besar. Algoritma yang lebih cepat daripada algoritma pencarian beruntun adalah algoritma bagidua, yang diberikan di bawah ini.

## Pencarian Bagidua

114

sekumpulan data yang sudah terurut (terurut menaik atau terurut menurun). Metode ini makan untuk kebutuhan pencarian dengan waktu yang cepat. Data yang terurut adalah syarat penerapan algoritma pencarian bagidua. Memang, syarat ini pula yang merupakan kelemahan ma pencarian bagidua, berbeda pada pencarian beruntun yang dapat diterapkan baik pada data maupun pada data yang tidak terurut. Namun, sebenaranya data yang sudah terurut banyak dalam kehidupan sehari-hari. Data nomor telepon di buku telepon misalnya, sudah berdasarkan nama/instansi pelanggan telepon dari A sampai Z. Data karyawan diurut sarkan nomor induknya dari nomor kecik ke nomor besar. Data mahasiswa diurut berdasarkan Nomor Induk Mahasiswa), kata-kata (entry) di dalam kamus bahasa Inggris/Indonesia telah dari A sampai Z, dan sebagainya.

satu keuntungan data yang terurut adalah memudahkan pencarian, yang dalam hal ini pencarian Sebenarnya, dalam kehidupan sehari-hari kita sering menerapkan pencarian bagidua. Untuk arti kata tertentu di dalam kamus (misalnya kamus Bahasa Inggris), kita tidak membuka itu dari halaman awal sampai halaman akhir satu per satu, namun kita mencarinya dengan cara belah atau membagi dua buku itu. Jika kata yang dicari tidak terletak di halaman pertengahan itu, mencari lagi di belahan bagian kiri atau belahan bagian kanan dengan cara membagi dua belahan dimaksud. Begitu seterusnya sampai kata yang dicari ditemukan. Hal ini hanya bisa dilakukan maa-kata di dalam kamus sudah terurut.

pencarian dengan membagi data atas dua bagian mengilhami algoritma pencarian bagidua.

yang disimpan di dalam larik harus sudah terurut. Untuk memudahkan pembahasan, misalkan larik sudah terurut menurun. Selama proses pencarian, kita memerlukan dua buah indeks larik, indeks terkecil dan indeks terbesar. Kita menyebut indeks terkecil sebagai indek ujung kiri larik indeks terbesar sebagai indeks ujung kanan larik. Istilah "kiri" dan "kanan" dinyatakan dengan bayangkan elemen larik terentang horizontal (lihat Gambar 1.1(a)).

The standard indeks kiri adalah Ia dan indek kanan adalah Ib. Pada mulanya, Ia = 1 dan Ib = N.

Bagi dua elemen larik pada elemen tengah. Elemen tengah adalah elemen dengan indeks k = (Ia + Ib) div 2

(Elemen tengah, L[k], membagi larik menjadi dua bagian, yaitu bagian kiri L[Ia..k-1] dan bagian kanan L[k+1..Ib])

Langkah 2: Periksa apakah L[k] = X. Jika L[k] = X, pencarian dihentikan sebab X si ditemukan. Tetapi, jika L[k] ≠ X, harus ditentukan apakah pencarian akan dilakukan larik bagian kiri atau di bagian kanan. Jika L[k] < X, maka pencarian dilakukan larik bagian kiri. Sebaliknya, jika L[k] > X, pencarian dilakukan pada larik bakanan.

Langkah 3: Ulangi Langkah 1 sampai X ditemukan atau Ia > Ib (yaitu, ukuran larik sudah nol

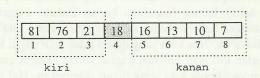
Contoh ilustrasi pencarian bagidua:

Misalkan diberikan larik L dengan delapan buah elemen yang sudah temmenurun seperti di bawah ini:

81	76	21	18	16	13	10	7
Ia=1	2	3	4	5	6	7	8=Ib

(i) Misalkan elemen yang dicari adalah X = 18.

Langkah 1: Ia = 1 dan Ib = 8 Indeks elemen tengah k = (1 + 8) DIV 2 = 4 (diarsir)



Langkah 2: L[4] = 18? Ya! (X ditemukan, pencarian dihentikan)

(ii) Misalkan elemen yang dicari adalah X = 16.

Langkah 1: Ia = 1 dan Ib = 8 Indeks elemen tengah k = (1 + 8) DIV 2 = 4 (diarsir)



Langkah 2:

L[4] = 16? Tidak!

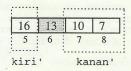
Harus diputuskan apakah pencarian akan dilakukan di bagian kiri abagian kanan dengan pemeriksaan sbb:

L[4] > 16? Ya! Lakukan pencarian pada larik bagian kanan dengan Ia = k + 1 = 5 dan Ib = 8 (tetap)

16	13	10	7
a=5	6	7	8=Ib

Langkah 1': La = 5 dan Ib = 8

Example 2 = 6 (diarsir)



Lengkah 2':
L[6] = 16? Tidak!

Barus diputuskan apakah pencarian akan dilakukan di bagian kiri atau di tagian kanan dengan pemeriksaan sbb:

L[6] > 16? Tidak! Lakukan pencarian pada larik bagian kiri dengan Ia = 5 (tetap) dan Ib = k - 1 = 5



Langkah 1":

Ia = 5 dan Ib = 5

Indeks elemen tengah k = (5 + 5) DIV 2 = 5 (diarsir)



Langkah 2\*:

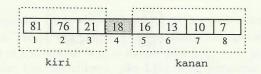
= 16? Ya! (X ditemukan, pencarian dihentikan)

Misalkan elemen yang dicari adalah X = 100

Langkah 1:

Dm = 1 dan Ib = 8

Indexs elemen tengah k = (1 + 8) DIV 2 = 4 (diarsir)



Langkah 2: L[4] = 100? Tidak!

Harus diputuskan apakah pencarian akan dilakukan di bagian kiri ata bagian kanan dengan pemeriksaan sbb:

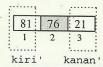
L[4] > 100? Tidak! Lakukan pencarian pada larik bagian kiri dengan Ia = 1 (tetap) dan Ib = k - 1 = 3



Langkah 1':

Ia = 1 dan Ib = 3

Indeks elemen tengah k = (1 + 3) DIV 2 = 2 (diarsir)



Langkah 2':

L[2] =: 100? Tidak!

Harus diputuskan apakah pencarian akan dilakukan di bagian kiri atau bagian kanan dengan pemeriksaan sbb:

L[2] > 100? Tidak! Lakukan pencarian pada larik bagian kiri dengan Ia = 1 dan Ib = k - 1 = 1



Langkah 1":

Ia = 1 dan Ib = 1 Indeks elemen tengah k = (1 + 1) DIV 2 = 1 (diarsir)

```
Ia = 1 dan Ib = k - 1 = 0
                     Karena Ia > Ib, maka tidak ada lagi bagian larik yang tersisa. Dengan
                     demikian, X tidak ditemukan di dalam larik. Pencarian dihentikan.
______ BagiDual(input L : Larik100, input N : integer, input X:integer,
input/output IX: integer)

input/output IX: integer)

input/output IX: integer)

input/output IX: integer)

input x: integer, input x: int
arga 0 jika X tidak ditemukan.
: Larik L[1..N] sudah berisi data yang sudah terurut menurun, dan X
                      adalah harga yang akan dicari
er: IX berisi indeks larik tempat X ditemukan, IX = 0 jika X tidak
                      ditemukan }
LOKAL
   integer : integer
    = integer
                                                                    { indeks elemen tengah}
    boolean
                                                                    {flag untuk menentukan ketemu atau tidak}
   AMTEMA
     Dm ← 1
     IIm ← N.
    ← false
     (not ketemu) and (Ia ≤ Ib) do

(Ia + Ib) div 2 { bagidua larik L pada posisi k }
           \underline{\underline{I}} (L[k] = X) then
                ketemu ← true
           ( akan lakukan pencarian pada larik bagian kanan, set indeks
                            ujung kiri larik yang baru }
                           Ia \leftarrow k + 1
                   else
                         { akan lakukan pencarian pada larik bagian kiri, set indeks
                             ujung kanan larik yang baru }
                           \texttt{Ib} \leftarrow \texttt{k-1}
                   endif
                dif
         = <u>true or</u> Ia > Ib}

then { X ditemukan }
                     { X tidak ditemukan di dalam larik }
```

Harus diputuskan apakah pencarian akan dilakukan di bagian kiri atau di

L[1] > 100? Tidak! Lakukan pencarian pada larik bagian kiri dengan

Langkah 2":

L[1] = 100? Tidak!

bagian kanan dengan pemeriksaan sbb: