

ALGORITMA DAN PEMROGRAMAN II

PENGURUTAN (SORTING)

Rosa Ariani Sukamto

ROSA ARIANI SUKAMTO

Blog: <http://hariiniadalahhadiah.wordpress.com>

Facebook: <https://www.facebook.com/rosa.ariani.sukamto>

Email: rosa_if_itb_01@yahoo.com



PENGURUTAN (SORTING)

- **Penyisipan (Insertion Sort)**
- **Seleksi (Selection Sort)**
- **Gelembung (Bubble Sort)**
- **Urut Cepat (Quick Sort)**



PENYISIPAN (INSERTION SORT) [1]

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 34 | 67 | 23 | 28 | 98 | 15 | 89 | 67 | 28 | 18 |
|----|----|----|----|----|----|----|----|----|----|

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 34 | 67 | 23 | 28 | 98 | 15 | 89 | 67 | 28 | 18 |
|----|----|----|----|----|----|----|----|----|----|

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 23 | 34 | 67 | 28 | 98 | 15 | 89 | 67 | 28 | 18 |
|----|----|----|----|----|----|----|----|----|----|

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 23 | 28 | 34 | 67 | 98 | 15 | 89 | 67 | 28 | 18 |
|----|----|----|----|----|----|----|----|----|----|

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 23 | 28 | 34 | 67 | 98 | 15 | 89 | 67 | 28 | 18 |
|----|----|----|----|----|----|----|----|----|----|

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 15 | 23 | 28 | 34 | 67 | 98 | 89 | 67 | 28 | 18 |
|----|----|----|----|----|----|----|----|----|----|

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 15 | 23 | 28 | 34 | 67 | 89 | 98 | 67 | 28 | 18 |
|----|----|----|----|----|----|----|----|----|----|

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 15 | 23 | 28 | 34 | 67 | 67 | 89 | 98 | 28 | 18 |
|----|----|----|----|----|----|----|----|----|----|

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 15 | 23 | 28 | 28 | 34 | 67 | 67 | 89 | 98 | 18 |
|----|----|----|----|----|----|----|----|----|----|

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 15 | 18 | 23 | 28 | 28 | 34 | 67 | 67 | 89 | 98 |
|----|----|----|----|----|----|----|----|----|----|

PENYISIPAN (INSERTION SORT) [2]

```
int main(){
    int tabInt[10] = {34, 67, 23, 28, 98, 15, 89, 67, 28, 18};
    int i;    int data_sisip;    int j;
    for(i=1; i<10; i++){
        data_sisip = tabInt[i];
        j = i - 1;
        while((data_sisip < tabInt[j]) && (j >= 0)){
            /*jika data array lebih kecil dari data sisip maka data array
            digeser ke belakang*/
            tabInt[j+1] = tabInt[j];
            j = j - 1;
        }
        /*menempatkan data sisip pada array*/
        tabInt[j+1] = data_sisip;
    }
    return 0;
}
```

SELEKSI (SELECTION SORT) [1]

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 34 | 67 | 23 | 28 | 98 | 15 | 89 | 67 | 28 | 18 |
| 15 | 67 | 23 | 28 | 98 | 34 | 89 | 67 | 28 | 18 |
| 15 | 18 | 23 | 28 | 98 | 34 | 89 | 67 | 28 | 67 |
| 15 | 18 | 23 | 28 | 98 | 34 | 89 | 67 | 28 | 67 |
| 15 | 18 | 23 | 28 | 98 | 34 | 89 | 67 | 28 | 67 |
| 15 | 18 | 23 | 28 | 28 | 34 | 89 | 67 | 98 | 67 |
| 15 | 18 | 23 | 28 | 28 | 34 | 89 | 67 | 98 | 67 |
| 15 | 18 | 23 | 28 | 28 | 34 | 67 | 89 | 98 | 67 |
| 15 | 18 | 23 | 28 | 28 | 34 | 67 | 67 | 98 | 89 |
| 15 | 18 | 23 | 28 | 28 | 34 | 67 | 67 | 89 | 98 |

SELEKSI (SELECTION SORT) [2]

```
int main(){
int tabInt[10] = {34, 67, 23, 28, 98, 15, 89, 67, 28, 18};
    int i; int temp; int minIndeks; int j;
    for(i=0; i<(10-1); i++){
        /*inisialisasi indeks elemen minimum*/
        minIndeks = i;

        /*perulangan mencari nilai minimum sepanjang indeks i + 1 sampai jumlah elemen array*/
        for(j=(i + 1); j<10; j++){
            if(tabInt[minIndeks] > tabInt[j]){
                minIndeks = j;
            }
        }

        //menukar posisi elemen
        temp = tabInt[i];
        tabInt[i] = tabInt[minIndeks];
        tabInt[minIndeks] = temp;
    }
    return 0;
}
```

BUBBLE SORT (1)

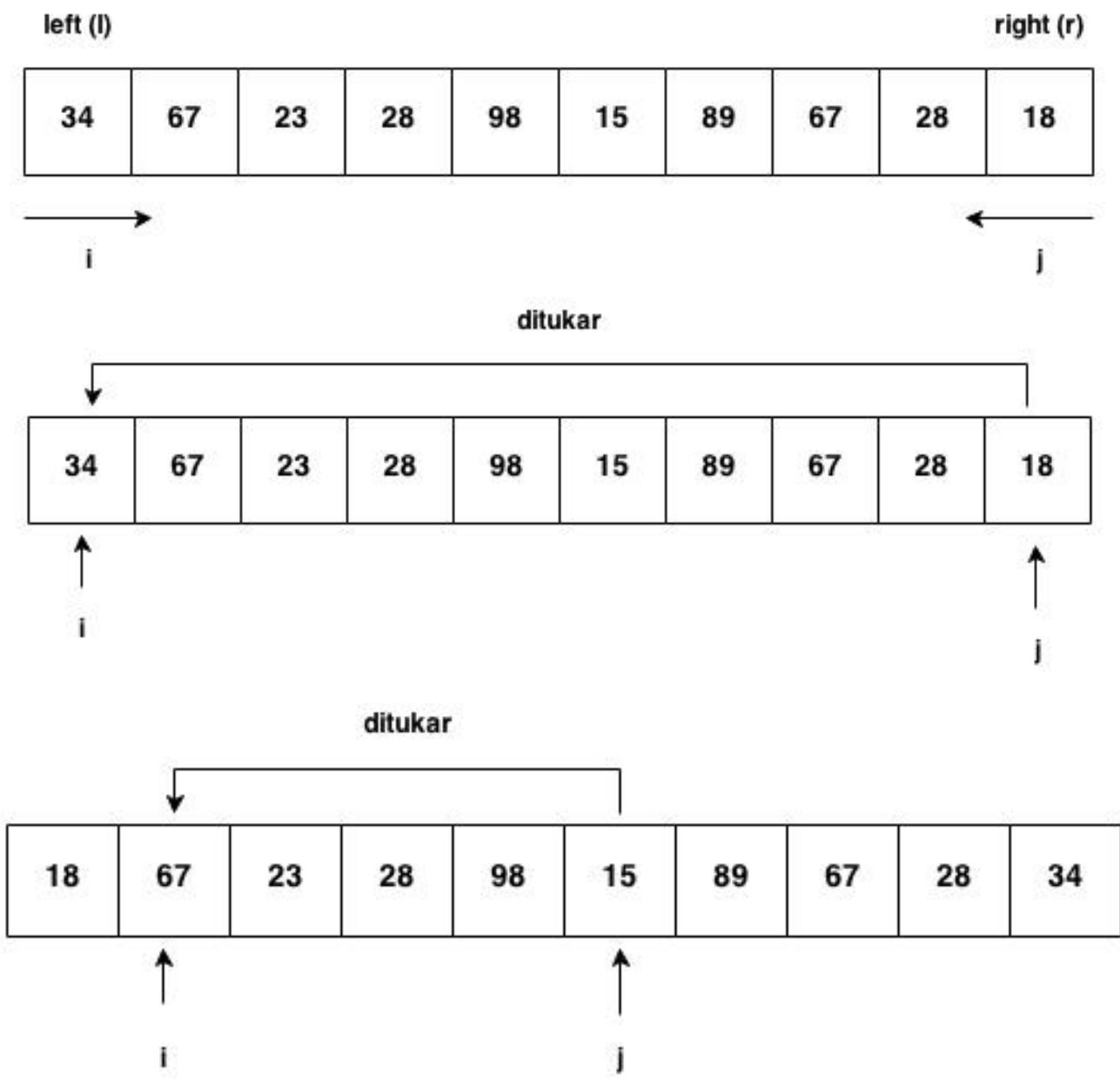
| | | | | |
|----|----|----|----|----|
| 34 | 67 | 23 | 28 | 98 |
| 34 | 67 | 23 | 28 | 98 |
| 34 | 23 | 67 | 28 | 98 |
| 34 | 23 | 28 | 67 | 98 |
| 34 | 23 | 28 | 67 | 98 |
| 23 | 34 | 28 | 67 | 98 |
| 23 | 28 | 34 | 67 | 98 |
| 23 | 28 | 34 | 67 | 98 |
| 23 | 28 | 34 | 67 | 98 |
| 23 | 28 | 34 | 67 | 98 |
| 23 | 28 | 34 | 67 | 98 |
| 23 | 28 | 34 | 67 | 98 |
| 23 | 28 | 34 | 67 | 98 |
| 23 | 28 | 34 | 67 | 98 |

| | | | | |
|----|----|----|----|----|
| 23 | 28 | 34 | 67 | 98 |
| 23 | 28 | 34 | 67 | 98 |
| 23 | 28 | 34 | 67 | 98 |
| 23 | 28 | 34 | 67 | 98 |

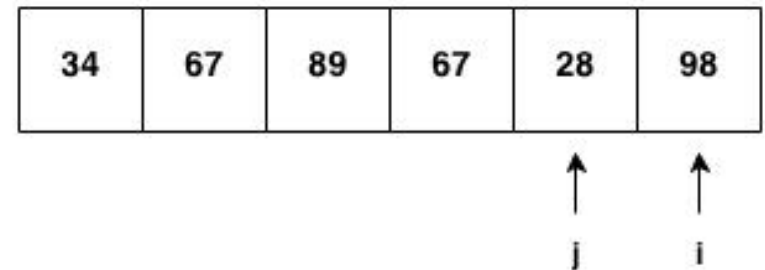
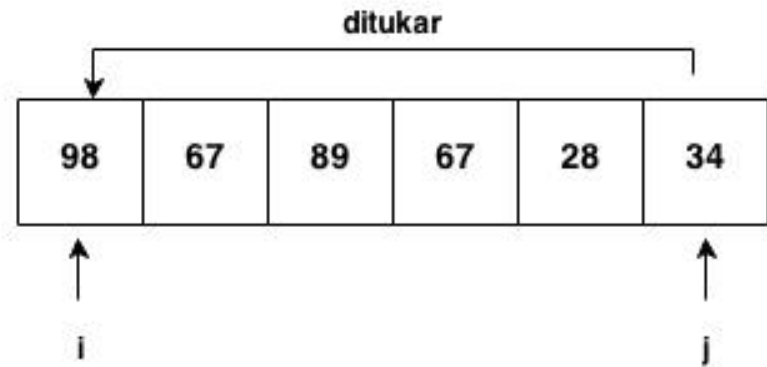
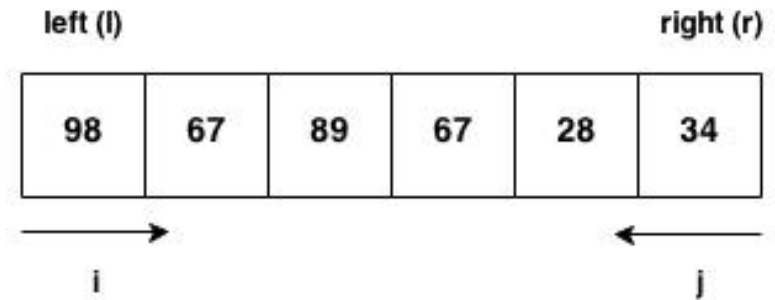
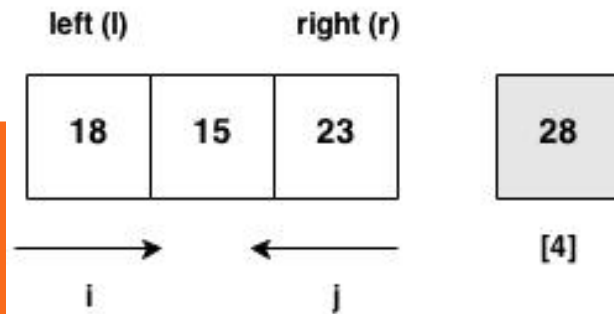
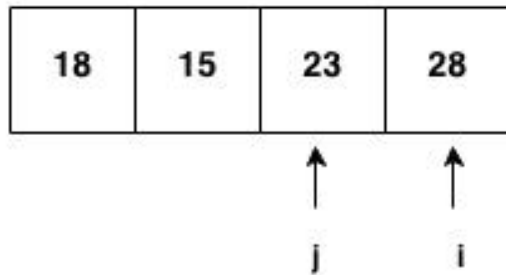
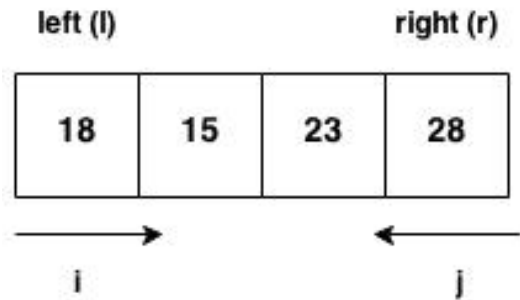
BUBBLE SORT (2)

```
int main(){  
    int tabInt[5] = {34, 67, 23, 28, 98};  
    int i;    int temp;    int tukar;  
    do{  
        /*inisialisasi nilai tukar sebelum ada pertukaran diset false */  
        tukar = 0;  
        /*pengulangan dan memeriksa apakah ada pertukaran */  
        for(i=0; i<(5-1); i++){  
            /*jika ada nilai yang dipertukarkan */  
            if(tabInt[i] > tabInt[i+1]){  
                /* menukar posisi elemen */  
                temp = tabInt[i];  
                tabInt[i] = tabInt[i+1];  
                tabInt[i+1] = temp;  
                tukar = 1;  
            }  
        }  
    }while(tukar == 1);  
    return 0;  
}
```

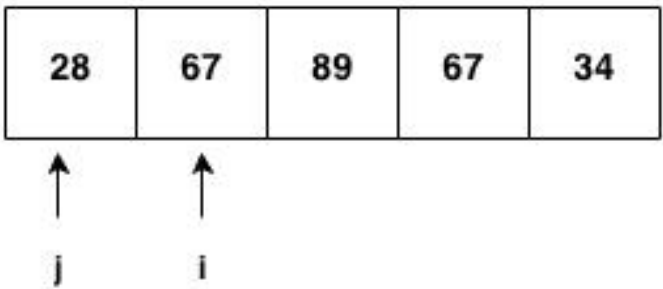
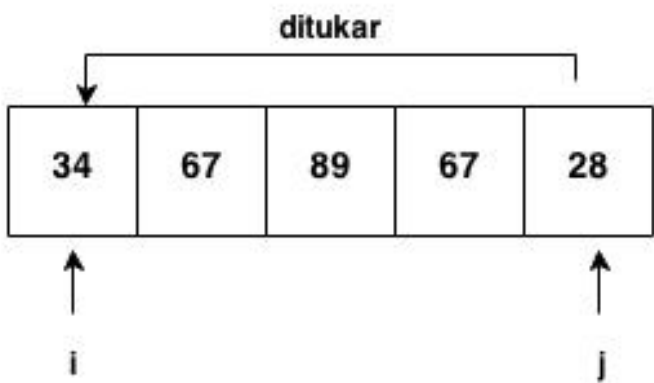
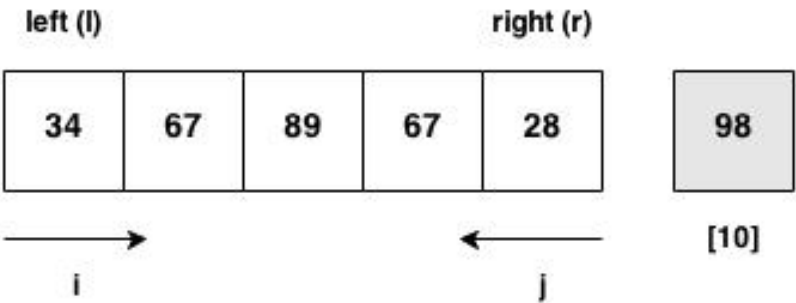
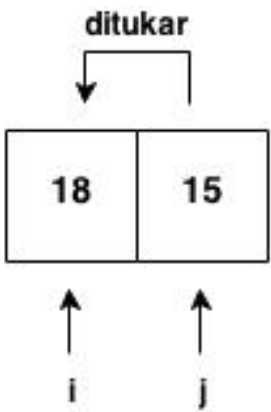
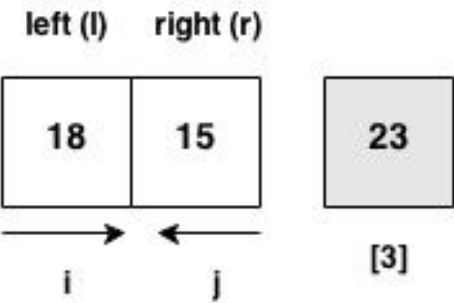
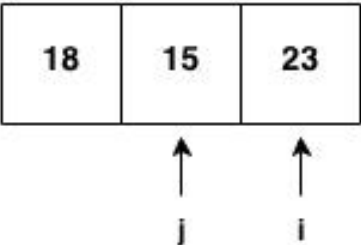
QUICK SORT (1)



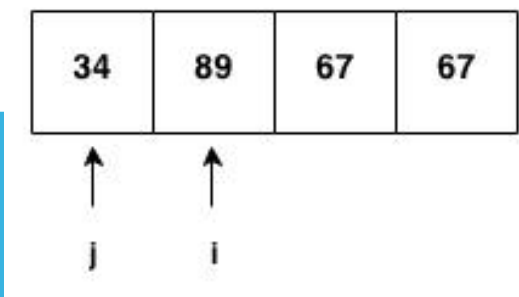
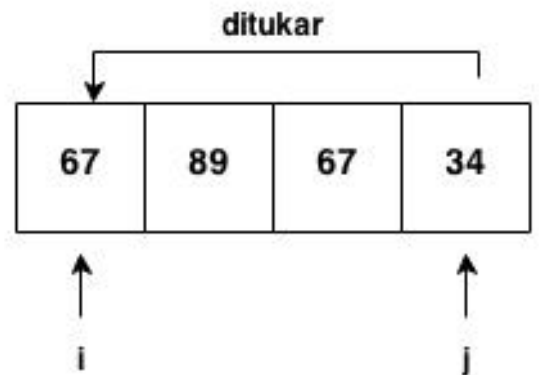
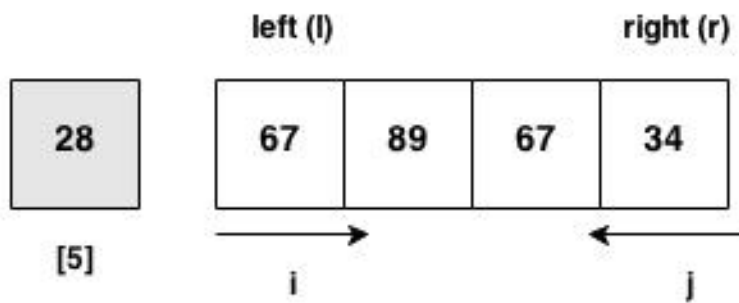
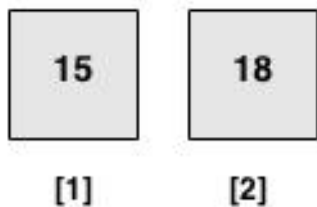
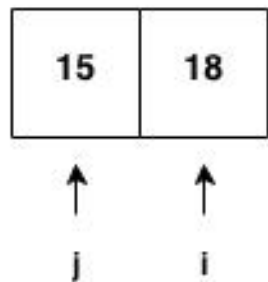
QUICK SORT (1-2)



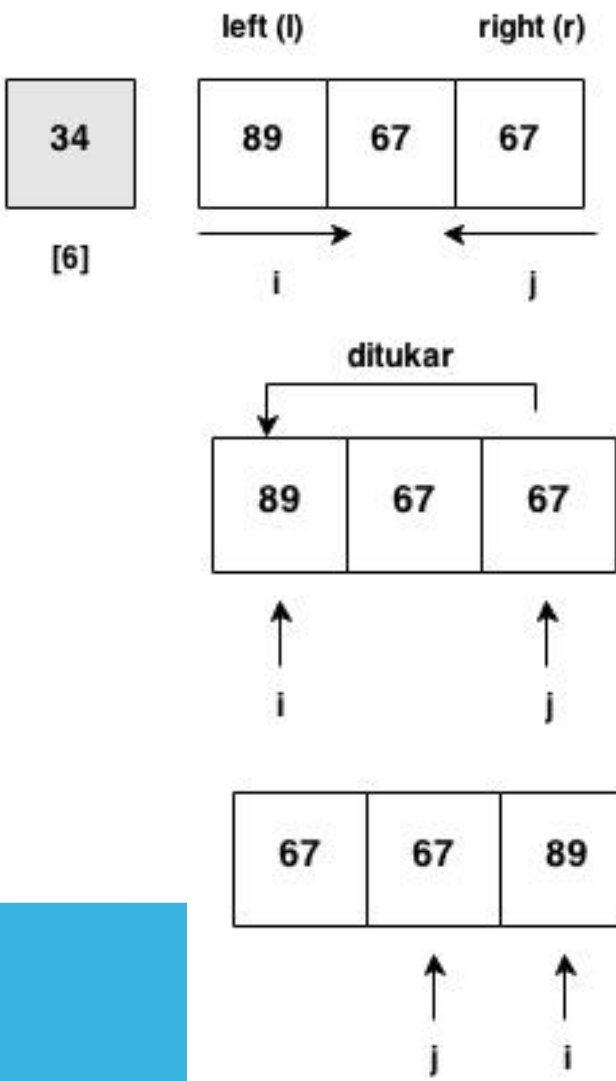
QUICK SORT (1-3)



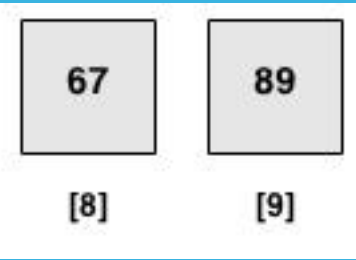
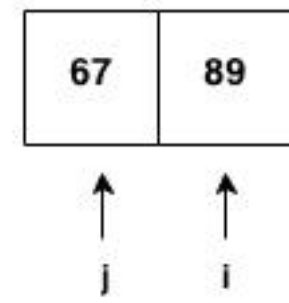
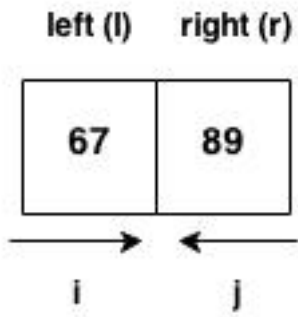
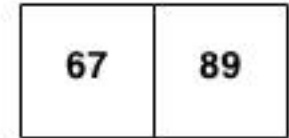
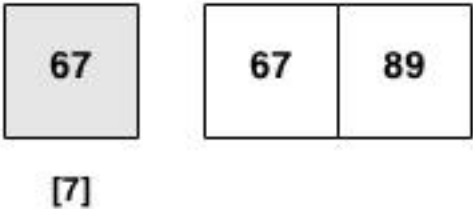
QUICK SORT (1-4)



QUICK SORT (1-5)



QUICK SORT (1-6)



QUICK SORT (1-7)

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| 15 | 18 | 23 | 28 | 28 | 34 | 67 | 67 | 89 | 98 |
| [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] |

QUICK SORT (2) - PIVOT PINGGIR

```
void quickSortPivotPinggir(int l,
    int r){
    int i;    int j;    int temp;
    i = l;    j = r;
    do{
        while((tabInt[i] < tabInt[r])
            && (i<=j)){
            /* mencegah i berjalan
            terlalu jauh melewati batas j
            */
            i = i + 1;
        }
        while((tabInt[j] > tabInt[l])
            && (i<=j)){
            /* mencegah i berjalan terlalu
            jauh melewati batas i */
            j = j - 1;
        }
    }
```

```
    if(i<j){
        temp = tabInt[i];
        tabInt[i] = tabInt[j];
        tabInt[j] = temp;
        i = i + 1;
        j = j - 1;
    }
    while(i<j);
    tulis();

    if(l < j){
        quickSortPivotPinggir(l,j);
    }
    if(i<r){
        quickSortPivotPinggir(i,r);
    }
}
```

QUICK SORT (2-1) - PIVOT PINGGIR

```
int tabInt[10] = {34, 67, 23, 28, 98, 15, 89, 67, 28, 18};
```

```
void tulis(){  
    int i;  
    for(i=0; i<10; i++){  
        printf("%d\n", tabInt[i]);  
    }  
}
```

.....

```
int main(){  
    tulis();  
    quickSortPivotPinggir(0, 9);  
    return 0;  
}
```

QUICK SORT (3) - PIVOT TENGAH

```
void quickSortPivotTengah(int l, int r){
    int i;    int j;    int temp;
    i = l;    j = r;
    int pivot = tabInt[(l+r)/2];
    do{
        while((tabInt[i] < pivot) && (i<=j)){
            /* mencegah i berjalan terlalu jauh
            melewati batas j */
            i = i + 1;
        }
        while((tabInt[j] > pivot) && (i<=j)){
            /* mencegah i berjalan terlalu jauh
            melewati batas i */
            j = j - 1;
        }
        if(i<j){
            temp = tabInt[i];
            tabInt[i] = tabInt[j];
```

```
            tabInt[j] = temp;
            i = i + 1;
            j = j - 1;
        }
    }while(i<j);
    tulis();
    if((l < j) && (j < r) && (i !=
    j)){
        /* karena ada kemungkinan i = j
        berhentinya, yang artinya
        harusnya sudah tidak perlu lagi
        dilakukan proses pengurutan 2
        kali, karena sudah dijalankan
        di rekursif bagian bawah */
        quickSortPivotTengah(l,j);
    }
    if((i < r) && (i > l)){
        /*karena ada kemungkinan
        ternyata i sama dengan l yang
        artinya bahkan i tidak berjalan
        sama sekali kedepan */
        quickSortPivotTengah(i,r);
    }
}
```

QUICK SORT (3-1) - PIVOT TENGAH

```
int tabInt[10] = {34, 67, 23, 28, 98, 15, 89, 67, 28, 18};
```

```
void tulis(){  
    int i;  
    for(i=0; i<10; i++){  
        printf("%d\n", tabInt[i]);  
    }  
}
```

.....

```
int main(){  
    tulis();  
    quickSortPivotTengah(0, 9);  
    return 0;  
}
```

HASIL SORTING QUICKSORT

```
int tabInt[10] = {10,9,8,7,6,5,4,3,2,1};
```

pivot pinggir

10 9 8 7 6 5 4 3 2 1

1 9 8 7 6 5 4 3 2 10

1 9 8 7 6 5 4 3 2 10

1 2 8 7 6 5 4 3 9 10

1 2 8 7 6 5 4 3 9 10

1 2 3 7 6 5 4 8 9 10

1 2 3 7 6 5 4 8 9 10

1 2 3 4 6 5 7 8 9 10

1 2 3 4 6 5 7 8 9 10

pivot tengah

1 2 3 4 5 6 7 8 9 10

1 2 3 4 5 6 7 8 9 10

1 2 3 4 5 6 7 8 9 10

1 2 3 4 5 6 7 8 9 10

1 2 3 4 5 6 7 8 9 10

1 2 3 4 5 6 7 8 9 10

1 2 3 4 5 6 7 8 9 10

HASIL SORTING QUICKSORT (2)

int tabInt[10] = {20,21,34,54,55,12,16,28,98,97};

pivot pinggir

34 67 23 28 98 15 89 67 28 18
18 15 23 28 98 67 89 67 28 34
15 18 23 28 98 67 89 67 28 34
15 18 23 28 34 67 89 67 28 98
15 18 23 28 28 67 89 67 34 98
15 18 23 28 28 34 89 67 67 98
15 18 23 28 28 34 67 67 89 98

pivot tengah

20 21 34 54 28 12 16 55 98 97
20 21 34 16 28 12 54 55 98 97
20 21 12 16 28 34 54 55 98 97
12 21 20 16 28 34 54 55 98 97
12 16 20 21 28 34 54 55 98 97
12 16 20 21 28 34 54 55 98 97
12 16 20 21 28 34 54 55 98 97
12 16 20 21 28 34 54 55 97 98
12 16 20 21 28 34 54 55 97 98

HASIL SORTING QUICKSORT (3)

int tabInt[10] = {34, 67, 23, 28, 98, 15, 89, 67, 28, 18};

pivot pinggir

34 67 23 28 98 15 89 67 28 18
18 15 23 28 98 67 89 67 28 34
15 18 23 28 98 67 89 67 28 34
15 18 23 28 34 67 89 67 28 98
15 18 23 28 28 67 89 67 34 98
15 18 23 28 28 34 89 67 67 98
15 18 23 28 28 34 67 67 89 98

pivot tengah

34 67 23 28 18 15 89 67 28 98
15 18 23 28 67 34 89 67 28 98
15 18 23 28 67 34 89 67 28 98
15 18 23 28 28 34 89 67 67 98
15 18 23 28 28 34 67 67 89 98
15 18 23 28 28 34 67 67 89 98

DAFTAR PUSTAKA

S, Rosa A. dan M. Shalahuddin. 2010. Modul Pembelajaran: Algoritma dan Pemrograman. Modula: Bandung.

