

Nama & NPM	Topik:	Tanggal:
1. Sevi Rina Pertiwi G1F024044 2. Anisa Jayanti G1F024046 3. Meida Dinafani G1F024058	Kelas, Objek dan Method	14 September 2024

Tugas Kelompok :

[No. 1] Identifikasi Masalah:

- 1) Apabila diketahui kelas induk adalah Mahasiswa dan Kelas anak adalah turunan dari mahasiswa maka:
 - (a) Analisa atribut, method, dan constructor dari kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!
 - (b) Evaluasi perbedaan kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!
 - (c) Rekomendasi atribut, method, dan constructor yang bisa digunakan bersama kelas induk dan kelas anak!
 - (d) Desain kode program Java yang berisi atribut, method, dan constructor dari kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!

Jawaban:

- (a) Atribut, method, dan constructor dari kelas induk Mahasiswa

Atribut :

Kelas Mahasiswa memiliki atribut Nama, NPM, Prodi, Tahun Angkatan, Asal Universitas.

Method :

Kelas Mahasiswa memiliki beberapa method yaitu :

void nama(String a): Menampilkan nama mahasiswa.

void npm(String a): Menampilkan NPM (nomor pokok mahasiswa).

void prodi(String a): Menampilkan program studi.

void tahunAngkatan(String a): Menampilkan tahun angkatan.

void asalUniversitas(String a): Menampilkan asal universitas.

void sukaBelajar(String a): Menampilkan minat belajar mahasiswa.

Constructor :

Tidak ada konstruktor yang didefinisikan dalam kelas-kelas tersebut. Secara default, konstruktor tanpa argumen disediakan.

Atribut, method, dan constructor dari kelas Anak (turunan dari Mahasiswa) yaitu Mahasiswa Baru :

Atribut :

Menggunakan atribut dari kelas induk Mahasiswa.

Method :

Menggunakan method yang sama dari kelas induk dengan isi variabel yang berbeda.

Constructor :

Constructor kelas Anak memanggil constructor kelas induk untuk menginisialisasi atribut.

- (b) Evaluasi perbedaan kelas induk Mahasiswa dengan kelas Anak (turunan dari Mahasiswa) yaitu :

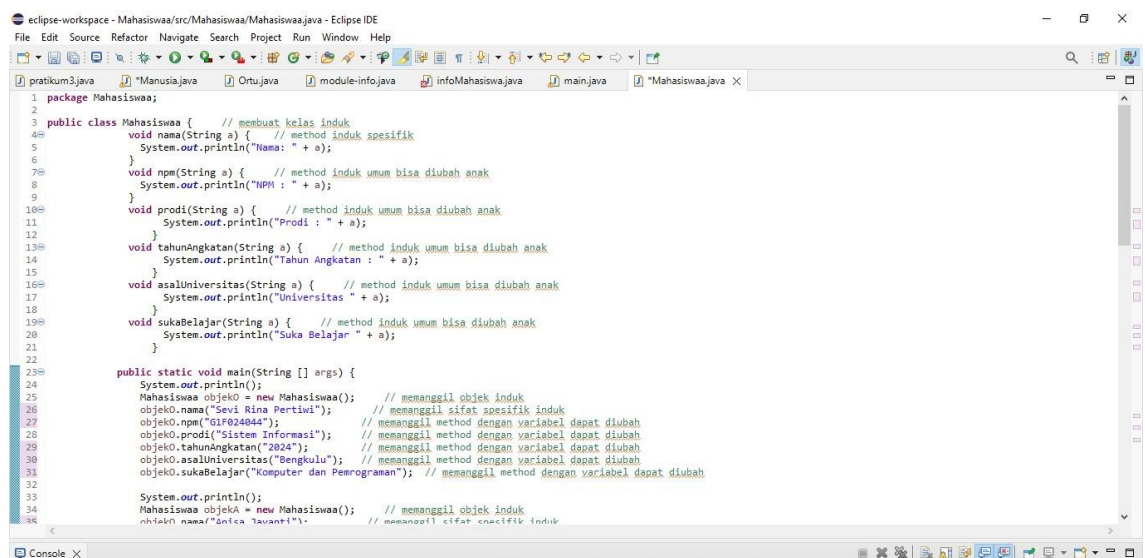
Kelas Mahasiswa berfungsi sebagai kelas induk yang mengandung method yang dapat dioverride oleh kelas anak (turunan dari mahasiswa) untuk perilaku yang lebih spesifik. Sedangkan, Kelas Anak (turunan dari Mahasiswa) yaitu kelas MahasiswaBaru dan MahasiswaBaru2, keduanya mewarisi dari Mahasiswa dan mengoverride methodnya. Mereka tidak menambahkan atribut atau method baru dan hanya mengurangi fungsionalitas dari kelas induk.

- (c) Atribut yang bisa digunakan bersama untuk kelas induk dan kelas anak yaitu nama, npm dan prodi karena setiap kelas induk mahasiswa dan kelas anak (mahasiswa baru) pasti memiliki atribut tersebut.

Method yang dapat digunakan bersama yaitu suka belajar.

Constructor yang dapat digunakan bersama yaitu dengan cara implementasi constructor di Mahasiswa yang menginisialisasi atribut Subkelas bisa memanggil constructor ini dengan cara menggunakan super().

- (d) Kami mendesain program seperti berikut :



```
1 package Mahasiswa;
2
3 public class Mahasiswa { // membuat kelas induk
4     void nama(String a) { // method induk spesifik
5         System.out.println("Nama: " + a);
6     }
7     void npm(String a) { // method induk umum bisa diubah anak
8         System.out.println("NPM : " + a);
9     }
10    void prodi(String a) { // method induk umum bisa diubah anak
11        System.out.println("Prodi : " + a);
12    }
13    void tahunAngkatan(String a) { // method induk umum bisa diubah anak
14        System.out.println("Tahun Angkatan : " + a);
15    }
16    void asalUniversitas(String a) { // method induk umum bisa diubah anak
17        System.out.println("Universitas " + a);
18    }
19    void sukaBelajar(String a) { // method induk umum bisa diubah anak
20        System.out.println("Suka Belajar " + a);
21    }
22
23    public static void main(String [] args) {
24        System.out.println();
25        Mahasiswa objek0 = new Mahasiswa(); // memanggil objek induk
26        objek0.nama("Sevi Rina Pertiwi"); // memanggil sifat spesifik induk
27        objek0.npm("1017024044"); // memanggil method dengan variabel dapat diubah
28        objek0.prodi("Sistem Informasi"); // memanggil method dengan variabel dapat diubah
29        objek0.tahunAngkatan("2024"); // memanggil method dengan variabel dapat diubah
30        objek0.asalUniversitas("Bengkulu"); // memanggil method dengan variabel dapat diubah
31        objek0.sukaBelajar("Komputer dan Pemrograman"); // memanggil method dengan variabel dapat diubah
32
33        System.out.println();
34        Mahasiswa objekA = new Mahasiswa(); // memanggil objek induk
35        objekA.nama("Antea Javanti"); // memanggil sifat spesifik induk
36    }
37 }
```

```
eclipse-workspace - Mahasiswaa/src/Mahasiswaa/Mahasiswaa.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

praktikum3.java Manusia.java Ortu.java module-info.java infoMahasiswaa.java main.java Mahasiswaa.java

33 System.out.println();
34 Mahasiswaa objekA = new Mahasiswaa(); // memanggil objek induk
35 objekA.nama("Anisa Jayanti"); // memanggil sifat spesifik induk
36 objekA.npm("G1F024046"); // memanggil method dengan variabel dapat diubah
37 objekA.prodi("Sistem Informasi"); // memanggil method dengan variabel dapat diubah
38 objekA.tahunAngkatan("2024"); // memanggil method dengan variabel dapat diubah
39 objekA.asalUniversitas("Bengkulu"); // memanggil method dengan variabel dapat diubah
40 objekA.sukaBelajar("Manajemen dan Kepemimpinan"); // memanggil method dengan variabel dapat diubah
41
42 System.out.println();
43 Mahasiswaa objekB = new Mahasiswaa(); // memanggil objek induk
44 objekB.nama("Meida Dinafani"); // memanggil sifat spesifik induk
45 objekB.npm("G1F024058"); // memanggil method dengan variabel dapat diubah
46 objekB.prodi("Sistem Informasi"); // memanggil method dengan variabel dapat diubah
47 objekB.tahunAngkatan("2024"); // memanggil method dengan variabel dapat diubah
48 objekB.asalUniversitas("Bengkulu"); // memanggil method dengan variabel dapat diubah
49 objekB.sukaBelajar("Algoritma dan Struktur Data"); // memanggil method dengan variabel dapat diubah
50
51 }
52
53 class MahasiswaaBaru extends Mahasiswaa {
54     void nama(String A) { // method induk spesifik
55         System.out.println("Nama : " + A); // method induk spesifik
56     }
57     void npm(String A) { // method induk umum bisa diubah anak
58         System.out.println("NPM : " + A);
59     }
60     void prodi(String A) { // method induk umum bisa diubah anak
61         System.out.println("Prodi : " + A);
62     }
63     void tahunAngkatan(String A) { // method induk umum bisa diubah anak
64         System.out.println("Tahun Angkatan : " + A);
65     }
66     void asalUniversitas(String A) { // method induk umum bisa diubah anak
67         System.out.println("Universitas : " + A);
68     }
69     void sukaBelajar(String A) { // method induk umum bisa diubah anak
70         System.out.println("Suka Belajar : " + A);
71     }
72 }
73
74 class MahasiswaaBaru2 extends Mahasiswaa {
75     void nama(String B) { // method induk spesifik
76         System.out.println("Nama : " + B);
77     }
78     void npm(String B) { // method induk umum bisa diubah anak
79         System.out.println("NPM : " + B);
80     }
81     void prodi(String B) { // method induk umum bisa diubah anak
82         System.out.println("Prodi : " + B);
83     }
84     void tahunAngkatan(String B) { // method induk umum bisa diubah anak
85         System.out.println("Tahun Angkatan : " + B);
86     }
87     void asalUniversitas(String B) { // method induk umum bisa diubah anak
88         System.out.println("Universitas : " + B);
89     }
90     void sukaBelajar(String B) { // method induk umum bisa diubah anak
91         System.out.println("Suka Belajar : " + B);
92     }
93 }
94
95 }
96
97
98
99
100
```

```
eclipse-workspace - Mahasiswaa/src/Mahasiswaa/Mahasiswaa.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

praktikum3.java Manusia.java Ortu.java module-info.java infoMahasiswaa.java main.java Mahasiswaa.java

66 void asalUniversitas(String A) { // method induk umum bisa diubah anak
67     System.out.println("Universitas : " + A);
68 }
69 void sukaBelajar(String A) { // method induk umum bisa diubah anak
70     System.out.println("Suka Belajar : " + A);
71 }
72 }
73
74 class MahasiswaaBaru2 extends Mahasiswaa {
75     void nama(String B) { // method induk spesifik
76         System.out.println("Nama : " + B);
77     }
78     void npm(String B) { // method induk umum bisa diubah anak
79         System.out.println("NPM : " + B);
80     }
81     void prodi(String B) { // method induk umum bisa diubah anak
82         System.out.println("Prodi : " + B);
83     }
84     void tahunAngkatan(String B) { // method induk umum bisa diubah anak
85         System.out.println("Tahun Angkatan : " + B);
86     }
87     void asalUniversitas(String B) { // method induk umum bisa diubah anak
88         System.out.println("Universitas : " + B);
89     }
90     void sukaBelajar(String B) { // method induk umum bisa diubah anak
91         System.out.println("Suka Belajar : " + B);
92     }
93 }
94
95 }
96
97
98
99
100
```

```
eclipse-workspace - Mahasiswaa/src/Mahasiswaa/Mahasiswaa.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

praktikum3.java Manusia.java Ortu.java module-info.java infoMahasiswaa.java main.java Mahasiswaa.java

66 void asalUniversitas(String A) { // method induk umum bisa diubah anak
67     System.out.println("Universitas : " + A);
68 }
69 void sukaBelajar(String A) { // method induk umum bisa diubah anak
70     System.out.println("Suka Belajar : " + A);
71 }
72 }
73
74 class MahasiswaaBaru2 extends Mahasiswaa {
75     void nama(String B) { // method induk spesifik
76         System.out.println("Nama : " + B);
77     }
78     void npm(String B) { // method induk umum bisa diubah anak
79         System.out.println("NPM : " + B);
80     }
81     void prodi(String B) { // method induk umum bisa diubah anak
82         System.out.println("Prodi : " + B);
83     }
84     void tahunAngkatan(String B) { // method induk umum bisa diubah anak
85         System.out.println("Tahun Angkatan : " + B);
86     }
87     void asalUniversitas(String B) { // method induk umum bisa diubah anak
88         System.out.println("Universitas : " + B);
89     }
90     void sukaBelajar(String B) { // method induk umum bisa diubah anak
91         System.out.println("Suka Belajar : " + B);
92     }
93 }
94
95 }
96
97
98
99
100
```

```
<terminated> Mahasiswaa [Java Application] C:\Users\CACCC\p2\poo\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.17.0.12.v20240802-1518\jre\bin\javaw.exe (Sep 18, 2024, 12:17:05 AM - 12:17:08 AM) [pid: 6204]

Nama: Sevi Rina Pertwi
NPM : G1F024044
Prodi : Sistem Informasi
Tahun Angkatan : 2024
Universitas Bengkulu
Suka Belajar Komputer dan Pemrograman

Nama: Anisa Jayanti
NPM : G1F024046
Prodi : Sistem Informasi
Tahun Angkatan : 2024
Universitas Bengkulu
Suka Belajar Manajemen dan Kepemimpinan

Nama: Meida Dinafani
NPM : G1F024058
Prodi : Sistem Informasi
Tahun Angkatan : 2024
Universitas Bengkulu
Suka Belajar Algoritma dan Struktur Data

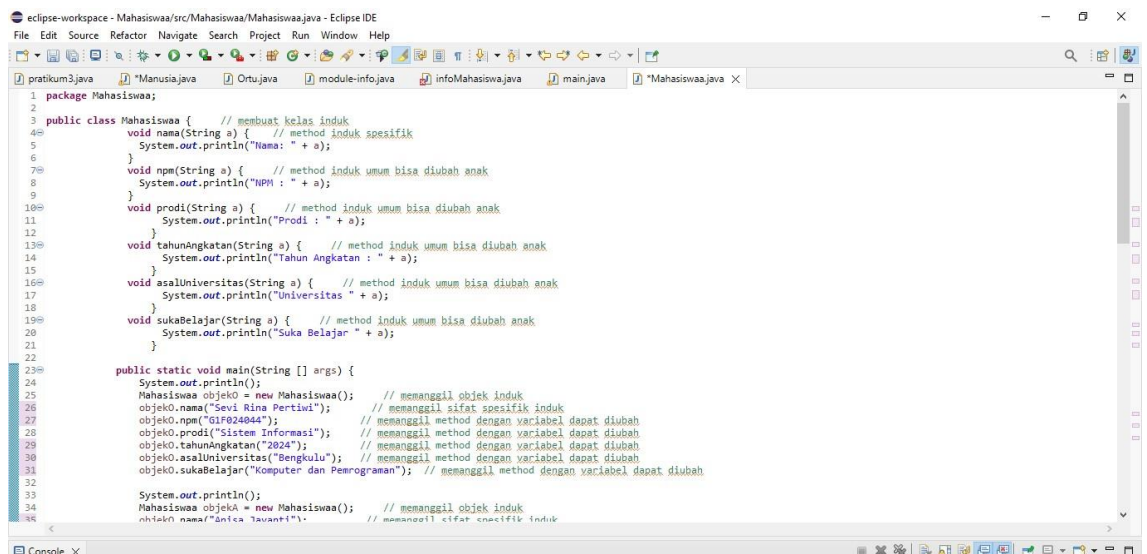
Writeable Smart Insert 66 : 89 : 3592
```

[No.1] Analisis dan Argumentasi

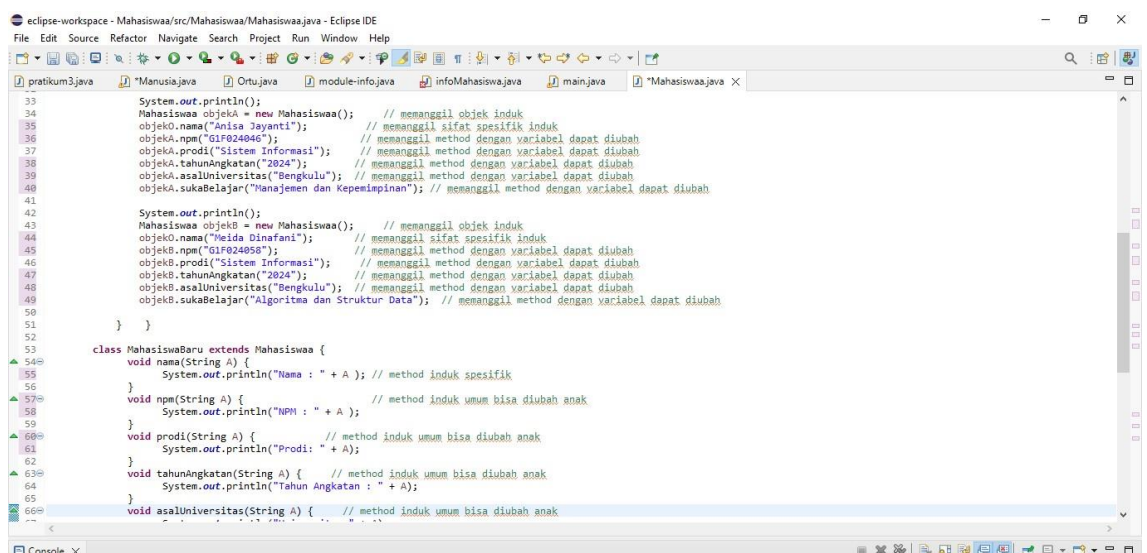
- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara membuat kode program yang berisi atribut, method, dan constructor dari kelas induk Mahasiswa dan kelas Anak (turunan dari Mahasiswa).
- 2) Alasan solusi ini karena kode program sesuai dengan permintaan di soal.
- 3) Perbaikan kode program dengan cara menambahkan atribut, method, dan constructor dengan menyesuaikan kelas induk Mahasiswa dengan kelas Anak (turunan dari Mahasiswa) yang diisi dengan data kelompok.

[No.1] Penyusunan Algoritma dan Kode Program

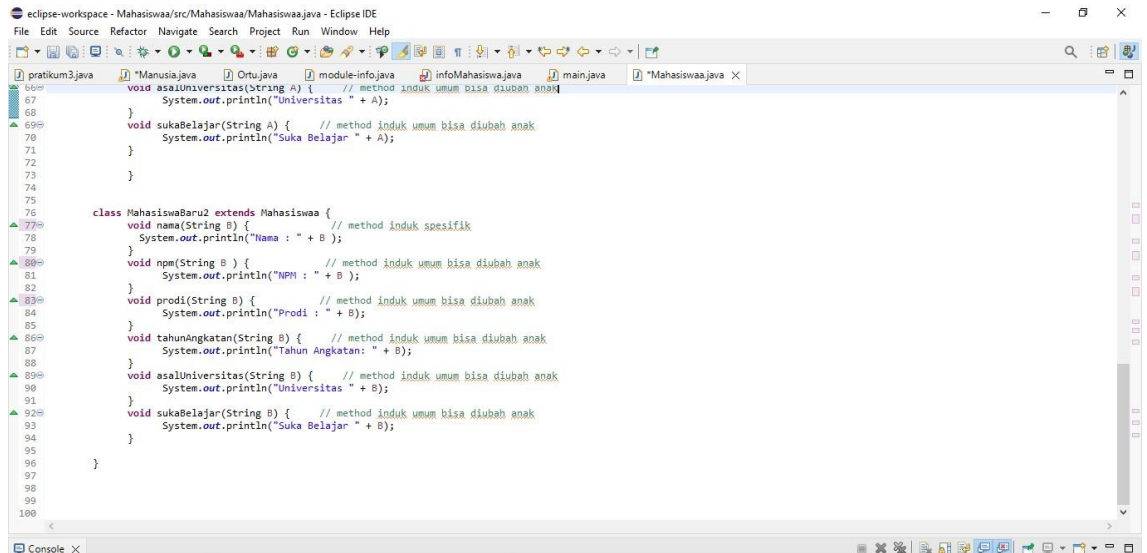
- 1) Algoritma
 - (a) Membuat file project java
 - (b) Membuat public class Mahasiswa
 - (c) Membuat atribut
 - (d) Membuat method dan constructor
 - (e) Membuat kelas anak (mahasiswa baru dan mahasiswa baru 2)
 - (f) Mencetak dan menampilkan isi dari kode program.
- 2) Kode program dan luaran
 - a) Kode program, komentar dan hasil luaran.



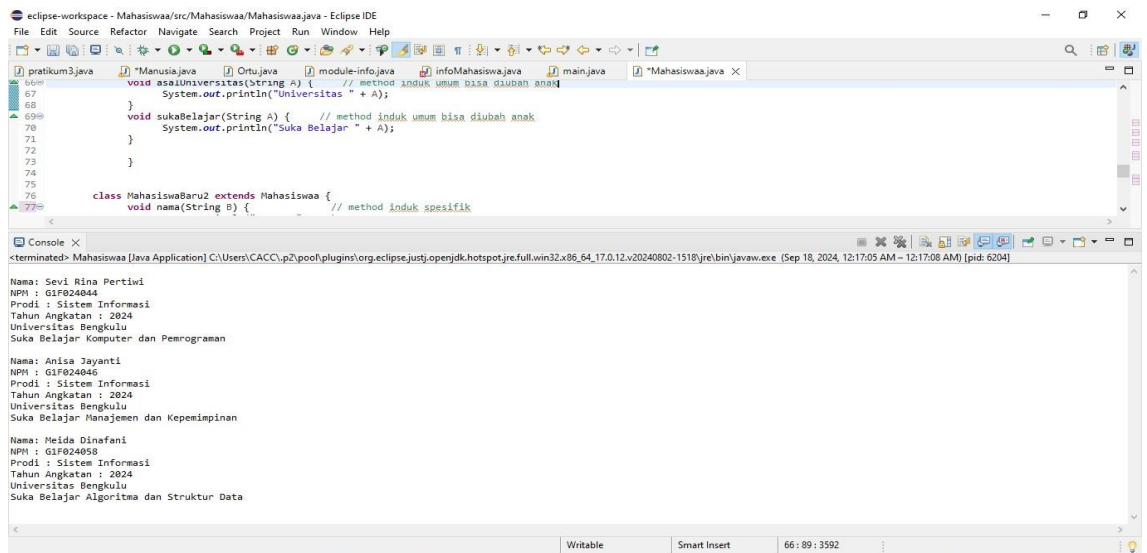
```
1 package Mahasiswa;
2
3 public class Mahasiswa { // membuat kelas induk
4     void nama(String a) { // method induk spesifik
5         System.out.println("Nama: " + a);
6     }
7
8     void npm(String a) { // method induk umum bisa diubah anak
9         System.out.println("NPM : " + a);
10    }
11
12    void prodi(String a) { // method induk umum bisa diubah anak
13        System.out.println("Prodi : " + a);
14    }
15
16    void tahunAngkatan(String a) { // method induk umum bisa diubah anak
17        System.out.println("Tahun Angkatan : " + a);
18    }
19
20    void asalUniversitas(String a) { // method induk umum bisa diubah anak
21        System.out.println("Universitas " + a);
22    }
23
24    void sukaBelajar(String a) { // method induk umum bisa diubah anak
25        System.out.println("Suka Belajar " + a);
26    }
27
28    public static void main(String [] args) {
29        System.out.println();
30        Mahasiswa objek0 = new Mahasiswa(); // memanggil objek induk
31        objek0.nama("Sevi Rina Pertiwi"); // memanggil sifat spesifik induk
32        objek0.npm("G1F024844"); // memanggil method dengan variabel dapat diubah
33        objek0.prodi("Sistem Informasi"); // memanggil method dengan variabel dapat diubah
34        objek0.tahunAngkatan("2024"); // memanggil method dengan variabel dapat diubah
35        objek0.asalUniversitas("Bengkulu"); // memanggil method dengan variabel dapat diubah
36        objek0.sukaBelajar("Komputer dan Pemrograman"); // memanggil method dengan variabel dapat diubah
37
38        System.out.println();
39        Mahasiswa objekA = new Mahasiswa(); // memanggil objek induk
40        objekA.nama("Anisa Jayanti"); // memanggil sifat spesifik induk
41        objekA.npm("G1F024856"); // memanggil method dengan variabel dapat diubah
42        objekA.prodi("Sistem Informasi"); // memanggil method dengan variabel dapat diubah
43        objekA.tahunAngkatan("2024"); // memanggil method dengan variabel dapat diubah
44        objekA.asalUniversitas("Bengkulu"); // memanggil method dengan variabel dapat diubah
45        objekA.sukaBelajar("Manajemen dan Kepemimpinan"); // memanggil method dengan variabel dapat diubah
46
47        System.out.println();
48        Mahasiswa objekB = new Mahasiswa(); // memanggil objek induk
49        objekB.nama("Meida Dinafani"); // memanggil sifat spesifik induk
50        objekB.npm("G1F024858"); // memanggil method dengan variabel dapat diubah
51        objekB.prodi("Sistem Informasi"); // memanggil method dengan variabel dapat diubah
52        objekB.tahunAngkatan("2024"); // memanggil method dengan variabel dapat diubah
53        objekB.asalUniversitas("Bengkulu"); // memanggil method dengan variabel dapat diubah
54        objekB.sukaBelajar("Algoritma dan Struktur Data"); // memanggil method dengan variabel dapat diubah
55
56    }
57
58    class MahasiswaBaru extends Mahasiswa {
59        void nama(String A) { // method induk spesifik
60            System.out.println("Nama : " + A);
61        }
62
63        void npm(String A) { // method induk umum bisa diubah anak
64            System.out.println("NPM : " + A);
65        }
66
67        void prodi(String A) { // method induk umum bisa diubah anak
68            System.out.println("Prodi : " + A);
69        }
70
71        void tahunAngkatan(String A) { // method induk umum bisa diubah anak
72            System.out.println("Tahun Angkatan : " + A);
73        }
74
75        void asalUniversitas(String A) { // method induk umum bisa diubah anak
76            System.out.println("Universitas : " + A);
77        }
78
79        void sukaBelajar(String A) { // method induk umum bisa diubah anak
80            System.out.println("Suka Belajar : " + A);
81        }
82    }
83 }
```



```
33 System.out.println();
34 Mahasiswa objekA = new Mahasiswa(); // memanggil objek induk
35 objekA.nama("Anisa Jayanti"); // memanggil sifat spesifik induk
36 objekA.npm("G1F024846"); // memanggil method dengan variabel dapat diubah
37 objekA.prodi("Sistem Informasi"); // memanggil method dengan variabel dapat diubah
38 objekA.tahunAngkatan("2024"); // memanggil method dengan variabel dapat diubah
39 objekA.asalUniversitas("Bengkulu"); // memanggil method dengan variabel dapat diubah
40 objekA.sukaBelajar("Manajemen dan Kepemimpinan"); // memanggil method dengan variabel dapat diubah
41
42 System.out.println();
43 Mahasiswa objekB = new Mahasiswa(); // memanggil objek induk
44 objekB.nama("Meida Dinafani"); // memanggil sifat spesifik induk
45 objekB.npm("G1F024858"); // memanggil method dengan variabel dapat diubah
46 objekB.prodi("Sistem Informasi"); // memanggil method dengan variabel dapat diubah
47 objekB.tahunAngkatan("2024"); // memanggil method dengan variabel dapat diubah
48 objekB.asalUniversitas("Bengkulu"); // memanggil method dengan variabel dapat diubah
49 objekB.sukaBelajar("Algoritma dan Struktur Data"); // memanggil method dengan variabel dapat diubah
50
51 }
52
53 class MahasiswaBaru extends Mahasiswa {
54     void nama(String A) { // method induk spesifik
55         System.out.println("Nama : " + A);
56     }
57
58     void npm(String A) { // method induk umum bisa diubah anak
59         System.out.println("NPM : " + A);
60     }
61
62     void prodi(String A) { // method induk umum bisa diubah anak
63         System.out.println("Prodi : " + A);
64     }
65
66     void tahunAngkatan(String A) { // method induk umum bisa diubah anak
67         System.out.println("Tahun Angkatan : " + A);
68     }
69
70     void asalUniversitas(String A) { // method induk umum bisa diubah anak
71         System.out.println("Universitas : " + A);
72     }
73
74     void sukaBelajar(String A) { // method induk umum bisa diubah anak
75         System.out.println("Suka Belajar : " + A);
76     }
77 }
78
79 }
```



```
praktikum3.java | *Manusia.java | Ortu.java | module-info.java | InfoMahasiswaa.java | main.java | *Mahasiswaa.java
66  void asalUniversitas(String A) { // method induk umum bisa diubah anak
67      System.out.println("Universitas " + A);
68  }
69  void sukaBelajar(String A) { // method induk umum bisa diubah anak
70      System.out.println("Suka Belajar " + A);
71  }
72  }
73  }
74  }
75  }
76  }
77  class MahasiswaBaru2 extends Mahasiswaa { // method induk spesifik
78      void nama(String B) { // method induk umum bisa diubah anak
79          System.out.println("Nama : " + B);
80      }
81      void npm(String B) { // method induk umum bisa diubah anak
82          System.out.println("NPM : " + B);
83      }
84      void prodi(String B) { // method induk umum bisa diubah anak
85          System.out.println("Prodi : " + B);
86      }
87      void tahunAngkatan(String B) { // method induk umum bisa diubah anak
88          System.out.println("Tahun Angkatan: " + B);
89      }
90      void asalUniversitas(String B) { // method induk umum bisa diubah anak
91          System.out.println("Universitas " + B);
92      }
93      void sukaBelajar(String B) { // method induk umum bisa diubah anak
94          System.out.println("Suka Belajar " + B);
95      }
96  }
97  }
98  }
99  }
100 }
```



```
terminated> Mahasiswaa [Java Application] C:\Users\CACCC\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.12.v20240802-1518\jre\bin\javaw.exe (Sep 18, 2024, 12:17:05 AM - 12:17:08 AM) [pid: 6204]
Nama: Sevi Rina Pertiwi
NPM : G1F024044
Prodi : Sistem Informasi
Tahun Angkatan : 2024
Universitas Bengkulu
Suka Belajar Komputer dan Pemrograman

Nama: Anisa Jayanti
NPM : G1F024046
Prodi : Sistem Informasi
Tahun Angkatan : 2024
Universitas Bengkulu
Suka Belajar Manajemen dan Kepemimpinan

Nama: Meida Dinafani
NPM : G1F024058
Prodi : Sistem Informasi
Tahun Angkatan : 2024
Universitas Bengkulu
Suka Belajar Algoritma dan Struktur Data
```

- b) Output(luaran) sudah benar dan sesuai dengan program yang disusun oleh kami. Atribut, method dan constructor yang ditampilkan telah sesuai dengan kelas induk Mahasiswa dan turunannya. Kami membuat program yang kami ketahui tentang atribut, method, dan constructor.

[No.1] Kesimpulan

1) Analisa

Pada program tersebut kami menggunakan bentuk kelas public dan menambahkan atribut, method dan constructor karena agar program dapat berjalan dengan baik. Agar dapat menampilkan luaran kami perlu menganalisis kebutuhan setiap atribut, method dan constructor dari kelas induk Mahasiswa dan kelas Anak (turunan dari Mahasiswa). Atribut adalah identitas sebuah objek, method adalah sifat perilaku baik (behavior) dan constructor adalah parameter yang berisi tipe data pada setiap atribut.

Refleksi

Pengalaman pada tugas kelompok minggu ini sangat banyak, kami bisa membuat program dengan atribut, method dan constructor dari kelas induk Mahasiswa dan kelas Anak (turunan dari Mahasiswa) meskipun kami masih perlu banyak belajar. Pengetahuan yang kami dapat adalah bagaimana cara penggunaan atribut, method dan constructor pada sebuah program. Tantangan yang dihadapi kelompok kami adalah program terkadang eror karena kurang tanda atau salah perintah di kode. Dari materi soal kelompok kami dapat mengetahui penggunaan atribut, method dan constructor dalam sebuah program.