Nama : Anang Nugraha

NIM : 09021381722106

KELAS : TIBIL5B

$$F_2 = \frac{1}{4}\sum_{x=0}^{3} f_x e^{-i.2.2px/4} = \frac{1}{4}(f_0 e^0 + f_1 e^{-ip} + f_2 e^{-i2p} + f_3 e^{-i3p})$$

$$= \frac{1}{4}(2 + 3[\cos(p) - i\,sin(p)] + 4[\cos(2p) - i\,sin(2p)] + 4[\cos(3p) - i\,sin(3p)])$$

$$= \frac{1}{4}(2 + 3[-1 - 0] + 4[1 - 0] + 4[-1 - 0\,]) = \frac{1}{4}(2 - 3 + 4 - 4) = \frac{1}{4}(-1) = -\frac{1}{4}$$

$$F_3 = \frac{1}{4}\sum_{x=0}^{3} f_x e^{-i.3.2px/4} = \frac{1}{4}\left(f_0 e^0 + f_1 e^{-i3p/2} + f_2 e^{-i3p} + f_3 e^{-i9p/2}\right)$$

$$= \frac{1}{4}(2 + 3[cos(3p/2) - i\,sin(3p/2)] + 4[cos(3p) - i\,sin(3p)] + 4[cos(9p/2) - i\,sin(9p/2)])$$

$$= \frac{1}{4}(2 + 3[0 + i] + 4[-1 - 0] + 4[0 - i\,]) = \frac{1}{4}(2 + 3i - 4 - 4i) = \frac{1}{4}(-2 - i) = -\frac{1}{4}(2 + i)$$

```
1.  package citra_tfd;
2.
3.  import java.util.Arrays;
4.  public class Citra_tfd {
5.      public static void main(String[] args) {
6.          int N = 4;
7.          int f1[] = {2, 3, 4, 4};
8.          double aI[] = new double[N];
9.          double aR[] = new double[N];
10.         double afbilimag[] = new double[N];
11.         double afreal[] = new double[N];
12.         //print f
13.         System.out.println(Arrays.toString(f1));
14.         tfd(N, f1, aR, aI);
```

```
15.     //printarrayR-I
16.     System.out.println(Arrays.toString(aR));
17.     System.out.println(Arrays.toString(aI));
18.     tfd_b(N, aR, aI, afreal, afbilimag);
19.     //printarrayrealdanimag
20.     System.out.println(Arrays.toString(afreal));
21.     System.out.println(Arrays.toString(afbilimag));
22.   }
23.
24.   public static void tfd (int N, int f[], double R[], double I[]) {
25.     int i, k;
26.     double tetha;
27.     for (i = 0; i < N; i++) {
28.       R[i] = 0.0;
29.       I[i] = 0.0;
30.     }
31.     for (k = 0; k < N; k++) {
32.       for (i = 0; i < N; i++) {
33.         tetha = k * 2 * 3.14 * i / (double) N;
34.         R[k] = R[k] + (f[i] * Math.cos(tetha)) / (double) N;
35.         I[k] = I[k] - (f[i] * Math.sin(tetha)) / (double) N;
36.       }
37.     }
38.   }
39.
40.   public static void tfd_b (int N, double R[], double I[], double fReal[], double fImag[]) {
41.     int j, k;
42.     double tet, eps = 1E-12;
43.     for (j = 0; j < N; j++) {
44.       fReal[j] = 0.0;
45.       fImag[j] = 0.0;
46.     }
47.     for (k = 0; k < N; k++) {
48.       for (j = 0; j < N; j++) {
49.         tet = k * 2 * 3.14 * j / (double) N;
50.         fReal[k] = fReal[k] + (R[j] * Math.cos(tet) - I[j] * Math.sin(tet));
51.         fImag[k] = fImag[k] + (I[j] * Math.cos(tet) + R[j] * Math.sin(tet));
52.       }
53.       if (fImag[k] < eps) {
54.         fImag[k] = 0;
55.       }
56.     }
57.   }
58.
```

59.
60. }\

Screen Shot :