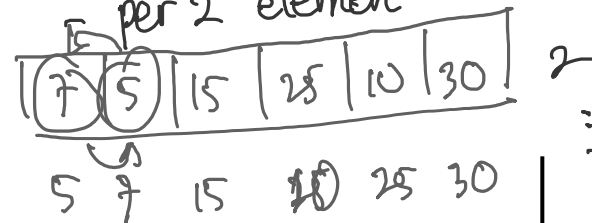
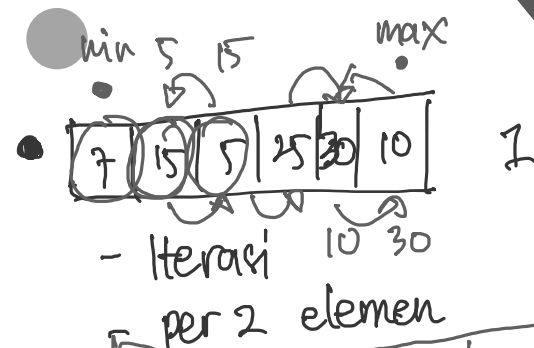
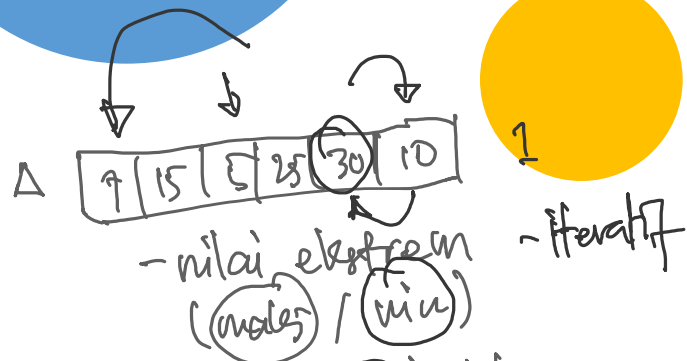
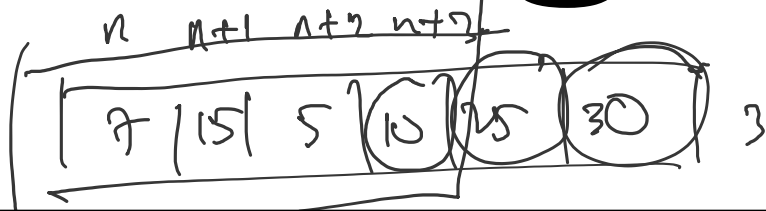


- Bubble sort
- Selection sort



Merge Sort



Algoritma Pemrograman 2

Program Studi Teknik
Informatika

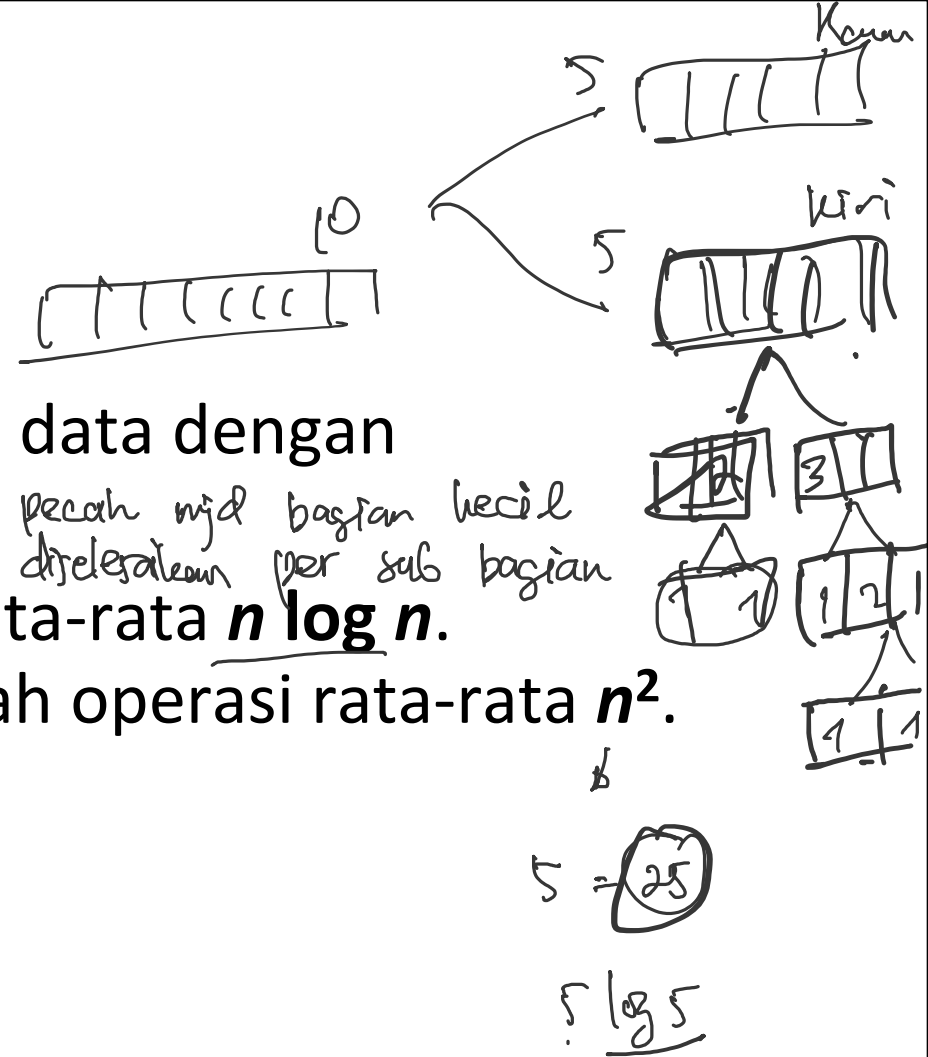
Institut Teknologi Sumatera

Pre-Test

- Bagaimanakah prinsip pengurutan pada algoritma ***Bubble Sort***?
- Berapakah jumlah penukaran elemen yang terjadi apabila kita ingin mengurutkan elemen dari bilangan terkecil hingga terbesar dengan kondisi awal array terurut dari bilangan terbesar hingga terkecil?

Pendahuluan

- **Merge Sort** merupakan algoritma pengurutan data dengan berdasarkan prinsip **Divide and Conquer** → pecah menjadi bagian kecil diselesaikan per sub bagian
- Algoritma ini efisien dengan jumlah operasi rata-rata **$n \log n$** .
Bandingkan dengan **Bubble Sort** dengan jumlah operasi rata-rata **n^2** .



Prinsip Kerja Algoritma

5

A =

336	480	352	979	321
-----	-----	-----	-----	-----

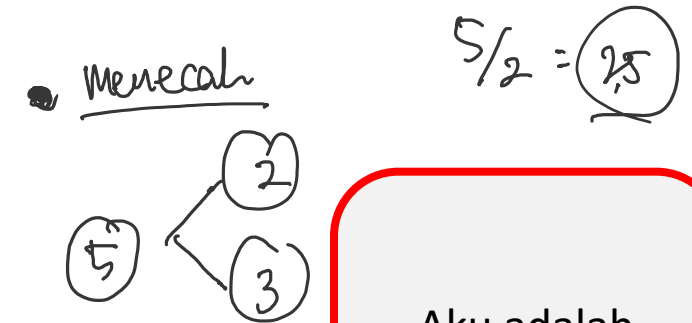
Sub program ✓

Pointer

Rekursif

Bagi array ke
dalam 2 bagian
dengan ukuran
seimbang
(diusahakan)

Prinsip Kerja Algoritma



Aku adalah
sub-array Kiri

- memecah (A)

A

0	1	2	3	4
336	480	352	979	321

K_i

336	480
-----	-----

- memecah (K_i)

- memecah (K_a)

K_a

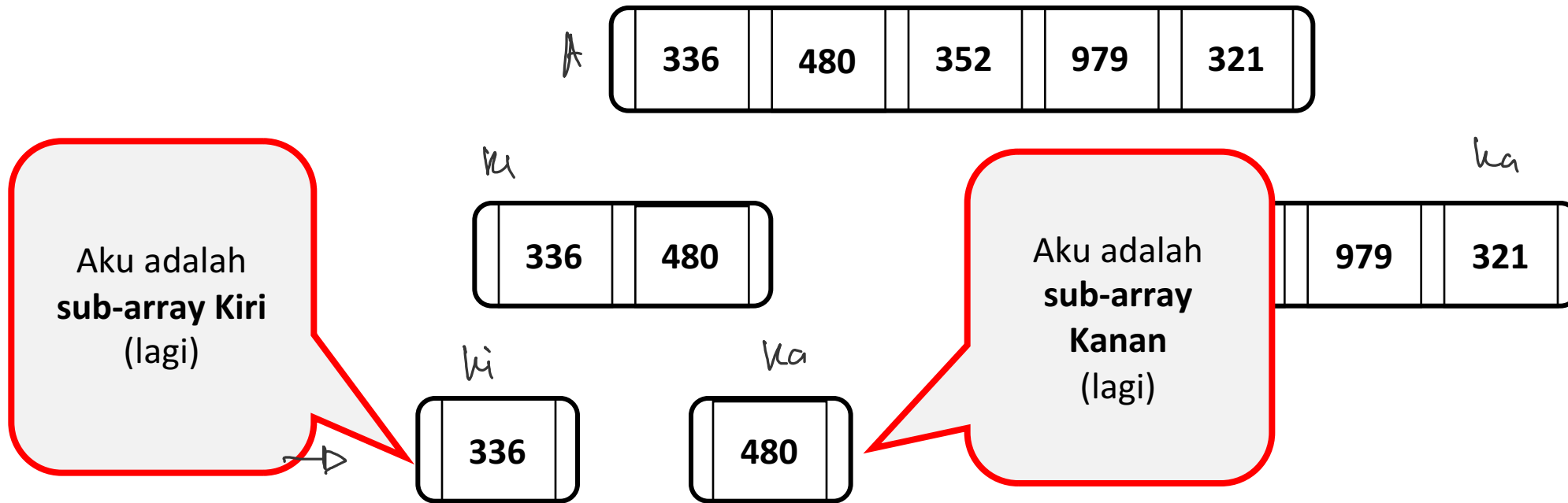
352	979	321
-----	-----	-----

Aku adalah
sub-array
Kanan

Split phase

Prinsip Kerja Algoritma

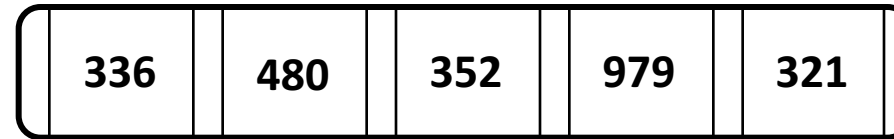
• Divide → • Rekursif



Split phase

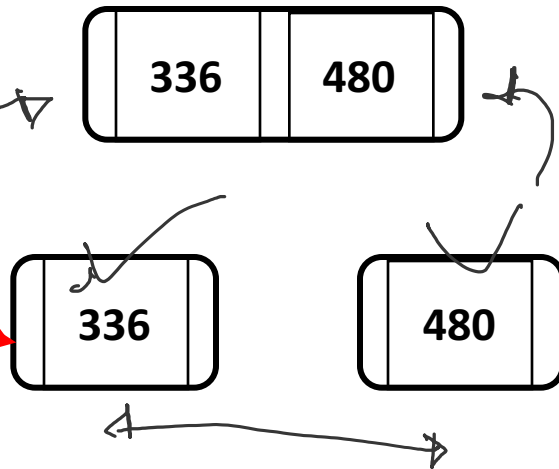
Prinsip Kerja Algoritma

- divide &
- conquer &



Aku juga sudah
terurut!

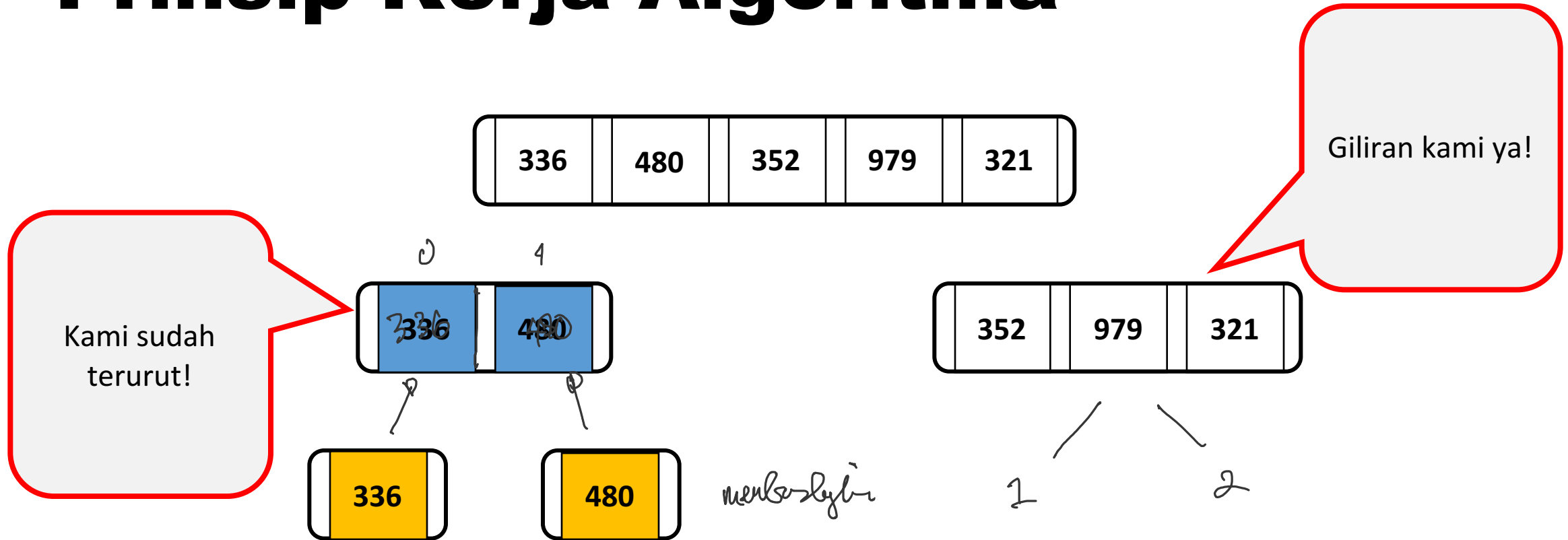
Ayuk kita
berbaris secara
terurut!



Aku sudah
terurut lho!

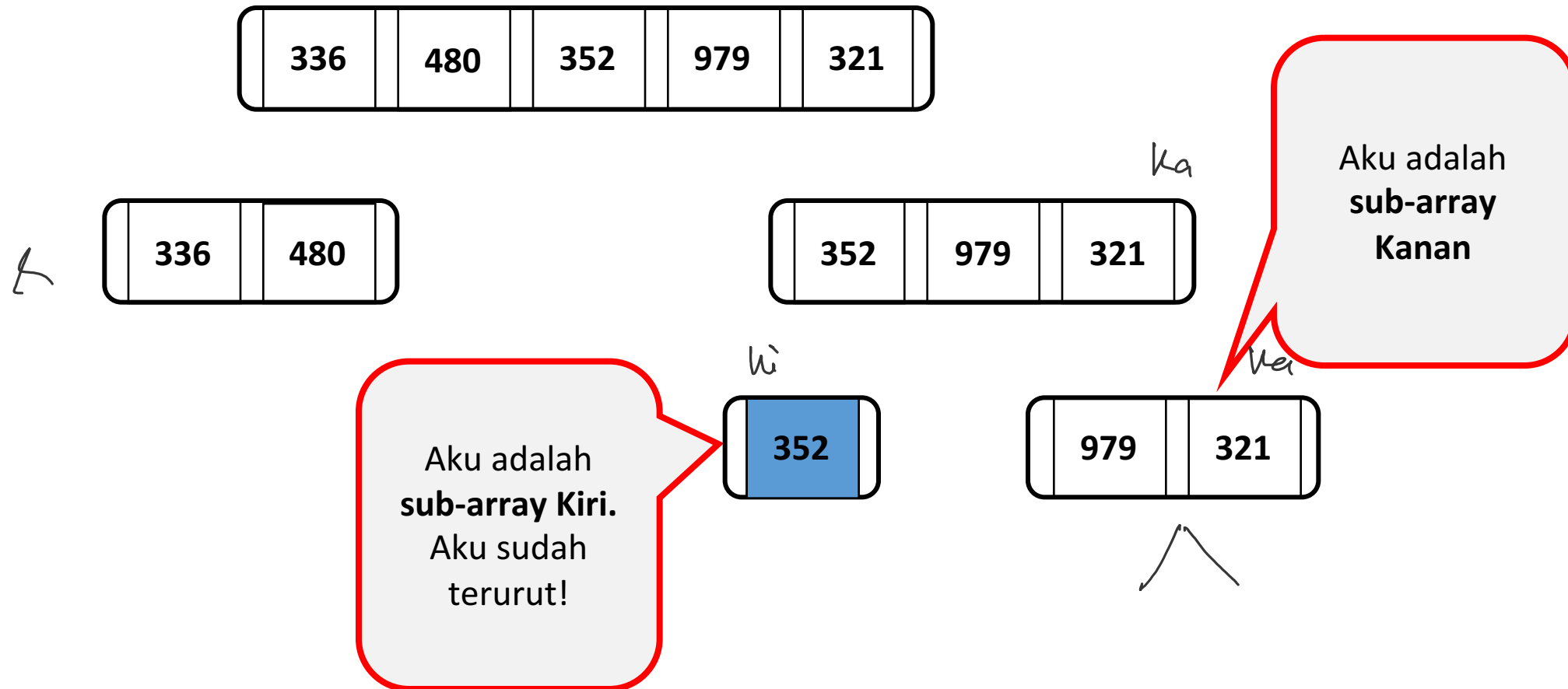
Merge phase

Prinsip Kerja Algoritma



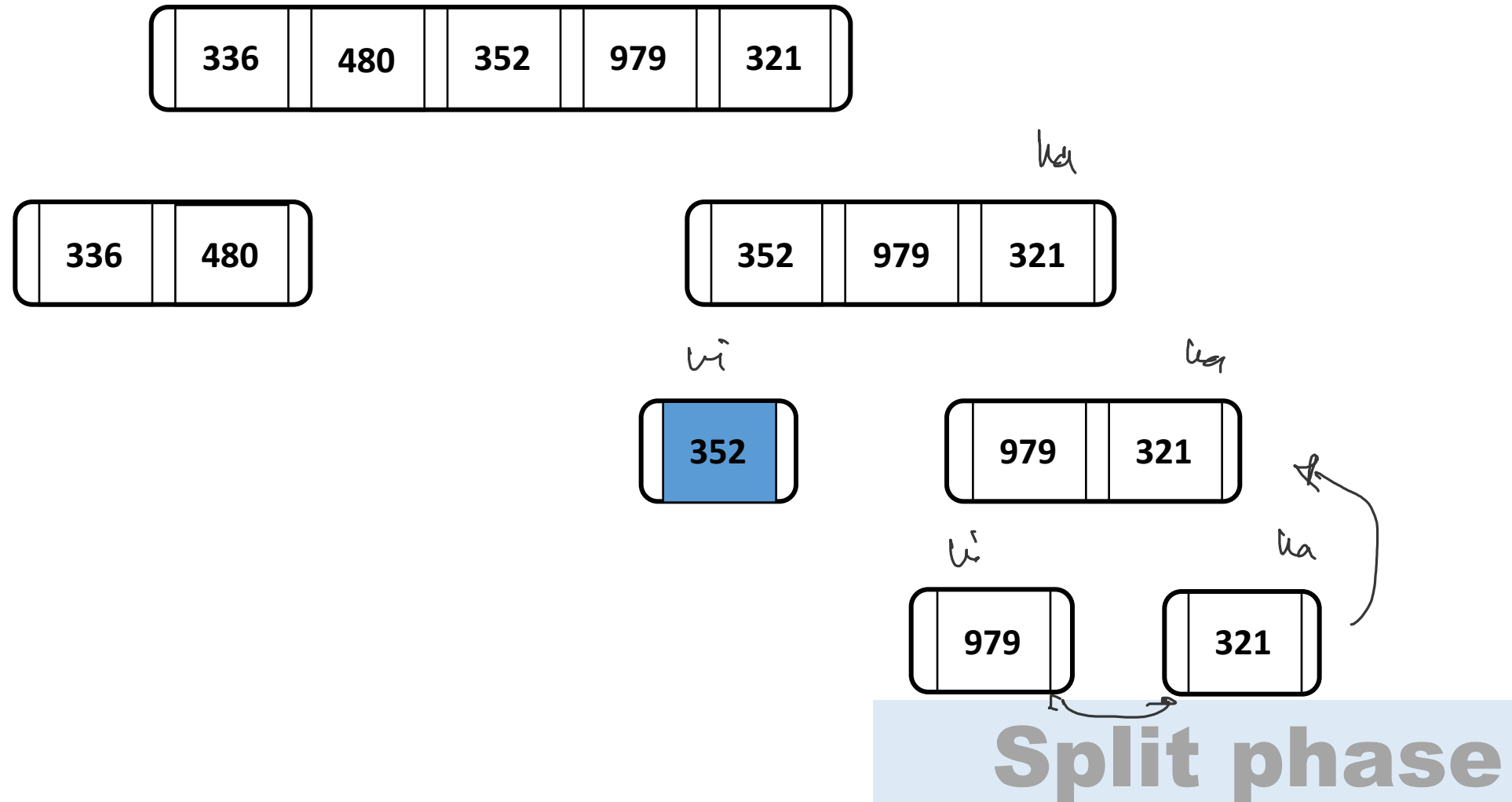
Merge phase

Prinsip Kerja Algoritma

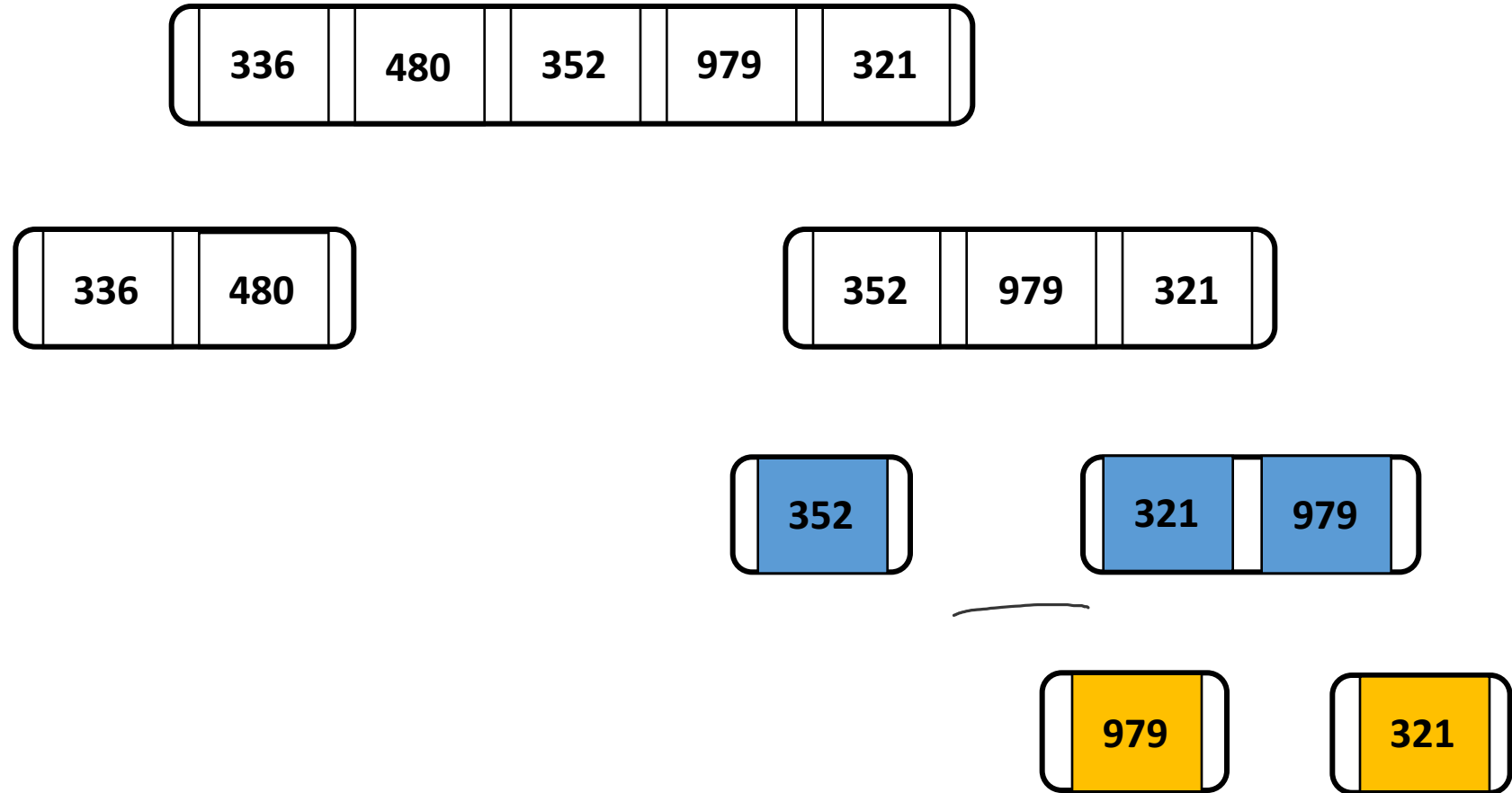


Split phase

Prinsip Kerja Algoritma

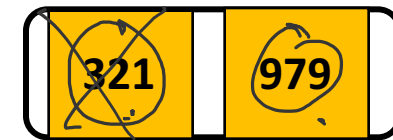
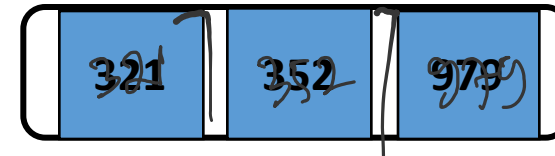
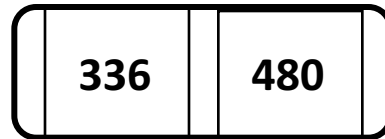
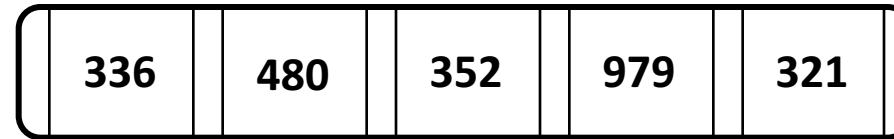


Prinsip Kerja Algoritma



Merge phase

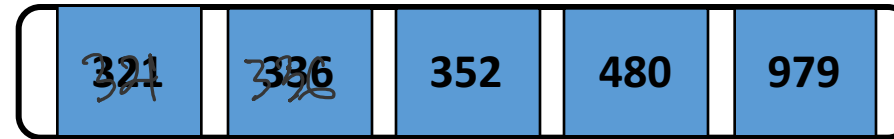
Prinsip Kerja Algoritma



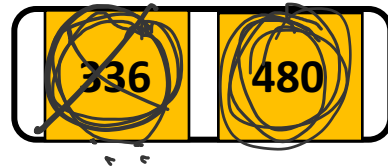
Merge phase

Prinsip Kerja Algoritma

- membagi (Array)
 - menyusun
- input
(membagi)

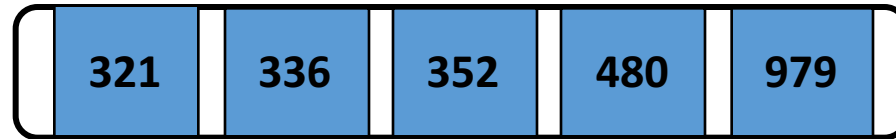


→ min / maksi



Merge phase

Prinsip Kerja Algoritma



Aku sudah
terurut
seluruhnya!



Lebih Baik?

- Apakah algoritma pengurutan *Merge Sort* optimal?
- Adakah algoritma pengurutan lain yang lebih optimal lagi?

Latihan 1

Divide & Conquer

- Lakukan simulasi pengurutan elemen pada *array* berikut dengan menggunakan algoritma **merge-sort**.

3 9 10 27 38 43 82

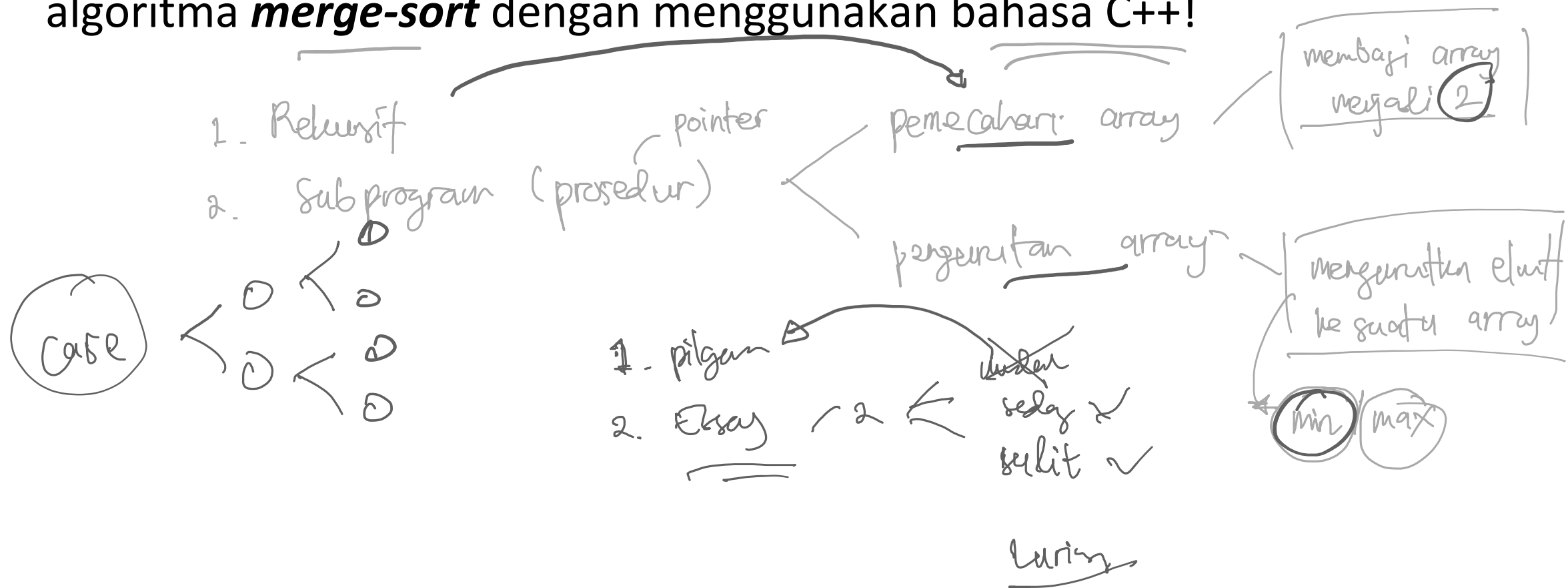
A

38	27	43	3	9	82	10
----	----	----	---	---	----	----



Latihan 2 (Tantangan)

- Berdasarkan prinsip kerja yang telah disampaikan, Implementasikan algoritma ***merge-sort*** dengan menggunakan bahasa C++!



Implementasi

Calling procedure

```
/* l is for left index and r is right index of the
sub-array of data to be sorted */
void mergeSort(int *data, int l, int r) {
    if (l < r) {
        // Same as (l+r)/2, but avoids overflow for
        // large l and h
        int m = (l + (r)) / 2;

        // Split first and second halves
        mergeSort(data, l, m);
        mergeSort(data, m + 1, r);

        // Finally merge first and second halves
        merge(data, l, m, r);
    }
}
```

```
void merge(int *data, int l, int m, int r) {
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;

    /* create temp data arrays */
    int L[n1], R[n2];

    /* Copy data to temp data arrays L[] and R[] */
    for (i = 0; i < n1; i++)
        L[i] = data[l + i];
    for (j = 0; j < n2; j++)
        R[j] = data[m + 1 + j];

    /* Merge the temp data arrays back into data[l..r] */
    i = 0; // Initial index of first subarray
    j = 0; // Initial index of second subarray
    k = l; // Initial index of merged subarray
    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
            data[k] = L[i];
            i++;
        } else {
            data[k] = R[j];
            j++;
        }
        k++;
    }

    /* Copy the remaining elements of L[], if there are any */
    while (i < n1) {
        data[k] = L[i];
        i++;
        k++;
    }

    /* Copy the remaining elements of R[], if there are any */
    while (j < n2) {
        data[k] = R[j];
        j++;
        k++;
    }
}
```





SEKIAN

Selamat Belajar! 😊