

MODUL MATA KULIAH

ANALISIS DAN DESAIN ALGORITMA

PG167 – 3 SKS



**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
SEPTEMBER 2019**

TIM PENYUSUN

Dr. Achmad Solichin, S.Kom., M.T.I
Atik Ariesta, S.Kom., M.Kom
Ita Novita, S.Kom., M.T.I



MODUL PERKULIAHAN #13

PENGGABUNGAN (MERGE) ARRAY SATU DIMENSI

Capaian Pembelajaran	:	Mahasiswa dapat memahami konsep manipulasi array satu dimensi, dan melakukan penggabungan dua atau lebih array menjadi array satu dimensi.
Sub Pokok Bahasan	:	13.1. Konsep dasar penggabungan (merge) array. 13.2. Penggabungan dua buah array menjadi satu array. 13.3. Penggabungan dua buah array menjadi satu array dengan kriteria tertentu.
Daftar Pustaka	:	1. Gaddis, nd.2011. Starting Out with C++ from Control Structures through Objects .8th. Boston: Addison-Wesley. 2. Institue of Distance & Open Learning, n.d. UNIT I Algorithms, Flowcharts & Program Design in: INTRODUCTION TO C++. p. 205 3. Sjukani,Moh .2014. Algoritma (Algoritma & Struktur Data 1) Dengan C, C++, dan Java Edisi 9", Mitra Wacana Media.

PENGGABUNGAN (MERGE) ARRAY SATU DIMENSI

13.1. KONSEP PENGGABUNGAN (MERGE) ARRAY

Merge artinya menggabungkan. Dalam pemrograman, merge biasanya dimaksudkan menggabungkan dua file data dalam *external storage*. Yang akan dipelajari di modul ini adalah menggabungkan data dalam dua buah array satu dimensi. Logika menggabungkan dua buah array dapat juga diaplikasikan untuk menggabungkan dua buah file (Sjukani, 2014).

Konsep penggabungan (merge) array penting untuk dipelajari, karena menjadi dasar bagi berbagai algoritme untuk menyelesaikan berbagai permasalahan. Beberapa penggunaan konsep penggabungan (merge) dan pemecahan (divide) antara lain:

1. Algoritme Divide-and-Conquer.

Algoritma Divide and Conquer merupakan algoritma yang sangat populer di dunia Ilmu Komputer. Divide and Conquer merupakan algoritma yang berprinsip memecah-mecah permasalahan yang terlalu besar menjadi beberapa bagian kecil sehingga lebih mudah untuk diselesaikan. Langkah-langkah umum algoritma Divide and Conquer : (1) **Divide** : Membagi masalah menjadi beberapa sub-masalah yang memiliki kemiripan dengan masalah semula namun berukuran lebih kecil (idealnya berukuran hampir sama); (2) **Conquer** : memecahkan (menyelesaikan) masing-masing masalah (secara rekursif); dan (3) **Combine** : menggabungkan solusi masing-masing masalah sehingga membentuk solusi masalah semula.

2. Algoritme Pengurutan Merge-Sort.

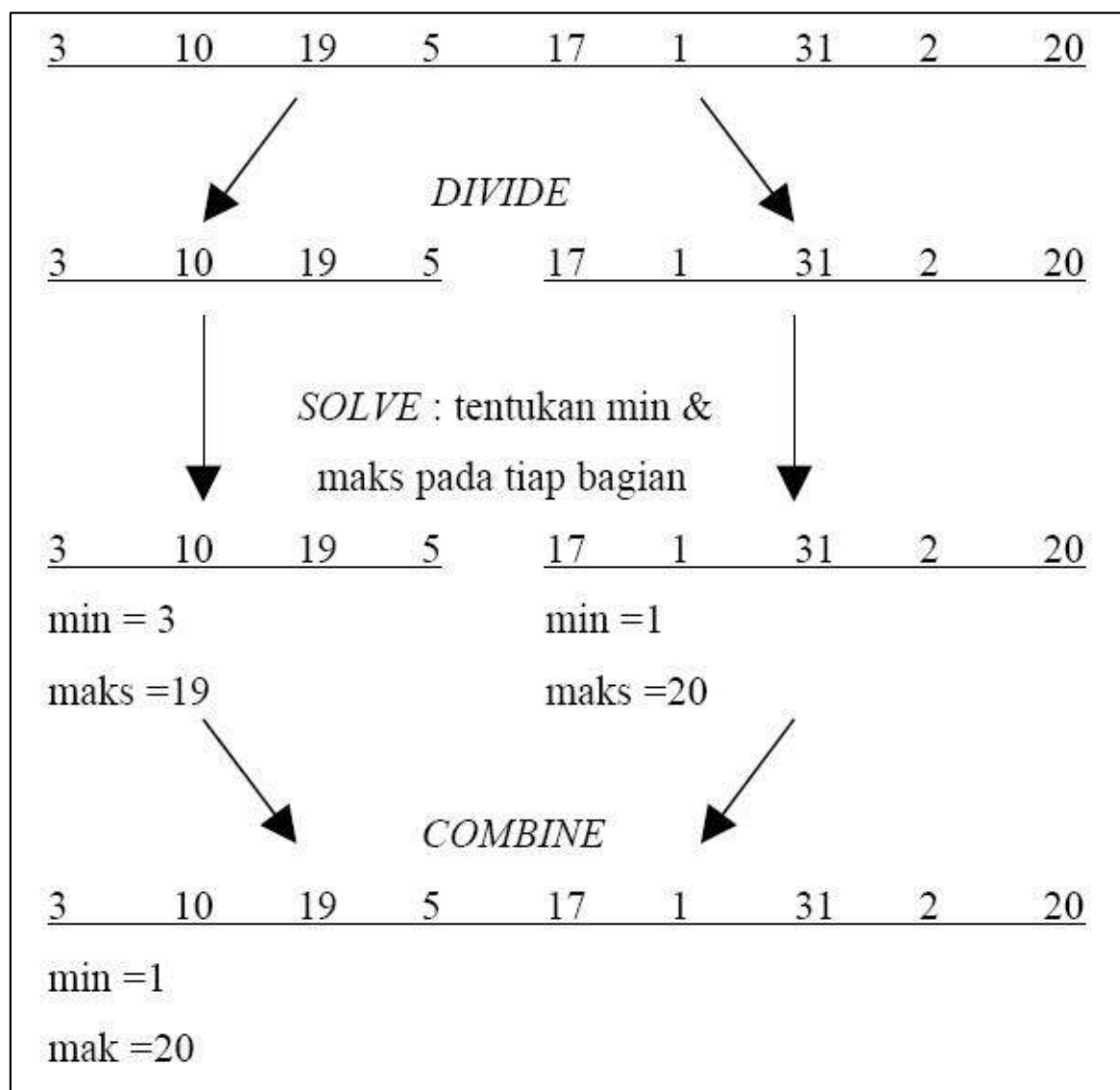
Merge sort merupakan algoritma pengurutan dalam ilmu komputer yang dirancang untuk memenuhi kebutuhan pengurutan atas suatu rangkaian data yang tidak memungkinkan untuk ditampung dalam memori komputer karena jumlahnya yang terlalu besar. Algoritma ini ditemukan oleh John von Neumann pada tahun 1945.

Algoritma pengurutan data merge sort dilakukan dengan menggunakan cara divide-and-conquer yaitu dengan memecah kemudian menyelesaikan setiap



bagian kemudian menggabungkannya kembali. Pertama data dipecah menjadi 2 bagian dimana bagian pertama merupakan setengah (jika data genap) atau setengah minus satu (jika data ganjil) dari seluruh data, kemudian dilakukan pemecahan kembali untuk masing-masing blok sampai hanya terdiri dari satu data tiap blok.

Gambar 1 menyajikan contoh penerapan konsep pemecahan dan penggabungan array untuk mencari nilai terbesar dan terkecil dari suatu array.



Gambar 1. Contoh ilustrasi penerapan algoritme Divide-and-Conquer untuk mencari nilai terkecil dan terbesar

13.2. PENGGABUNGAN DUA BUAH ARRAY MENJADI SATU ARRAY

Pada bagian ini akan dijelaskan teknik penggabungan dua buah array menjadi satu array. Untuk mempermudah penjelasan, perhatikan contoh soal 13.1 berikut ini.

CONTOH SOAL 13.1 – PENGGABUNGAN ARRAY

Sudah ada array **A** satu dimensi yang dibuat dengan **int A[5]**. Dan array **B** yang dibuat dengan **int B[7]**. Kedua buah array sudah ada isinya dengan ilustrasi sebagai berikut :

	0	1	2	3	4
A	12	17	10	5	15

	0	1	2	3	4	5	6
B	25	11	7	25	16	22	14

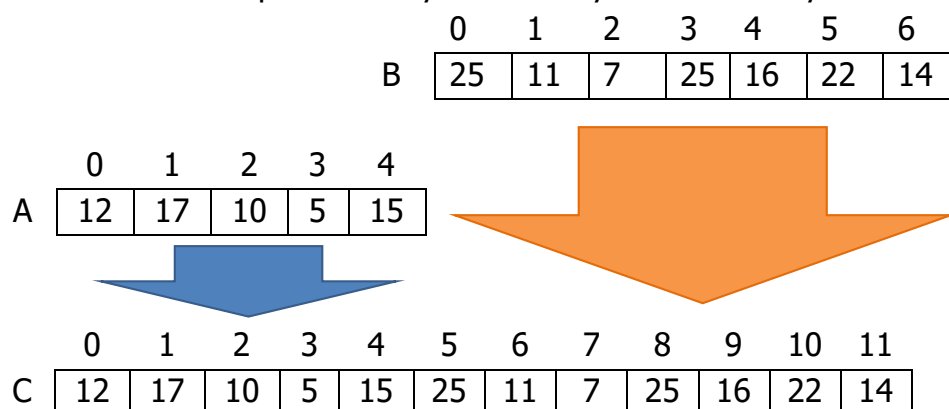
Sudah ada array C satu dimensi yang dibuat dengan **int C[12]**. Belum diisi dengan ilustrasi sebagai berikut :

	0	1	2	3	4	5	6	7	8	9	10	11
C												

Susun program untuk menyalin (menggabungkan) isi array A dan array B ke array C, sehingga isi array C menjadi :

	0	1	2	3	4	5	6	7	8	9	10	11
C	12	17	10	5	15	25	11	7	25	16	22	14

Berikut ini ilustrasi proses menyalin isi array A dan isi array B ke dalam array C



Analisis dan Penyelesaian Soal

Untuk melakukan penggabungan array A dan B, pada dasarnya tinggal melakukan perulangan untuk setiap elemen array A[] dan B[] dan memindahkannya ke array C[].



Berikut ini algoritme atau kerangka pikir untuk menyelesaikan soal 13.1 di atas:

1. Inisialisasi array A[] dan isinya
2. Inisialisasi array B[] dan isinya
3. Inisialisasi array C[] tanpa isi (kosong)
4. Lakukan penelusuran isi array A[] lalu masukkan ke array C[] secara berurutan.
5. Lakukan penelusuran isi array B[] lalu masukkan ke array C[] secara berurutan (indeks melanjutkan dari langkah 4).
6. Cetak isi array C.

Berdasarkan algoritme (kerangka pikir) di atas, selanjutnya dapat diimplementasikan ke dalam bentuk program seperti pada Program 1 berikut ini.

Program 1. Penggabungan dua array menjadi satu array.

```
1  #include <stdio.h>
2  int main()
3  {
4      int A[5] = {12,17,10,5,15};
5      int B[7] = {25,11,7,25,16,22,14};
6      int C[12] = {0};
7      int I, J;
8
9      //cetak isi array A
10     printf("Isi Array A : ");
11     for (I=0; I<5; I++) {
12         printf("%3i", A[I]);
13     }
14     //cetak isi array B
15     printf("\nIsi Array B : ");
16     for (I=0; I<7; I++) {
17         printf("%3i", B[I]);
18     }
19
20     J = 0; //index ini untuk array C
21     for(I=0; I<5; I++) {
22         C[J] = A[I];
23         J++;
24     }
25     for(I=0; I<7; I++) {
26         C[J] = B[I];
27         J++;
28     }
29
30     //cetak isi array C
31     printf("\nIsi Array C : ");
```



```

32     for (I=0; I<12; I++) {
33         printf("%3i", C[I]);
34     }
35     return 0;
36 }

```

Penjelasan Program 1:

- Pada baris 4-7 dilakukan deklarasi array A[] dan B[] lengkap dengan isinya, array C[] masih kosong, dan beberapa variabel yang digunakan dalam program.
- Pada baris 9-18, program melakukan pencetakan isi dari array A[] dan B[] agar saat dijalankan kita dapat mengetahui isi array aslinya.
- Baris 20-28 merupakan proses utama program yang melakukan penelusuran ke setiap elemen array A[] dan B[], serta memindahkannya secara berurutan ke array C[]. Variabel yang digunakan untuk mengatur indeks dari array C[] adalah J.
- Bagian akhir program, yaitu baris 32-34, dilakukan pencetakan isi array C[] untuk memastikan bahwa seluruh isi array A[] dan B[] telah berhasil dipindahkan (digabungkan) dengan benar.

13.3. PENGGABUNGAN DUA BUAH ARRAY MENJADI SATU ARRAY DENGAN KRITERIA TERTENTU

Contoh Soal 13.1 di atas merupakan contoh penggabungan 2 buah array satu dimensi, yang mana seluruh data diikutsertakan dalam penggabungan. Pada kasus lainnya, terkadang diminta untuk menggabungkan 2 buah array, namun dengan kriteria penggabungan tertentu. Contoh soal 13.2 merupakan contoh soal penggabungan array dengan kriteria tertentu.

CONTOH SOAL 13.2 – PENGGABUNGAN ARRAY DENGAN KRITERIA

Sudah ada array **A** satu dimensi yang dibuat dengan **int A[5]**. Dan array **B** yang dibuat dengan **int B[7]**. Kedua buah array sudah ada isinya dengan ilustrasi sebagai berikut :

	0	1	2	3	4
A	12	17	10	5	15



	0	1	2	3	4	5	6
B	25	11	7	25	16	22	14

Sudah ada array C satu dimensi yang dibuat dengan int **C[12]**. Belum diisi dengan ilustrasi sebagai berikut :

	0	1	2	3	4	5	6	7	8	9	10	11
C												

Susun program untuk menyalin (menggabungkan) isi array A yang bernilai ganjil, dan isi array B yang lebih besar dari 15, ke array C, sehingga isi array C menjadi :

	0	1	2	3	4	5	6	7	8	9	10	11
C	17	5	15	25	25	16	22					

Analisis dan Penyelesaian Soal

Untuk menyelesaikan soal 13.2 pada dasarnya sama seperti menyelesaikan soal 13.1. Keduanya melibatkan penelusuran array menggunakan perulangan. Namun demikian, pada soal 13.2 perlu ditambahkan kondisi untuk menyeleksi nilai A[] dan B[] yang akan dimasukkan ke array C[].

Berikut ini algoritme atau kerangka pikir untuk menyelesaikan soal 13.2 di atas:

1. Inisialisasi array A[] dan isinya
2. Inisialisasi array B[] dan isinya
3. Inisialisasi array C[] tanpa isi (kosong)
4. Lakukan penelusuran isi array A[], jika nilai merupakan bilangan ganjil maka masukkan ke array C[] secara berurutan.
5. Lakukan penelusuran isi array B[], jika nilai merupakan bilangan yang lebih besar dari 15, masukkan ke array C[] secara berurutan (indeks melanjutkan dari langkah 4).
6. Cetak isi array C.

Berdasarkan algoritme (kerangka pikir) di atas, selanjutnya dapat diimplementasikan ke dalam bentuk program seperti pada Program 2 berikut ini.



Program 2. Penggabungan dua array menjadi satu array dengan kriteria tertentu.

```
1  #include <stdio.h>
2  int main()
3  {
4      int A[5] = {12,17,10,5,15};
5      int B[7] = {25,11,7,25,16,22,14};
6      int C[12] = {0};
7      int I, J;
8
9      //cetak isi array A
10     printf("Isi Array A : ");
11     for (I=0; I<5; I++) {
12         printf("%3i", A[I]);
13     }
14     //cetak isi array B
15     printf("\nIsi Array B : ");
16     for (I=0; I<7; I++) {
17         printf("%3i", B[I]);
18     }
19     J = 0; //index ini untuk array C
20     for(I=0; I<5; I++) {
21         if (A[I]%2==1) {
22             C[J] = A[I];
23             J++;
24         }
25     }
26     for(I=0; I<7; I++) {
27         if (B[I] > 15) {
28             C[J] = B[I];
29             J++;
30         }
31     }
32
33     //cetak isi array C
34     printf("\nIsi Array C : ");
35     for (I=0; I<12; I++) {
36         printf("%3i", C[I]);
37     }
38     return 0;
39 }
```

Penjelasan Program 2:

Program 2 di atas pada dasarnya sama seperti program 1. Perbedaannya terletak pada penambahan struktur kondisi pada baris 21-24 yang hanya memasukkan isi array A[] yang bernilai ganjil, serta kondisi pada baris 27-30 yang hanya memasukkan isi array B[] yang bernilai > 15. Perhatikan juga variabel J sebagai pengatur indeks array C[].



Variabel J hanya bertambah jika kriteria yang diminta terpenuhi, atau dengan kata lain, penambahan nilai variabel J harus diletakkan di dalam kondisi IF.



SOAL LATIHAN

Soal 1.

Sudah ada array X satu dimensi yang dibuat dengan int **X[4]** dan array Y yang dibuat dengan int **Y[6]**. Kedua buah array tersebut sudah ada isinya dengan ilustrasi sebagai berikut:

	0	1	2	3
X[4]	12	2	7	10

	0	1	2	3	4	5
Y[6]	15	4	16	20	25	30

Sudah ada array Z satu dimensi yang dibuat dengan int **Z[10]** belum ada isinya. Susun program bahasa C untuk menggabung nilai yang lebih kecil dari 10 untuk isi array **X** dan nilai yang ada dilokasi ganjil untuk isi array Y, sehingga isi array Z menjadi sebagai berikut:

	0	1	2	3	4	5	6	7	8	9
Z [10]	2	7	4	20	30					

Berasal dari array X Berasal dari array Y

Soal 2.

Pak Budi merupakan dosen di Universitas Budi Luhur yang mengajar 3 matakuliah yaitu Algoritma, Pemrograman Web, dan Metodologi Riset. Asumsikan bahwa jumlah mahasiswa di ketiga matakuliah tersebut adalah 20 mahasiswa dan asumsikan bahwa di ketiga matakuliah tersebut merupakan mahasiswa yang sama. Buatlah program untuk menginput nilai ujian ketiga matakuliah tersebut untuk seluruh mahasiswa secara bergantian, lalu simpan dalam 3 array berbeda. Identitas mahasiswa direpresentasikan dalam bentuk nomor indeks dari array.

Selanjutnya, Pak Budi ingin memilih mahasiswa-mahasiswa berprestasi untuk diikutsertakan dalam lomba PkM, dengan kriteria:



1. Rata-rata nilai dari ketiga matakuliah lebih besar dari 80.
2. Tidak ada matakuliah yang mendapat nilai < 60 .

Bantulah Pak Budi dengan membuat program untuk menyimpan nomor indeks mahasiswa berprestasi ke dalam suatu array satu dimensi sesuai dengan kriteria di atas. Cetaklah daftar mahasiswa berprestasi tersebut di layar (nomor indeksnya).

Ilustrasi isi array:

AL[] : 80 60 90 85 90 90 50 60 85 75 50 60 90 85 90 90 50 60 85 75

PW[] : 90 90 75 40 60 90 50 60 85 85 100 90 50 60 85 75 80 60 90 85

MR[] : 75 40 60 90 85 90 80 60 90 85 100 90 60 85 75 50 60 85 90 50

Isi array mahasiswa berprestasi dari ilustrasi di atas:

PRESTASI[] : 0 5 8 9 14 18



KESIMPULAN

Konsep penggabungan (merge) array penting untuk dipelajari, karena menjadi dasar bagi berbagai algoritme untuk menyelesaikan berbagai permasalahan. Beberapa penggunaan konsep penggabungan (merge) dan pemecahan (divide) antara lain: algoritme divide-and-conquer, algoritma pengurutan merge-sort, konversi bilangan desimal ke biner dengan teknik pembagian 8, dan lain-lain.





FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR

Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan

Jakarta Selatan, 12260

Telp: 021-5853753 Fax : 021-5853752

<http://fti.budiluhur.ac.id>