normal

Distribusi normal adalah distribusi probabilitas kontinu yang paling penting dalam seluruh bidang statistika. Bentuk graf dari distribusi normal adalah kurva normal, yakni sebuah kurva yang berbentuk lonceng. Distribusi normal menjelaskan banyak fenomena yang terjadi pada alam, industri, maupun penelitian. Sebagai contoh, pengukuran fisis pada bidang meteorologi, penelitian hujan, dan pengukuran produksi seringkali dijelaskan menggunakan distribusi normal. Selain itu, kesalahan pada pengukuran ilmiah sangat baik dijelaskan dengan distribusi normal. Pada tahun 1733, Abraham DeMoivre mengembangkan persamaan matematika untuk kurva normal. Distribusi normal sering juga disebut sebagai distribusi Gauss, merujuk kepada Karl Friedrich Gauss yang juga menurunkan persamaan distribusi normal melalui penelitian kesalahan pada pengukuran berulang dengan kuantitas yang sama [2].

B. Distribusi Normal Standar

Distribusi normal standar merupakan distribusi normal yang memiliki rata-rata senilai 0 dan simpangan baku senilai 1 [2]. Luas di bawah kurva distribusi normal standar dapat ditemukan dengan menggunakan tabel distribusi normal standar. Tidak ada variabel yang diukur secara alami memiliki distribusi ini. Akan tetapi, semua distribusi normal lainnya setara dengan distribusi ini apabila unit pengukurannya diubah menjadi pengukuran standar deviasi dari rata-rata. Pada makalah ini, akan digunakan variabel Z untuk melambangkan distribusi normal standar [3].



Gambar 2. Kurva normal

Sumber: <https://statistics.laerd.com/statistical-guides/normal-distribution-calculations.php> (akses: 12 Mei 2018)

C. Tabel Normal Standar

Tabel normal standar yang biasa disebut tabel Z merupakan tabel matematika untuk nilai Φ , yang merupakan nilai dari fungsi distribusi kumulatif dari distribusi normal standar. Tabel ini digunakan untuk mencari probabilitas statistika yang diamati di bawah, di atas, atau di antara nilai yang terdapat pada distribusi normal standar. Karena terdapat berbagai variasi distribusi normal yang jumlahnya tidak berhingga, maka tidak mungkin mencetak tabel probabilitas untuk setiap distribusi normal. Oleh karena itu, biasanya orang-orang akan mengubah distribusi normal menjadi distribusi normal standar kemudian menggunakan tabel normal standar untuk menemukan peluang [4].

D. Algoritma Decrease and Conquer

Decrease and conquer merupakan metode perancangan algoritma yang bekerja dengan mereduksi suatu persoalan menjadi beberapa sub-persoalan yang lebih kecil, kemudian memproses satu sub-persoalan

E. Algoritma Decrease and Conquer

Decrease and conquer merupakan metode perancangan algoritma yang bekerja dengan mereduksi suatu persoalan menjadi beberapa sub-persoalan yang lebih kecil, kemudian memproses satu sub-persoalan yang mengarah ke solusi. Hal ini berbeda dengan algoritma *divide and conquer* yang memproses setiap sub-persoalan yang telah dipecah kemudian menggabungkannya. *Decrease and conquer* terdiri dari dua tahap, yakni *decrease* dan *conquer*. Tahap *decrease* mereduksi persoalan menjadi beberapa persoalan yang lebih kecil. Tahap *conquer* adalah memproses satu sub-persoalan secara rekursif [5]. *Decrease and conquer* memiliki tiga variasi, yakni:

1. Decrease by a constant

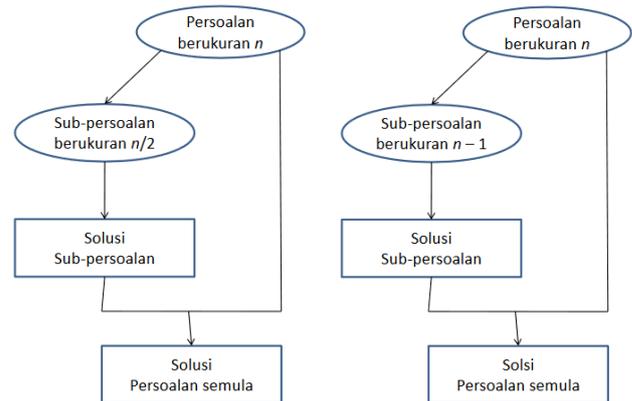
Pada variasi ini, persoalan direduksi sebesar konstanta yang sama dalam setiap iterasi algoritma. Aplikasi dari variasi ini antara lain persoalan perpangkatan a^n dan *insertion sort*.

2. Decrease by a constant factor

Pada variasi ini, ukuran instans persoalan direduksi sebesar faktor konstanta yang sama dalam setiap iterasi algoritma. Aplikasi dari variasi ini antara lain pencarian biner, persoalan koin palsu, dan perkalian petani Rusia.

3. Decrease by a variable size

Pada variasi ini, ukuran instans persoalan direduksi dengan besar yang bervariasi pada setiap iterasi algoritma. Aplikasi dari variasi ini antara lain menghitung median, pencarian interpolasi, dan algoritma Euclid.



(a)

(b)

Gambar 3. Diagram algoritma *decrease and conquer* (a) *decrease by a constant*, (b) *decrease by a constant factor*

Sumber: [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Decrease-and-Conquer-\(2018\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Decrease-and-Conquer-(2018).pdf) (akses: 12 Mei 2018)

F. Pencarian Biner

Pencarian biner merupakan metode pencarian sebuah elemen pada larik terurut. Pencarian biner mencari elemen dengan membagi-bagi elemen menjadi dua. Pada awalnya, interval pencarian mencakup seluruh isi larik. Jika nilai yang dicari lebih kecil dari elemen yang ada di tengah interval, maka interval dipersempit sehingga hanya melingkupi setengah elemen terendah. Sebaliknya, jika nilai yang dicari lebih besar dari elemen yang ada di tengah interval, maka interval dipersempit sehingga hanya melingkupi setengah elemen tertinggi. Pencarian diulangi sampai nilai ditemukan atau interval kosong [6].

Adapun langkah yang lebih rinci dari pencarian biner adalah sebagai berikut:

1. Bandingkan nilai yang dicari dengan elemen pada tengah interval.
2. Apabila nilai pada indeks yang diperiksa sama dengan elemen pada tengah interval, kembalikan indeks dan keluar dari iterasi
3. Apabila nilai yang dicari kurang dari nilai yang diperiksa, maka periksa larik bagian kiri, jika sebaliknya, maka periksa larik bagian kanan.

Ulangi iterasi sampai indeks ditemukan atau sub-larik tereduksi hingga tidak memiliki elemen.

G. Pencarian Interpolasi

Pencarian interpolasi merupakan metode pencarian sebuah elemen pada larik terurut. Pencarian interpolasi merupakan perbaikan dari pencarian biner. Jika pencarian biner selalu mencari elemen paling tengah untuk diperiksa, pencarian

interpolasi dapat memeriksa ke lokasi yang berbeda tergantung pada nilai yang sedang dicari. Sebagai contoh, jika nilai yang dicari lebih dekat ke elemen terakhir, maka pencarian interpolasi akan memulai pencariannya dari bagian akhir larik [7].

Langkah dari pencarian interpolasi adalah sebagai berikut:

1. Kalkulasi indeks dari posisi yang akan diperiksa menggunakan rumus berikut:

$$idx = idx_{low} + (idx_{upper} - idx_{low}) \times (K - K_{low}) / (K_{upper} - K_{low})$$
 dengan K adalah nilai yang dicari.
2. Apabila nilai pada indeks yang diperiksa sama dengan nilai yang dicari, kembalikan indeks dan keluar dari iterasi
3. Apabila nilai yang dicari kurang dari nilai yang diperiksa, maka periksa larik bagian kiri, jika sebaliknya, maka periksa larik bagian kanan
4. Ulangi iterasi sampai indeks ditemukan atau sub-larik tereduksi hingga tidak memiliki elemen.

III. PENERAPAN ALGORITMA PADA PERSOALAN

A. Cara Kerja Algoritma

Cara kerja dari algoritma pencarian biner adalah melakukan pemeriksaan pada elemen tengah interval, kemudian bila belum ditemukan, akan dicari pada setengah interval yang mengarah ke solusi. Hal tersebut dapat dijelaskan melalui contoh soal di bawah ini:

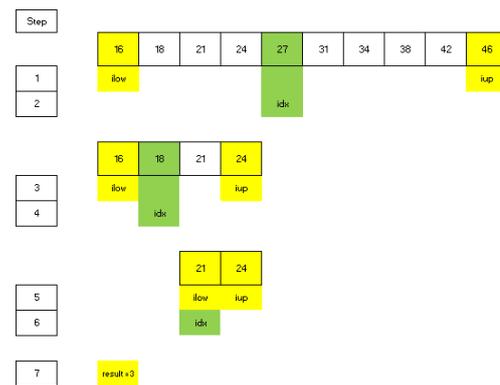
Terdapat larik berisi 10 elemen mengurut dengan isi 16, 18, 21, 24, 27, 31, 34, 38, 42, 46. Kemudian kita mencari indeks dengan elemen 21, maka pencarian dengan menggunakan pencarian biner adalah sebagai berikut:

1. Inisialisasi nilai i_{low} dengan indeks pertama pada larik yakni 1 dan i_{up} dengan indeks terakhir pada larik yakni 10.
2. Periksa elemen pada tengah interval, yakni dengan indeks 5. Ternyata, elemen tersebut berisi 27, yang lebih besar dari 21.
3. Karena masih lebih besar, maka ubah nilai i_{up} menjadi indeks tengah dikurangi 1.
4. Periksa kembali elemen pada tengah interval, yakni elemen dengan indeks 2. Ternyata, elemen tersebut berisi 18, yang lebih kecil dari 21.
5. Karena masih lebih kecil, maka ubah nilai i_{low} menjadi indeks tengah ditambah 1.
6. Periksa kembali elemen pada tengah interval, yakni elemen dengan indeks 3. Ternyata, elemen tersebut berisi 21, sesuai dengan nilai yang kita cari.
7. Kembalikan nilai 3 ke program utama

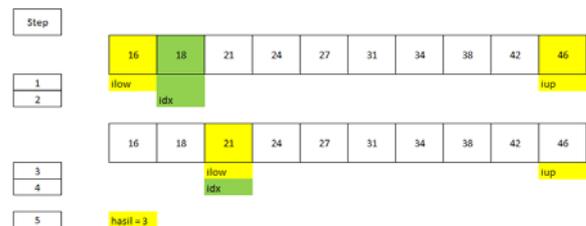
Adapun cara kerja dari algoritma pencarian interpolasi adalah melakukan interpolasi dengan menggunakan nilai bawah, nilai atas, dan nilai yang dicari dari program, kemudian membagi dua larik berdasarkan hasil interpolasi dan mengecek salah satu larik yang mengarah pada solusi. Dengan

menggunakan permasalahan yang sama dengan pencarian biner, berikut adalah alur dari pencarian interpolasi:

1. Inisialisasi nilai i_{low} dengan indeks pertama pada larik yakni 1 dan i_{up} dengan indeks terakhir pada larik yakni 10.
2. Hitung nilai interpolasi. Dari perhitungan tersebut, didapatkan nilai 2. Selanjutnya, periksa apakah nilai pada indeks 2 sesuai dengan nilai yang kita cari. Ternyata nilainya masih lebih kecil.
3. Karena masih lebih kecil, maka ubah nilai i_{low} menjadi nilai interpolasi ditambah 1, yakni 3.
4. Hitung kembali nilai interpolasi. Dari perhitungan tersebut, didapatkan nilai 3. Selanjutnya, periksa apakah nilai pada indeks 3 sesuai dengan nilai yang kita cari. Ternyata nilainya sudah sesuai, yakni 21.
5. Kembalikan nilai 3 ke program utama



(a)



(b)

Gambar 4. Ilustrasi pencarian dengan hasil ditemukan (a) pencarian biner (b) pencarian interpolasi

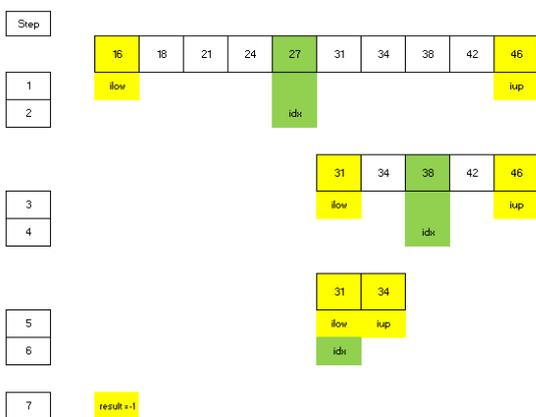
Dengan larik yang sama, kita dapat menguji kasus pada saat elemen tidak berada di dalam larik, misalnya kita mencari bilangan 28. Untuk pencarian biner, langkah-langkahnya adalah sebagai berikut:

1. Inisialisasi nilai i_{low} dengan indeks pertama pada larik yakni 1 dan i_{up} dengan indeks terakhir pada larik yakni 10.
2. Periksa elemen pada tengah interval, yakni dengan indeks 5. Ternyata, elemen tersebut berisi 27, yang lebih kecil dari 28.

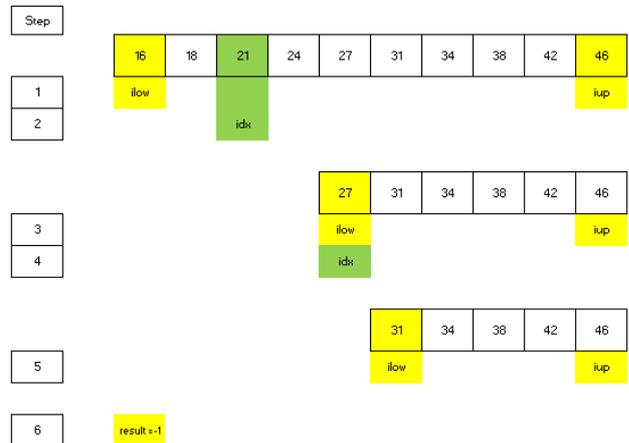
3. Karena masih lebih kecil, maka ubah nilai i_{low} menjadi indeks tengah ditambah 1.
4. Periksa kembali elemen pada tengah interval, yakni elemen dengan indeks 8. Ternyata, elemen tersebut berisi 38, yang lebih besar dari 28.
5. Karena masih lebih besar, maka ubah nilai i_{up} menjadi indeks tengah dikurang 1.
6. Periksa kembali elemen pada tengah interval, yakni elemen dengan indeks 6. Ternyata, elemen tersebut berisi 31, lebih kecil dari nilai yang kita inginkan. Akan tetapi, nilai tersebut sudah berada indeks paling kiri. Oleh karena itu, dapat disimpulkan bahwa elemen 28 tidak ada pada larik.
7. Kembalikan nilai -1 ke program utama

Adapun pada pencarian interpolasi, langkahnya adalah sebagai berikut:

1. Inisialisasi nilai i_{low} dengan indeks pertama pada larik yakni 1 dan i_{up} dengan indeks terakhir pada larik yakni 10.
2. Hitung nilai interpolasi. Dari perhitungan tersebut, didapatkan nilai 3. Selanjutnya, periksa apakah nilai pada indeks 3 sesuai dengan nilai yang kita cari. Ternyata nilainya masih lebih kecil.
3. Karena masih lebih kecil, maka ubah nilai i_{low} menjadi nilai interpolasi ditambah 1, yakni 4.
4. Hitung kembali nilai interpolasi. Dari perhitungan tersebut, didapatkan nilai 4. Selanjutnya, periksa apakah nilai pada indeks 4 sesuai dengan nilai yang kita cari. Ternyata nilainya masih lebih kecil.
5. Karena masih lebih kecil, maka ubah nilai i_{low} menjadi nilai interpolasi ditambah 1, yakni 5. Akan tetapi, nilai indeks 5 lebih besar dari nilai yang kita cari. Sehingga kita mengambil kesimpulan bahwa nilai yang kita cari tidak terdapat pada larik.
6. Kembalikan nilai -1 ke program utama



(a)



(b)

Gambar 5. Ilustrasi pencarian dengan hasil tidak ditemukan (a) pencarian biner (b) pencarian interpolasi

B. Pseudocode Algoritma

Pseudocode dari algoritma pencarian biner dalam kasus pada makalah ini sama seperti pencarian biner lainnya, yakni sebagai berikut:

```

function pencarian_biner (input a: ArrayOfReal;
    input ilow, iup : integer, k: real) →
    integer

Deklarasi
    mid: integer

Algoritma
    while (iup >= ilow) do
        mid ← (ilow + (iup - ilow)/2)
        if (amid = k) then
            {Nilai ditemukan}
            → mid
        else if (amid > k) then
            ilow = mid + 1
        else
            iup = mid - 1
        endif
    endwhile
    {Tidak ditemukan}
    → -1

```

Adapun pseudocode dari algoritma pencarian interpolasi dalam makalah ini juga tidak jauh berbeda dengan algoritma interpolasi yang sering ditemukan, yakni sebagai berikut:

```

function pencarian_interpolasi (input a:
    ArrayOfReal; input n : integer, K: real) →
    integer

Deklarasi
    ilow, iup: integer

Algoritma
    while(ilow <= iup and k >= ailow and k <= aiup)do
        i ← ilow + (iup-ilow)*(K-ailow)/(aup-ailow)
        if ai = k then {K ditemukan}
            → i          {kembalikan indeks}
        else
            if (ai < k) then
                ilow ← i + 1
            else
                iup ← i + 1
            endif
        endif
    endwhile
    → -1 {Tidak ditemukan}

```

Adapun program utama berfungsi untuk memperoleh data tabel normal standar dari file eksternal, menerima masukan K dari pengguna, memanggil fungsi pencarian biner atau interpolasi, dan mengeluarkan hasilnya. Berikut adalah pseudocode dari program utama:

```

Deklarasi
    aluas, anilaiz: ArrOfReal
    n, idx : integer
    k, hasil : real

Algoritma
    file_input(aluas)
    file_input(anilaiz)
    input(K)
    n = aluas.length
    {Untuk pencarian biner}
    idx1 ← pencarian_interpolasi(aluas, 1, n, k)
    {Untuk pencarian interpolasi}
    idx2 ← pencarian_interpolasi(aluas, n, k)
    if idx1 ≠ -1 then
        output(anilaizidx1)
    else
        output("Nilai tidak ditemukan")
    if idx2 ≠ -1 then
        output(anilaizidx2)
    else
        output("Nilai tidak ditemukan")

```

IV. HASIL IMPLEMENTASI ALGORITMA

Sebelum melakukan pengujian, perlu diketahui bahwa data tabel normal standar yang digunakan pada makalah ini merujuk pada tabel yang tertera pada buku *Probability and Statistics for Engineers and Scientists* karya Ronald E. Walpole, dkk. yakni tabel dengan range -3.49 hingga 3.49 dengan pembulatan 4 angka di belakang koma.

A. Pencarian Nilai Acak pada Tabel

Pencarian ini dilakukan dengan memilih satu nilai acak pada tabel kemudian melakukan pencarian biner dan pencarian interpolasi. Adapun nilai yang dipilih adalah 0.0068 dan 0.9871. Hasilnya dapat dilihat pada gambar 6.

```

D:\Rifqi Rifaldi Utomo\!RI
Masukkan nilai k: 0.0068
=Pencarian Biner=
Jumlah iterasi = 9
Nilai z = -2.47
=Pencarian Interpolasi=
Jumlah iterasi = 34
Nilai z = -2.47

D:\Rifqi Rifaldi Utomo\!RI
Masukkan nilai k: 0.9871
=Pencarian Biner=
Jumlah iterasi = 8
Nilai z = 2.23
=Pencarian Interpolasi=
Jumlah iterasi = 19
Nilai z = 2.23

```

Gambar 6. Hasil pencarian nilai acak pada tabel

B. Pencarian Nilai Tengah dan Ujung pada Tabel

Pencarian ini dilakukan dengan memilih nilai tengah pada tabel yakni 0.5 dan nilai ujung pada tabel yakni 0.0002 dan 0.9998. Hasilnya dapat dilihat pada gambar 7.

<pre> Masukkan nilai k: 0.9998 =Pencarian Biner= Jumlah iterasi = 10 Nilai z = 3.49 =Pencarian Interpolasi= Jumlah iterasi = 1 Nilai z = 3.49 </pre>	<pre> Masukkan nilai k: 0.0002 =Pencarian Biner= Jumlah iterasi = 9 Nilai z = -3.49 =Pencarian Interpolasi= Jumlah iterasi = 1 Nilai z = -3.49 </pre>
(a)	(b)
<pre> Masukkan nilai k: 0.5 =Pencarian Biner= Jumlah iterasi = 1 Nilai z = 0.0 =Pencarian Interpolasi= Jumlah iterasi = 1 Nilai z = 0.0 </pre>	
(c)	

Gambar 7. Hasil pencarian nilai tengah dan ujung pada tabel

C. Pencarian Nilai yang Memiliki Lebih Dari Satu Kemunculan pada Tabel

Pencarian ini dilakukan dengan memilih nilai pada tabel dengan kemunculan lebih dari satu. Pada kasus ini, penulis memilih nilai 0.0003 yang terdapat pada nilai z -3.39 hingga -3.48. Hasilnya dapat dilihat pada gambar 8.

```

Masukkan nilai k: 0.0003
=Pencarian Biner=
Jumlah iterasi = 6
Nilai z = -3.39
=Pencarian Interpolasi=
Jumlah iterasi = 2
Nilai z = -3.48

```

Gambar 8. Hasil pencarian nilai yang memiliki lebih dari satu kemunculan pada tabel

V. ANALISIS DAN KESIMPULAN

Pada pencarian nilai z, terdapat 2 pendekatan yang dilakukan, yakni pencarian biner dan pencarian interpolasi. Pencarian biner memiliki kompleksitas $O(\log N)$, sedangkan pencarian interpolasi memiliki kompleksitas yang bervariasi, dengan kemungkinan terburuk $O(N)$ dan kemungkinan terbaik $O(\log \log N)$

Dalam kasus pencarian nilai-z, metode pencarian biner memiliki kompleksitas yang relatif stabil, dengan jumlah iterasi berkisar antara 6 hingga 10. Adapun metode pencarian interpolasi memiliki kompleksitas yang bervariasi, mulai dari 1 iterasi hingga 34 iterasi. Hal ini membuktikan bahwa kompleksitas pencarian interpolasi memang beragam. Pencarian interpolasi memiliki jumlah iterasi yang lebih sedikit dibandingkan pencarian biner hanya pada kasus-kasus tertentu. Akan tetapi, secara keseluruhan, pencarian interpolasi memiliki kinerja yang buruk dalam pencarian nilai z. Hal ini disebabkan distribusi nilai yang terdapat pada tabel normal standar tidaklah seragam, tetapi normal. Oleh karena itu, dapat disimpulkan bahwa dalam melakukan pencarian nilai z, sebaiknya menggunakan pencarian biner.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Allah SWT, karena berkat izin-Nya saya dapat menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada kedua orang tua penulis yang telah mengasuh dan mendidik penulis sejak masih dalam kandungan. Tak luput penulis ungkapkan terima kasih kepada Ibu Dr. Masayu Leylia Khodra, S.T., M.T., Ibu Dr. Nur Ulfa Maulidevi, S.T., M.Sc., dan Bapak Dr. Ir. Rinaldi Munir, M.T selaku dosen pengajar mata kuliah IF 2211 Strategi Algoritma yang telah memberikan ilmunya kepada penulis. Penulis juga mengucapkan terima kasih kepada pihak-pihak yang telah membantu penulis dalam membuat makalah ini.

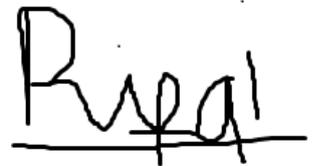
REFERENSI

- [1] Hasil Pencarian – KBBI Daring, <https://kbbi.kemdikbud.go.id/entri/statistika>, diakses pada tanggal 12 Mei 2018, pukul 10.03
- [2] Walpole, Ronald E. dkk. 2016. "Probability and Statistics for Engineers and Scientists". Amerika Serikat: Pearson Higher Ed USA
- [3] Standard Normal Distribution. <http://www.oswego.edu/~srp/stats/z.htm>, diakses pada tanggal 12 Mei 2018 pukul 14.42
- [4] Larson, Ron; Farber, Elizabeth. 2004. "Elementary Statistics: Picturing the World". Amerika Serikat: Pearson Higher Ed USA
- [5] Munir, Rinaldi. "Diktat Kuliah IF2211 Strategi Algoritma," Program Studi Teknik Informatika Institut Teknologi Bandung, 2009, Bandung, Indonesia
- [6] Binary Search, <https://www.geeksforgeeks.org/binary-search/>, diakses pada tanggal 12 Mei 2018, pukul 17.30
- [7] Interpolation Search, <https://www.geeksforgeeks.org/interpolation-search/>, diakses pada tanggal 13 Mei 2018, pukul 20.04

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 14 Mei 2018



Rifqi Rifaldi Utomo - 1316098