MODUL MATA KULIAH

ALGORITMA DAN STRUKTUR DATA 1

KP002 – 3 SKS

FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR

JAKARTA
SEPTEMBER 2019

TIM PENYUSUN

Painem, S.Kom, M.Kom
Reva Ragam, S.Kom., M.Kom
Ir. Moch. Sjukani

UNIVERSITAS BUDI LUHUR
FAKULTAS TEKNOLOGI INFORMASI

MODUL PERKULIAHAN #10
MERGE SORT

Capaian Pembelajaran	:	Mahasiswa mampu memahami karakteristik algoritma merge sort dan mengimplementasi algoritma merge ke dalam bahasa pmrograman
Sub Pokok Bahasan	:	1.1. Pengertian Merge Sort 1.2. Merge sort dengan jumlah elemen array ganjil 1.3. Merge Sort dengan jumlah elemen array genap
Daftar Pustaka	:	1. Sjukani M, "Struktur data dengan C++(Algoritma dan Struktur Data 2 dengan C, C++)", Mitra Wacana Media, 2007 2. Kristanto Andri, "Algoritma dan Pemrograman dengan C++", Graha Ilmu, 2003 3. Darmawan Erico, "Pemrograman Dasar C-Java-C#",Informatika,2012

MERGE SORT

1.1. KONSEP MERGE SORT

Merge Sort adalah algoritma yang berdasarkan strategi divide-and-conquer. Strateginya adalah dengan membagi sekelompok data yang akan diurutkan menjadi beberapa kelompok kecil terdiri dari maksimal dua nilai untuk dibandingkan dan digabungkan lagi secara keseluruhan.

1) *Divide*

   Memilah elemen-elemen dari rangkaian data menjadi dua bagian dan mengulangi pemilahan hingga satu elemen terdiri maksimal dua nilai

2) *Conquer*

   Mengurutkan masing-masing elemen

3) *Combine*.

   Menggabungkan dua bagian tersebut secara rekursif untuk mendapatkan rangkaian data berurutan. Proses rekursi berhenti jika mencapai elemen dasar. Hal ini terjadi bilamana bagian yang akan diurutkan menyisakan tepat satu elemen. Sisa pengurutan satu elemen tersebut menandakan bahwa bagian tersebut telah trurut sesuai rangkaian.

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 ge nap) 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. Setelah itu digabungkan kembali dengan membandingkan pada blok yang sama apakah data pertama lebih besar daripada data ke - tengah+1, jika ya maka data ke - tengah+1 dipindah sebagai data pertama, kemudian data ke - pertama sampai ke - tengah digeser menjadi data ke - dua sampai ke - tengah+1, demikian seterusnya sampai menjadi satu blok utuh seperti awalnya. Sehingga metode merge sort merupakan metode yang membutuhkan fungsi rekursi untuk penyelesaiannya.

1.2. MERGE SORT DENGAN JUMLAH ELEMEN ARRAY GENAP

Langkah-langkah pengurutan dengan merge sort :

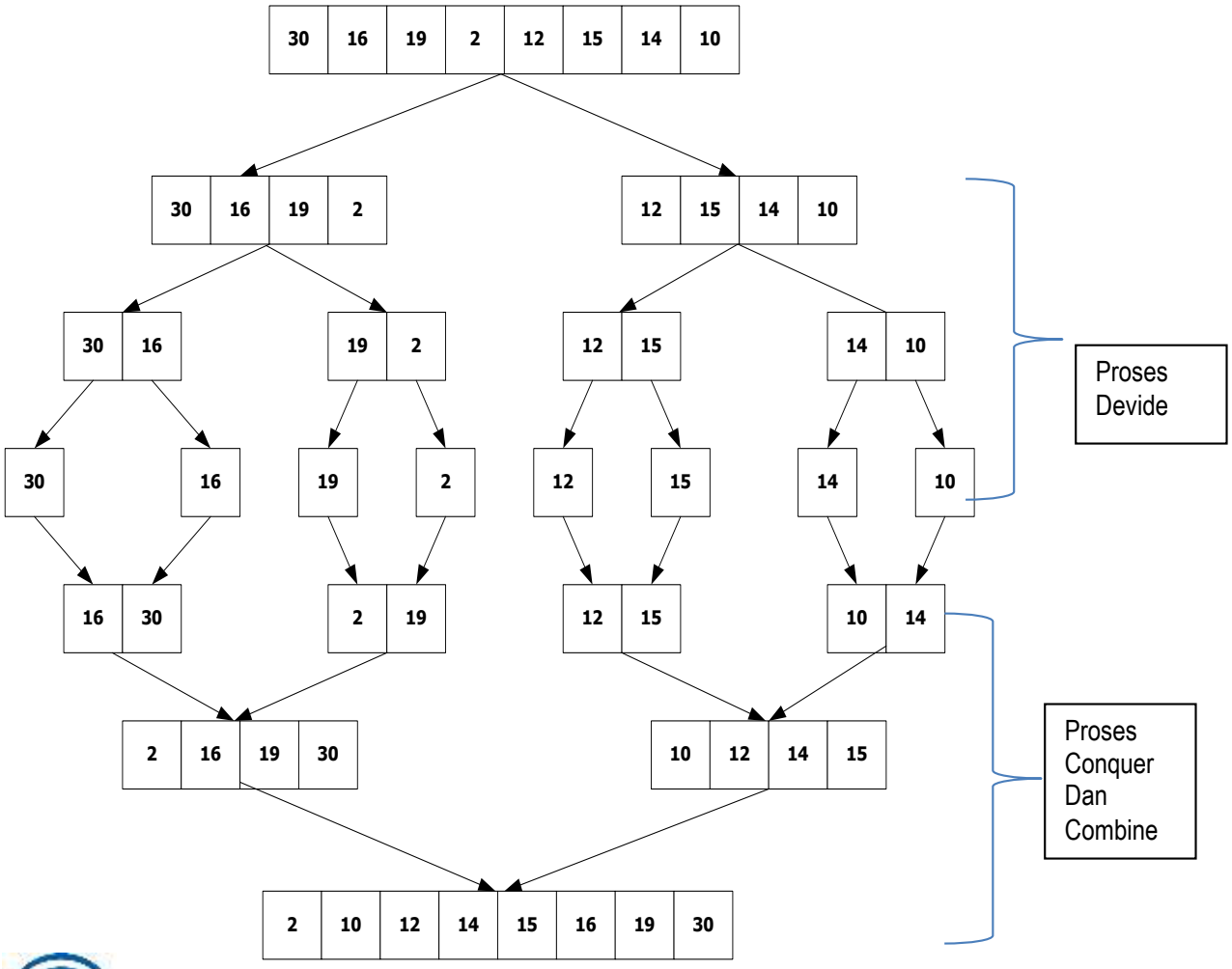
Misalkan sudah ada array satu dimensi dengan jumlah elemen array 8 dengan ilustrasi sebagai berikut :

	0	1	2	3	4	5	6	7
A	30	16	19	2	12	5	14	10

Buat program untuk mengurutkan (sorting) dengan metode Merge Sort sehingga isi array menjadi sebagai berikut :

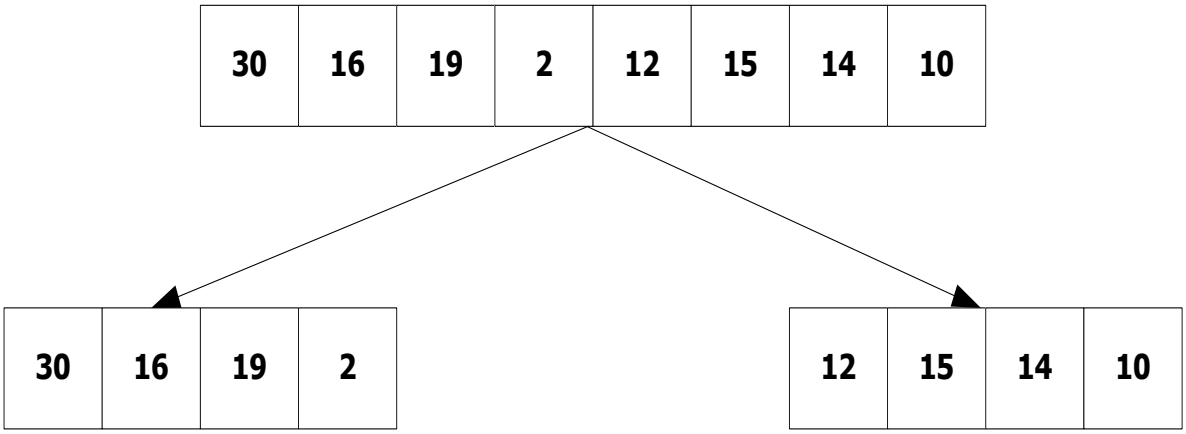
0	1	2	3	4	5	6	7
2	5	10	12	14	16	19	30

Ilustrasi merge sort :

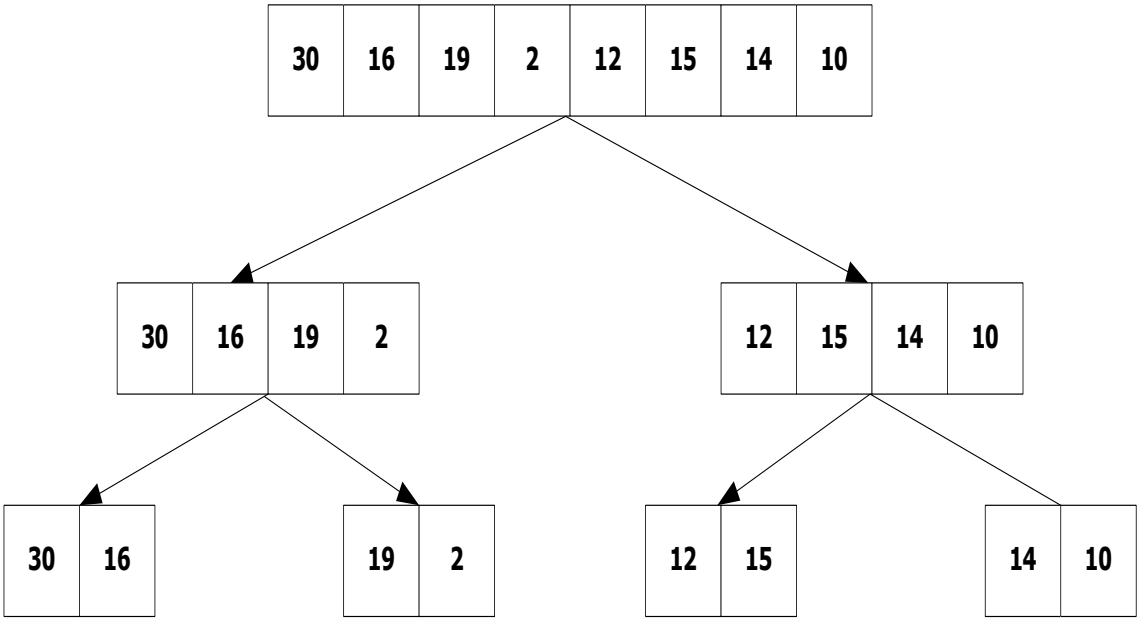


Langkah-langkah merge sort :

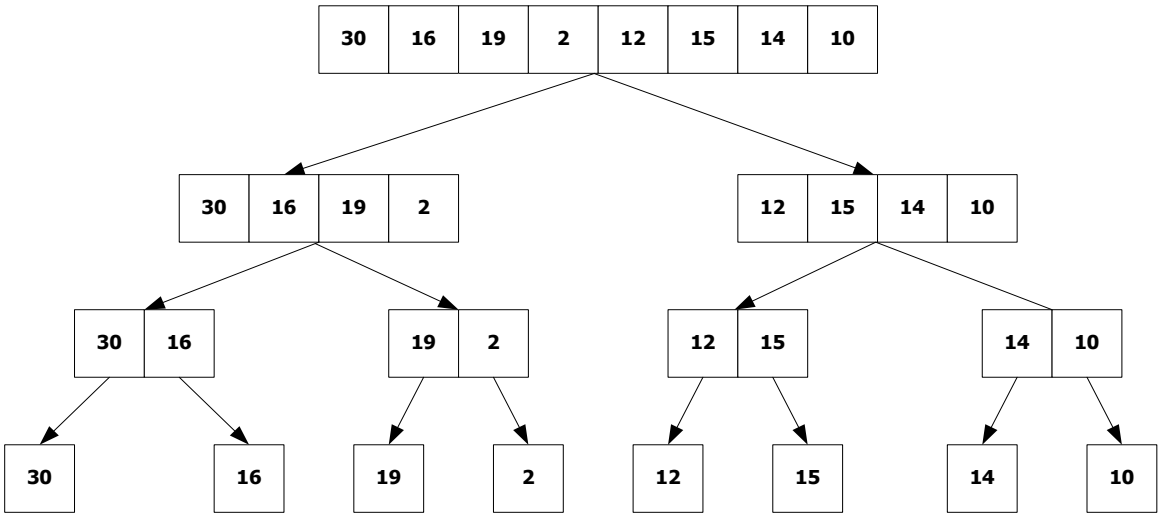
a) Jumlah elemen array dibagi menjadi dua(mid=(n/2)) , (sebelah kiri elemen array yang isinya 30,16,19,2) dan sebelah kanan (12,5,14,10). Ilustrasi sebagai berikut :



b) Langkah berikutnya elemen array kiri dibagi 2, dan elemen array kanan dibagi dua. Ilustrasi sebagai berikut :



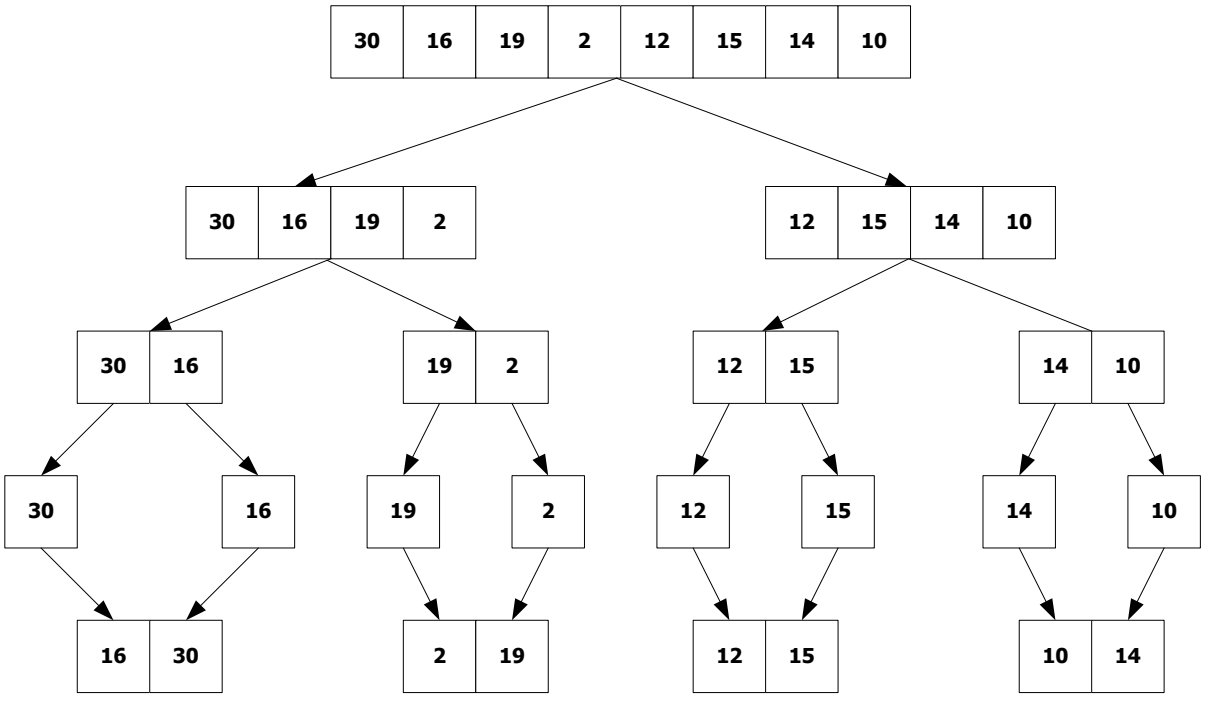
c) Sama dengan langkah yang ke B, masing –masing dibagi dua. Ilustrasi sebagai berikut :



d) Untuk setiap elemen akan dilakukan perbandingan. Jika isi elemen array sebelah kanan lebih kecil dari isi array sebelah kiri maka dilakukan nilainya ditukar

Contoh :

Karena 16 lebih kecil dari 30 maka kedua bilangan tersebut ditukar kemudian digabungkan sehingga mendapatkan hasil= 16,30. Lakukan hal yang sama untuk bagian yang lain. Ilustrasi sebagai berikut :

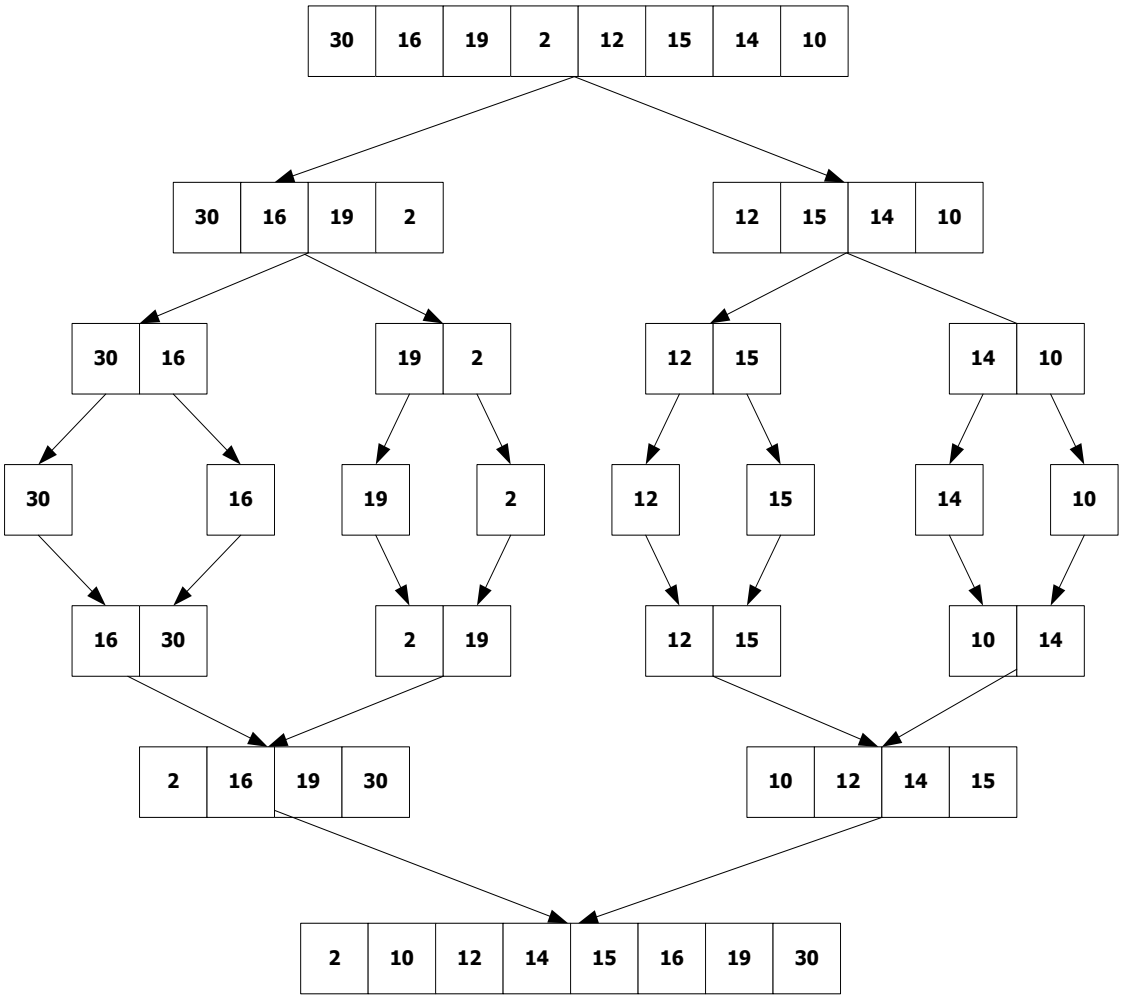


e) Langkah selanjutkan adalah melakukan pengurutan dan penggabungan array seperti tahap d).

Contoh :

2 dibandingkan dengan 16, karena 2 lebih kecil maka masukkan angka 2 ke array , kemudian 16 dibandingkan dengan 19, karena 16 lebih kecil maka 16 yang dimasukkan kedalam array, berikutnya 30 dbandingkan dengan 19, karena 19 lebih kecil makan 19 dimasukkan kedalam array, baru kemudian 30

Ilustrasi sebagai berikut :



Program merge sort dengan jumlah elemen array genap :

[*] mergesort.cpp

```
#include <stdio.h>
#define MAX 8
int Data[MAX];
int temp[MAX];
void merge(int Data[], int temp[], int kiri, int tengah, int kanan)
{
   int i, left_end, num_elements, tmp_pos;
   left_end = tengah - 1;
   tmp_pos = kiri;
   num_elements = kanan - kiri + 1;
   while ((kiri <= left_end) && (tengah <= kanan))
     {
        if (Data[kiri] <= Data[tengah])
        {
            temp[tmp_pos] = Data[kiri];
            tmp_pos = tmp_pos + 1;
            kiri = kiri +1;
        }
        else
        {
            temp[tmp_pos] = Data[tengah];
            tmp_pos = tmp_pos + 1;
            tengah = tengah + 1;
        }
     }

      while (kiri <= left_end)
      {
            temp[tmp_pos] = Data[kiri];
            kiri = kiri + 1;
            tmp_pos = tmp_pos + 1;
      }
      while (tengah <= kanan)
      {
            temp[tmp_pos] = Data[tengah];
            tengah = tengah + 1;
            tmp_pos = tmp_pos + 1;
      }

      for (i=0; i <= num_elements; i++)
      {
            Data[kanan] = temp[kanan];
            kanan = kanan - 1;
      }
}
```

```
46
47  void m_sort(int Data[], int temp[], int kiri, int kanan)
48  {
49      int tengah;
50      if (kanan > kiri)
51      {
52          tengah = (kanan + kiri) / 2;
53          m_sort(Data, temp, kiri, tengah);
54          m_sort(Data, temp, tengah+1, kanan);
55          merge(Data, temp, kiri, tengah+1, kanan);
56      }
57  }
58
59  void mergeSort(int Data[], int temp[], int array_size)
60  {
61      m_sort(Data, temp, 0, array_size - 1);
62  }
```

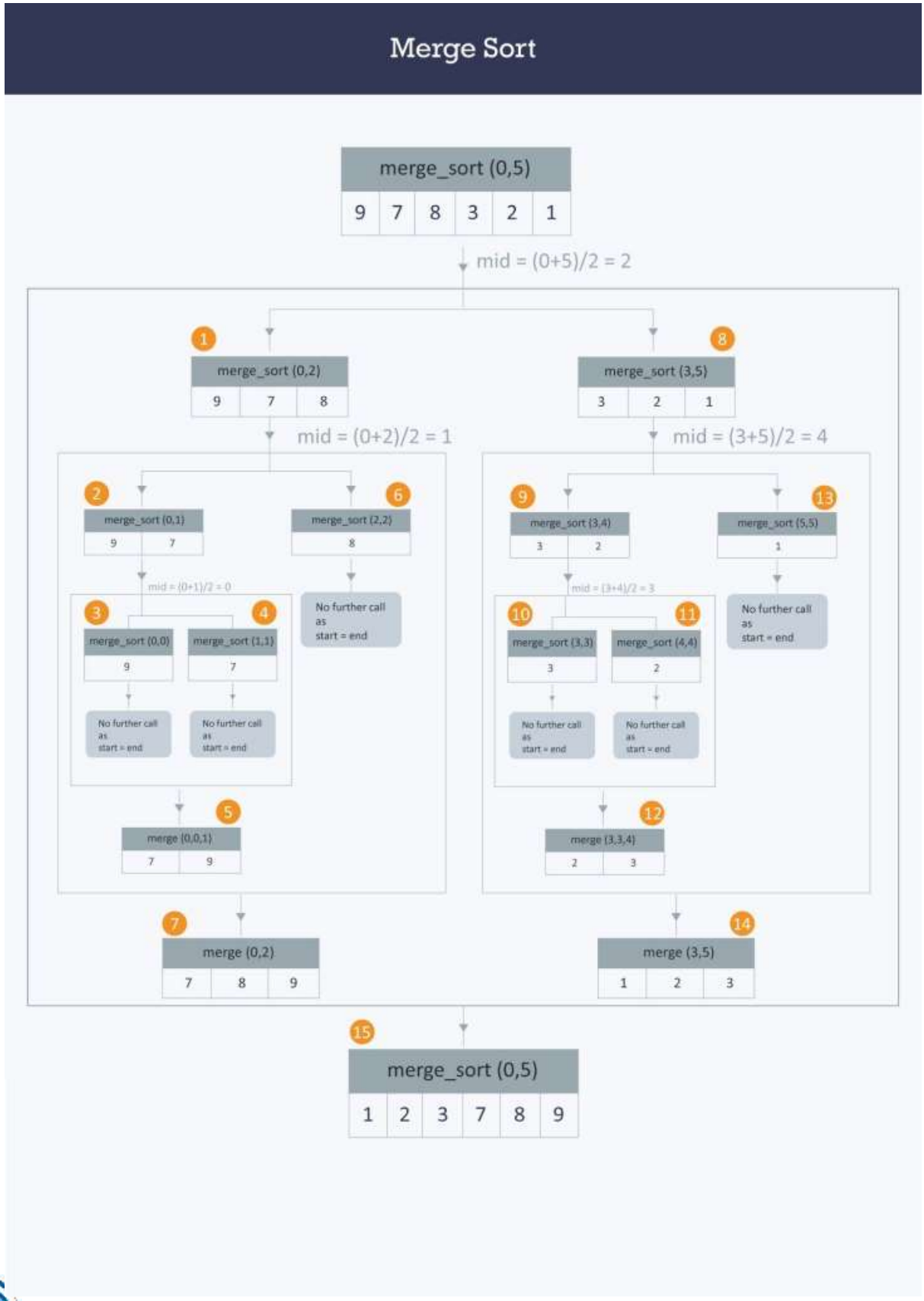
```
63
64  main()
65  {
66      int i;
67      printf("Masukkan DATA SEBELUM TERURUT : \n");
68      for (i = 0; i < MAX; i++)
69      {
70          printf ("Data ke %i : ", i+1);
71          scanf ("%d", &Data[i]);
72      }
73      mergeSort(Data, temp, MAX);
74      printf("\nDATA SETELAH TERURUT : ");
75      for (i = 0; i < MAX; i++)
76      printf("%d  ", Data[i]);
77      printf("\n");
78
79      return(0);
80  }
```

Output

```
D:\Raker RPS 13 -14 Feb 2018\program\mergesort.exe
Masukkan DATA SEBELUM TERURUT :
Data ke 1 : 30
Data ke 2 : 16
Data ke 3 : 19
Data ke 4 : 2
Data ke 5 : 12
Data ke 6 : 5
Data ke 7 : 14
Data ke 8 : 10

DATA SETELAH TERURUT : 2  5  10  12  14  16  19  30
```

1.3. MERGE SORT DENGAN JUMLAH ELEMEN ARRAY GENAP



Sebagaimana yang dapat dipahami dari gambar di atas, pada setiap langkah, list dengan ukuran M dibagi menjadi 2 sublist berukuran M/2, hingga pembagian tidak dapat lagi dilakukan. Lebih mudahnya, diketahui terdapat array A yang lebih kecil dengan elemen (9,7,8)

Pada langkah pertama, list berukuran 3 dibagi menjadi 2 sublist dengan sublist pertama terdiri dari (9,7) dan sublist kedua terdiri dari (8). Sekarang, list pertama yang terdiri dari elemen (9,7) dibagi lagi menjadi 2 sublist terdiri dari elemen (9) dan (7) secara berurutan.

Sublist tersebut tidak dapat dibagi lagi karena setiap sublist hanya memiliki 1 elemen, kini kita akan mulai menggabungkan list tersebut. 2 sublist yang terbentuk pada langkah terakhir akan digabungkan sesuai urutan menggunakan prosedur yang disebutkan di atas dan membentuk list baru (7,9). Dengan melakukan backtracking, kita akan menggabungkan list yang berisi elemen (8)dengan list ini, sehingga list baru yang tersortir terbentuk (7,8,9).

SOAL LATIHAN

Soal -1

Sudah ada array satu dimensi dengan jumlah elemen array int A[9] sudah ada isinya dengan ilustrasi sebagai berikut :

	0	1	2	3	4	5	6	7	8
A	30	16	19	2	12	5	14	10	20

Buat program untuk mengurutkan (sorting) dengan metode Merge Sort sehingga isi array menjadi sebagai berikut : (Sertakan tahapan merge sort)

0	1	2	3	4	5	6	7	8
2	5	10	12	14	16	19	20	30

KESIMPULAN

Merge Sort adalah algoritma yang berdasarkan strategi divide-and-conquer. Strateginya adalah dengan membagi sekelompok data yang akan diurutkan menjadi beberapa kelompok kecil terdiri dari maksimal dua nilai untuk dibandingkan dan digabungkan lagi secara keseluruhan.

a) *Divide*

   Memilah elemen-elemen dari rangkaian data menjadi dua bagian dan mengulangi pemilahan hingga satu elemen terdiri maksimal dua nilai

b) *Conquer*

   Mengurutkan masing-masing elemen

c) *Combine*.

   Menggabungkan dua bagian tersebut secara rekursif untuk mendapatkan rangkaian data berurutan.

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