Nama : Haikal Muhammad Kurniawan

Nim : 20220040008

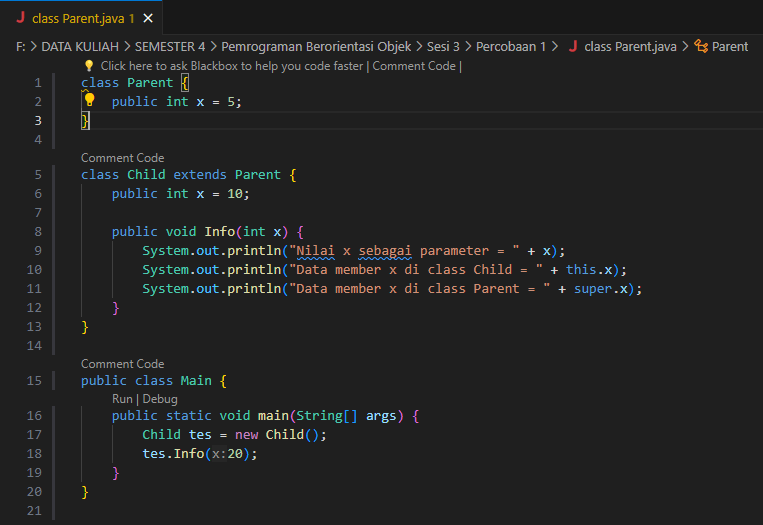
Kelas : TI22M

**Percobaan 1**

Ada beberapa kesalahan dalam kode tersebut:

1. Tidak ada import yang diperlukan.
2. Penulisan "super.x" yang salah pada baris ***System.out.println("Data member x di class Parent = 11 + super.x);***, seharusnya menjadi "super.x" saja tanpa angka 11.
3. Terdapat spasi yang tidak diperlukan sebelum kata "Info" pada baris **tes. Info (20);**.
4. Penamaan kelas "Nilaix" yang sebaiknya diubah menjadi sesuatu yang lebih deskriptif atau relevan.

Berikut adalah perbaikan kode tersebut:



Perubahan yang dilakukan:

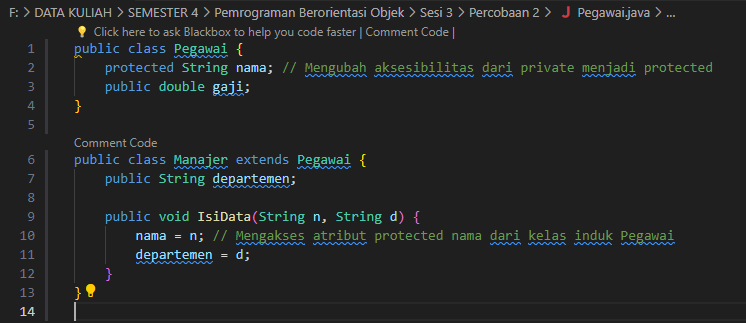
1. Menghilangkan spasi sebelum kata "Info".
2. Mengubah "11 + super.x" menjadi "super.x".
3. Mengubah nama kelas menjadi "Main".
4. Menggunakan kurung kurawal yang lebih konsisten untuk blok kode.

**Percobaan 2**

Kode yang diberikan mengalami kesalahan karena atribut **nama** dari kelas **Pegawai** memiliki akses privat (**private**), sehingga tidak dapat diakses langsung dari kelas turunannya (**Manajer**). Dalam Java, atribut dengan akses privat hanya dapat diakses secara langsung dari dalam kelas yang sama.

Solusinya adalah dengan mengubah aksesibilitas atribut **nama** menjadi **protected** atau **public** agar dapat diakses dari kelas turunannya (**Manajer**).

Berikut adalah kode yang sudah diperbaiki:



Perubahan yang dilakukan:

1. Mengubah aksesibilitas atribut ***nama*** dalam kelas ***Pegawai*** menjadi ***protected***.
2. Mengakses atribut ***nama*** dari kelas turunan ***Manajer*** menggunakan aksesibilitas ***protected*** yang telah diperbaiki.

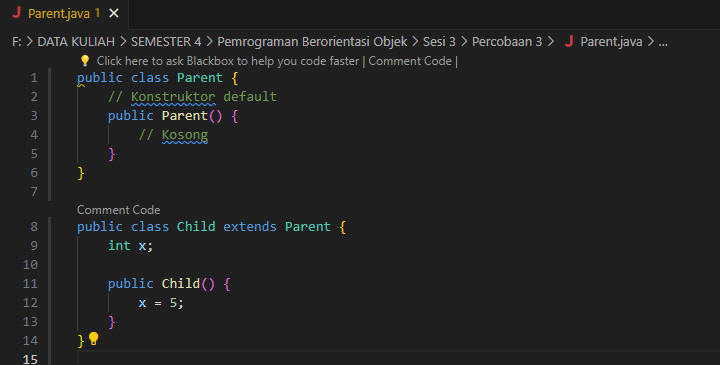
**Percobaan 3**

Dalam Java, ketika membuat kelas turunan ***(child class)***, konstruktor default ***(tanpa parameter)*** dari kelas induk ***(parent class)*** akan dipanggil secara implisit kecuali ada konstruktor yang didefinisikan di kelas induk yang membutuhkan parameter.

Dalam kasus ini, kelas induk *(****Parent****)* tidak memiliki konstruktor default, sehingga ketika kelas turunan *(****Child****)* mencoba membuat konstruktor tanpa parameter, akan terjadi kesalahan karena konstruktor yang diwariskan secara implisit tidak ditemukan.

Solusinya adalah dengan menambahkan konstruktor default (tanpa parameter) di kelas induk *(****Parent****)* atau secara eksplisit memanggil konstruktor kelas induk yang tersedia menggunakan kata kunci ***super()*** dalam konstruktor kelas turunan *(****Child****)*.

Berikut adalah contoh perbaikan dengan menambahkan konstruktor default di kelas induk *(****Parent****)*:



Dalam perbaikan ini, konstruktor default ditambahkan ke kelas **Parent** untuk memastikan bahwa kelas turunan (**Child**) dapat mengaksesnya secara implisit.

**Percobaan 4**

Terdapat beberapa kesalahan:

1. Typo pada pemanggilan konstruktor pada objek ***Utama*** di dalam method ***main***.
2. Pada konstruktor di kelas ***Employee***, terdapat kesalahan penulisan konstanta ***BASE\_SALARY***.
3. Penulisan ***DOB*** pada konstruktor kelas ***Employee*** yang seharusnya ***DoB***.
4. Pada kelas ***TestManager***, pada pemanggilan konstruktor kedua objek ***Utama*** terdapat kesalahan sintaksis.

Terdapat beberapa kesalahan:

1. Typo pada pemanggilan konstruktor pada objek ***Utama*** di dalam method ***main***.
2. Pada konstruktor di kelas ***Employee***, terdapat kesalahan penulisan konstanta ***BASE\_SALARY***.
3. Penulisan ***DOB*** pada konstruktor kelas ***Employee*** yang seharusnya ***DoB***.
4. Pada kelas ***TestManager***, pada pemanggilan konstruktor kedua objek ***Utama*** terdapat kesalahan sintaksis.

Berikut adalah perbaikan kode tersebut:



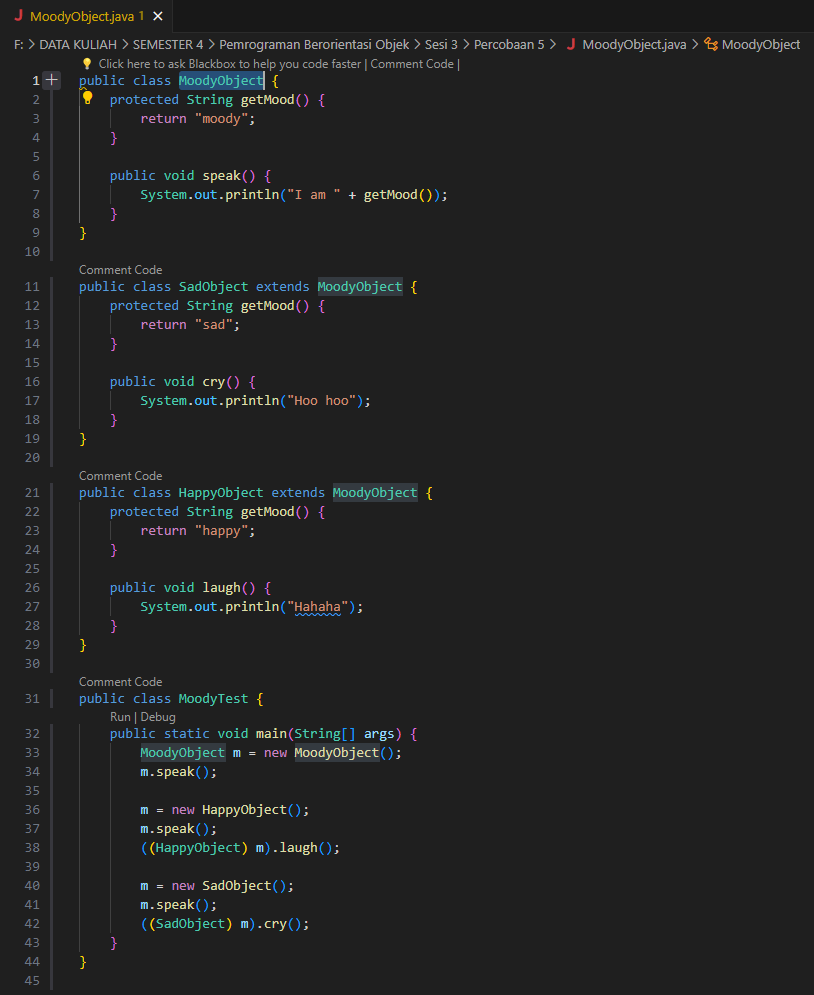
**Perubahan yang dilakukan:**

1. Memperbaiki pemanggilan konstruktor pada objek ***Utama***.
2. Mengubah ***DOB*** menjadi ***DoB*** pada konstruktor kelas ***Employee***.
3. Mengubah ***Utama new Manager("Michael", "Accounting");*** menjadi ***Manager newUtama = new Manager("Michael", "Accounting");*** pada kelas ***TestManager***.

**Percobaan 5**

Terdapat beberapa kesalahan:

1. Metode ***cry()*** tidak ada dalam kelas ***MoodyObject*** tetapi dideklarasikan di luar kelas, sehingga menyebabkan kesalahan kompilasi.
2. Deklarasi metode ***laugh()*** pada kelas ***MoodyObject*** tidak memiliki body metode, dan juga deklarasi tersebut tidak perlu ada di kelas induk karena metode itu hanya ada pada kelas ***HappyObject***.

Berikut adalah perbaikan kode tersebut:

**Perubahan yang dilakukan:**

1. Menghapus deklarasi metode ***cry()*** dan ***laugh()*** dari kelas ***MoodyObject*** karena keduanya hanya ada di kelas turunannya.
2. Mengubah penggunaan objek dan memanggil metode ***laugh()*** dan ***cry()*** sesuai dengan jenis objeknya.

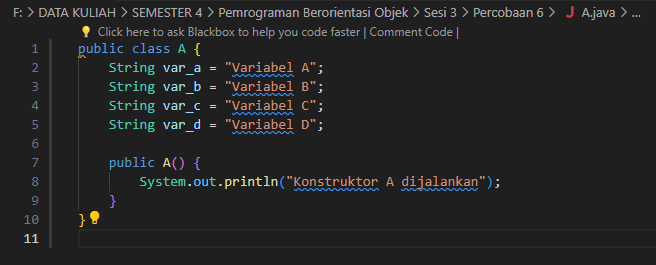
**Percobaan 6**

Terdapat beberapa kesalahan:

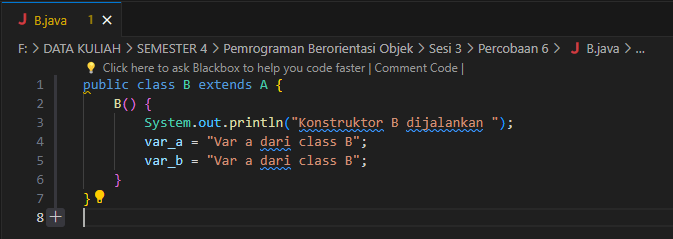
1. Terdapat kesalahan penulisan nama kelas di kelas ***B***, yang seharusnya adalah ***extends A***, bukan ***extends Al***.
2. Terdapat kesalahan penulisan dalam deklarasi variabel pada konstruktor kelas ***B***.
3. Terdapat kesalahan dalam pemanggilan variabel pada method ***main()***, karena variabel yang dipanggil seharusnya dimulai dengan huruf kecil ***var\_a***, ***var\_b***, ***var\_c***, dan ***var\_d***.

Berikut adalah perbaikan dari kode tersebut:

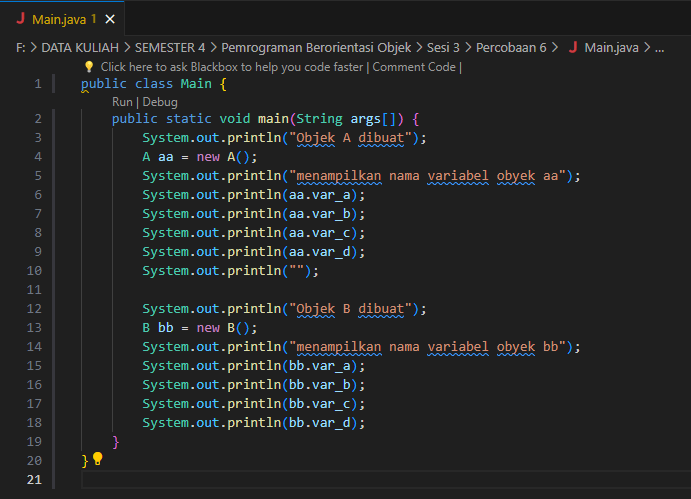
**Kelas A (A.java):**

****

**Kelas B (B.java):**

****

**Main Class (Main.java):**

****

**Perubahan yang dilakukan:**

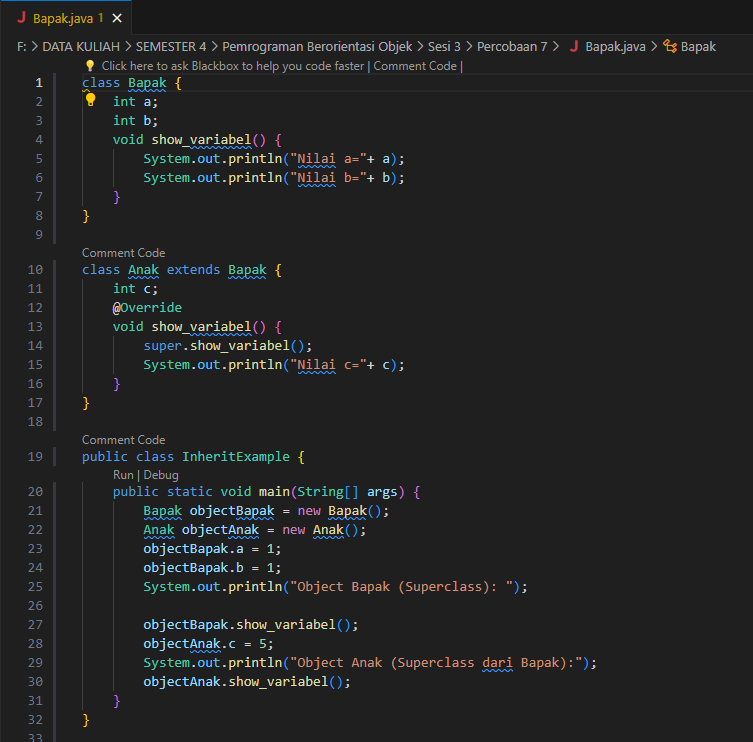
1. Penyesuaian penamaan kelas B yang seharusnya adalah ***extends A***.
2. Penyesuaian penulisan variabel pada konstruktor kelas B agar sesuai dengan sintaks yang benar.
3. Penyesuaian pemanggilan variabel pada method ***main()*** agar sesuai dengan nama variabel yang telah dideklarasikan.

**Percobaan 7**

Terdapat beberapa kesalahan:

1. Terdapat kesalahan dalam penulisan nama method ***show variabel*** dan ***show\_variabel***.
2. Penulisan method ***show\_variabel()*** pada kelas **Anak** tidak menggunakan keyword ***@Override***.
3. Pada method ***show\_variabel()*** pada kelas ***Anak***, nilai variabel ***a*** dan ***b*** tidak diambil dari superclass menggunakan super.
4. Pada pemanggilan method ***show\_variabel()*** pada objek objectAnak, terdapat kesalahan penulisan spasi antara ***show*** dan ***variabel***.

**Berikut adalah perbaikan dari kode tersebut:**

****

**Perubahan yang dilakukan:**

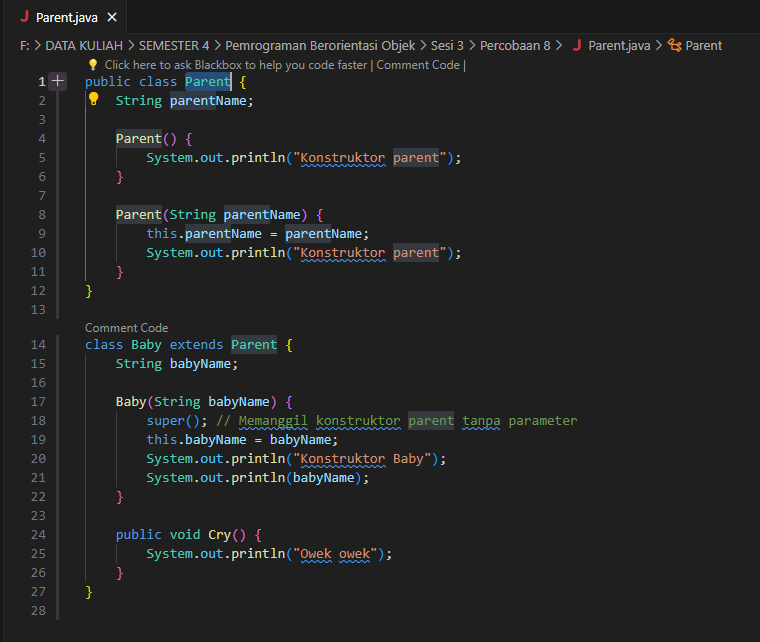
1. Memperbaiki penulisan nama method menjadi ***show\_variabel***.
2. Menambahkan annotation ***@Override*** pada method ***show\_variabel()*** di kelas ***Anak***.
3. Menggunakan ***super.show\_variabel()*** di dalam method ***show\_variabel()*** di kelas Anak untuk memanggil method dari superclass dan menampilkan nilai variabel ***a*** dan ***b***.
4. Memperbaiki penulisan pemanggilan method ***show\_variabel()*** pada objek ***objectAnak***.

**Percobaan 8**

Terdapat beberapa kesalahan:

1. Konstruktor kelas ***Baby*** tidak memanggil konstruktor superclass dengan parameter ***parentName***.
2. Pada kelas Parent, tidak ada konstruktor yang tanpa parameter untuk dipanggil secara implisit ketika menggunakan ***super()*** di konstruktor kelas ***Baby***.

**Berikut adalah perbaikan dari kode tersebut:**

****

**Perubahan yang dilakukan:**

1. Pada konstruktor kelas ***Baby***, menambahkan pemanggilan konstruktor superclass ***super()*** tanpa parameter untuk memanggil konstruktor superclass yang sesuai.
2. Menghapus parameter ***parentName*** pada konstruktor ***Parent*** yang kosong, karena tidak digunakan dalam kode.

**Top of Form**