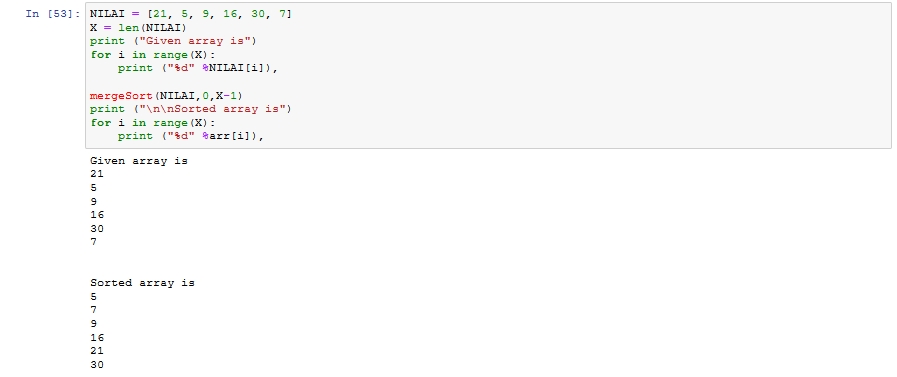
**TUGAS DAA WEEK 5**

**Jonathan(2019071044)**

1. Screenshot output code



1. Source code

Mergesort

def merge(arr, l, m, r):

n1 = m - l + 1

n2 = r- m

L = [0] \* (n1)

R = [0] \* (n2)

# Copy data to temp arrays L[] and R[]

for i in range(0 , n1):

L[i] = arr[l + i]

for j in range(0 , n2):

R[j] = arr[m + 1 + j]

i = 0

j = 0

k = l

while i < n1 and j < n2 :

if L[i] <= R[j]:

arr[k] = L[i]

i += 1

else:

arr[k] = R[j]

j += 1

k += 1

while i < n1:

arr[k] = L[i]

i += 1

k += 1

while j < n2:

arr[k] = R[j]

j += 1

k += 1

def mergeSort(arr,l,r):

if l < r:

m = (l+(r-1))//2

mergeSort(arr, l, m)

mergeSort(arr, m+1, r)

merge(arr, l, m, r)

NILAI = [21, 5, 9, 16, 30, 7]

X = len(NILAI)

print ("Given array is")

for i in range(X):

print ("%d" %NILAI[i]),

mergeSort(NILAI,0,X-1)

print ("\n\nSorted array is")

for i in range(X):

print ("%d" %arr[i]),

**Quicksort**

**def partition(array, low, high):**

**pivot = array[high]**

**i = low - 1**

**for j in range(low, high):**

**if array[j] <= pivot:**

**i = i + 1**

**(array[i], array[j]) = (array[j], array[i])**

**(array[i + 1], array[high]) = (array[high], array[i + 1])**

**return i + 1**

**def quickSort(array, low, high):**

**if low < high:**

**pi = partition(array, low, high)**

**quickSort(array, low, pi - 1)**

**quickSort(array, pi + 1, high)**

**nilai = [17, 12, 57, 8, 16, 25]**

**size = len(nilai)**

**quickSort(nilai, 0, size - 1)**

**print(nilai)**