**TUGAS PBO MINGGU 3**

Memenuhi Tugas Mata Kuliah Pemrograman Berbasis Objek

Dosen : Odhitya Desta Triswidrananta, S.Pd, M.Pd



Nama : Raden Dimas Erlangga   
Kelas : D-III Manajemen Informatika 2E  
Nim : 2031710121

**PROGRAM STUDI D-III MANAJEMEN INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

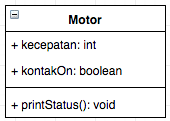
**2021**

**JOBSHEET 3**

**Percobaan 1**

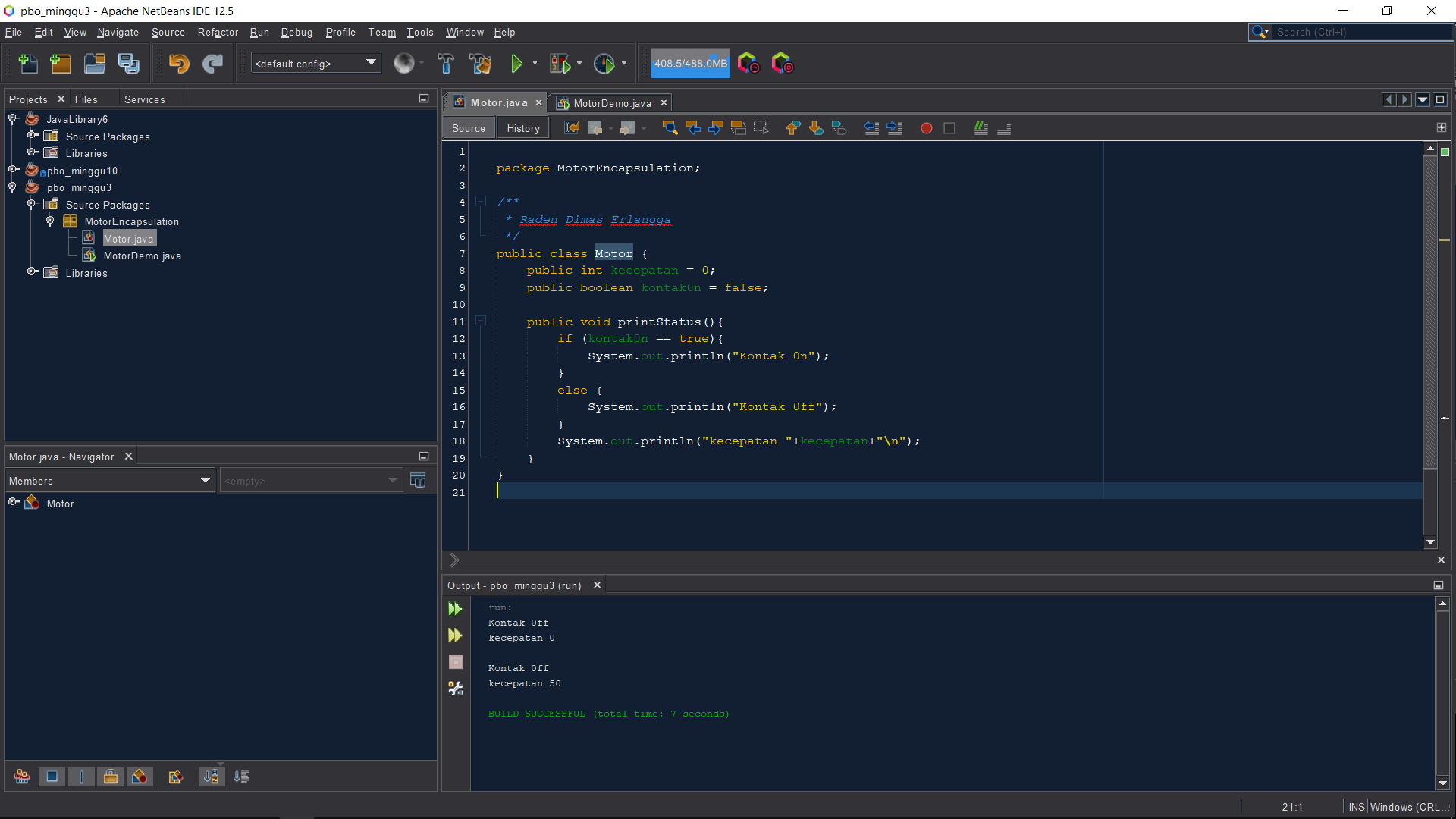
Didalam percobaan enkapsulasi, buatlah class Motor yang memiliki atribut kecepatan dan kontakOn, dan memiliki method printStatus() untuk menampilkan status motor. Seperti berikut

bentuk UML class diagram class Motor adalah sebagai berikut:

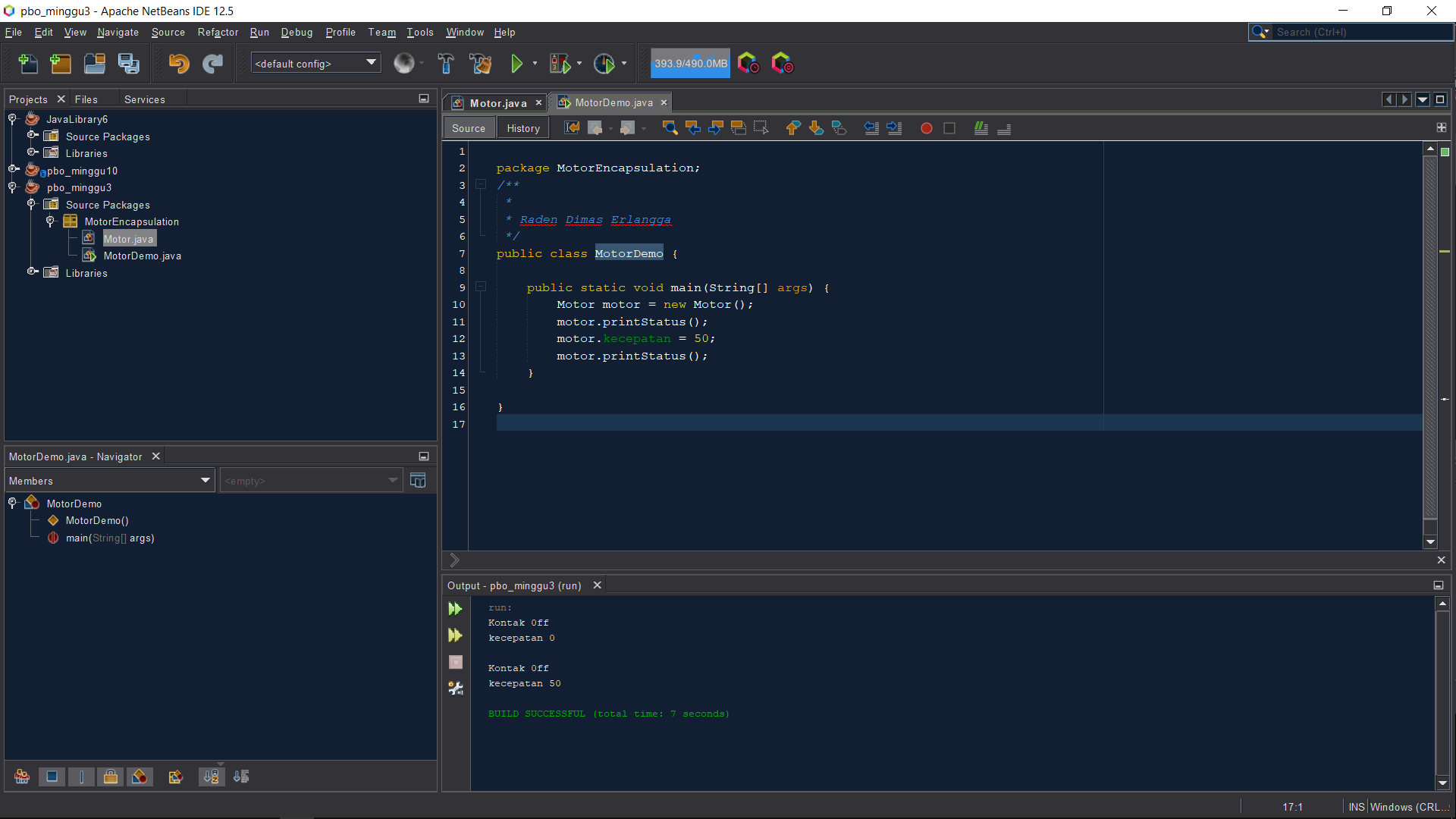


Program Java :

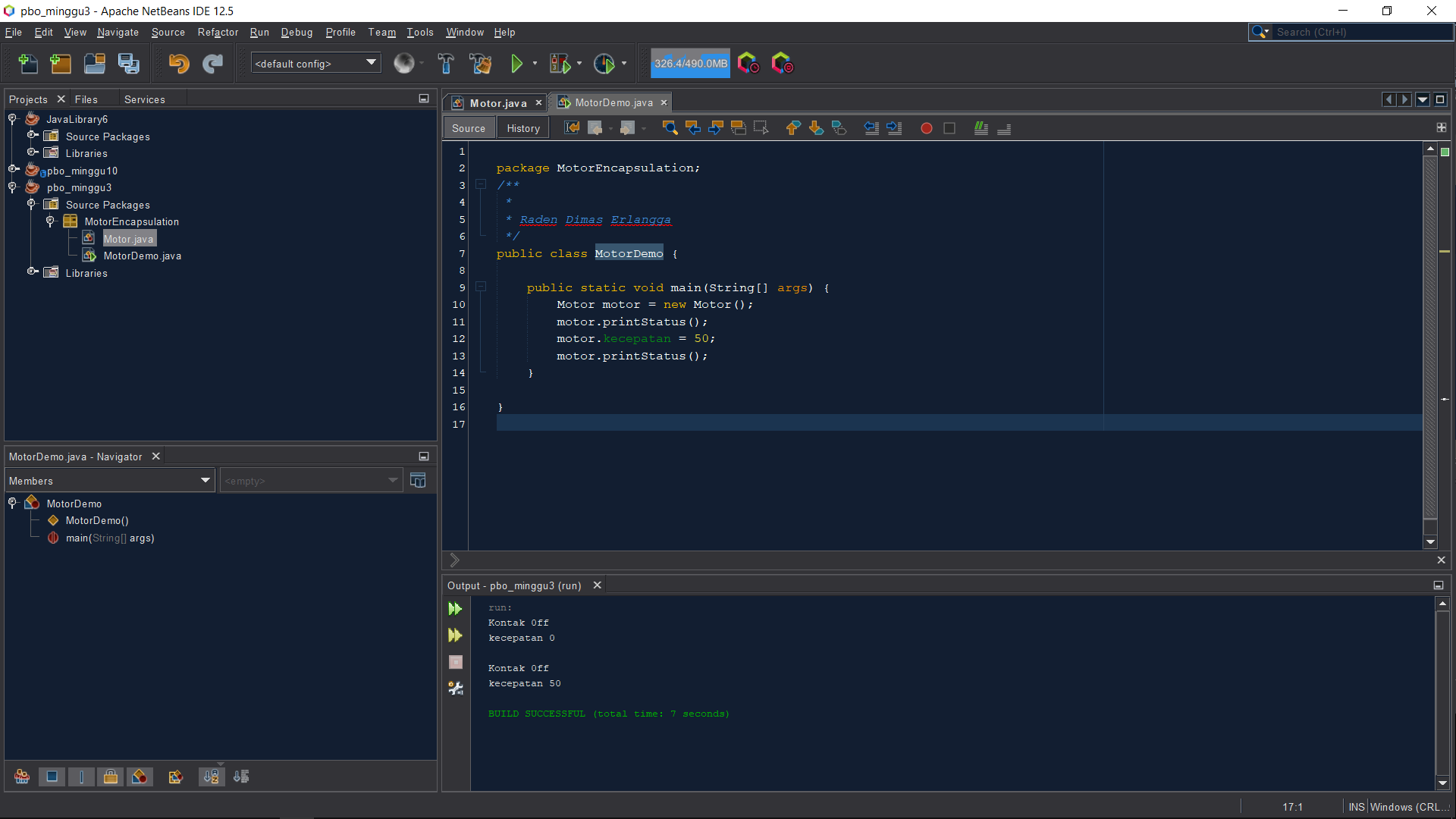
A. Class Motor



B. Class MotorDemo



C. Hasilnya adalah sebagai berikut:



Dari percobaan 1 - enkapsulasi, menurut anda, adakah yang janggal?

Yaitu, kecepatan motor tiba-tiba saja berubah dari 0 ke 50. Lebih janggal lagi, posisi kontak motor masih dalam kondisi OFF. Bagaimana mungkin sebuah motor bisa sekejap berkecepatan dari nol ke 50, dan itupun kunci kontaknya OFF?

dalam hal ini, akses ke atribut motor ternyata tidak terkontrol. Padahal, objek di dunia nyata selalu memiliki batasan dan mekanisme bagaimana objek tersebut dapat digunakan. Lalu, bagaimana kita bisa memperbaiki class Motor diatas agar dapat digunakan dengan baik? Kita bisa pertimbangkan beberapa hal berikut ini:

1. Menyembunyikan atribut internal (kecepatan, kontakOn) dari pengguna (class lain)
2. Menyediakan method khusus untuk mengakses atribut.

Untuk itu mari kita lanjutkan percobaan berikutknya tentang Access Modifier.

**Percobaan 2 Access Modifier**

Pada percobaan ini akan digunakan access modifier untuk memperbaiki cara kerja class Motor pada percobaan ke-1.

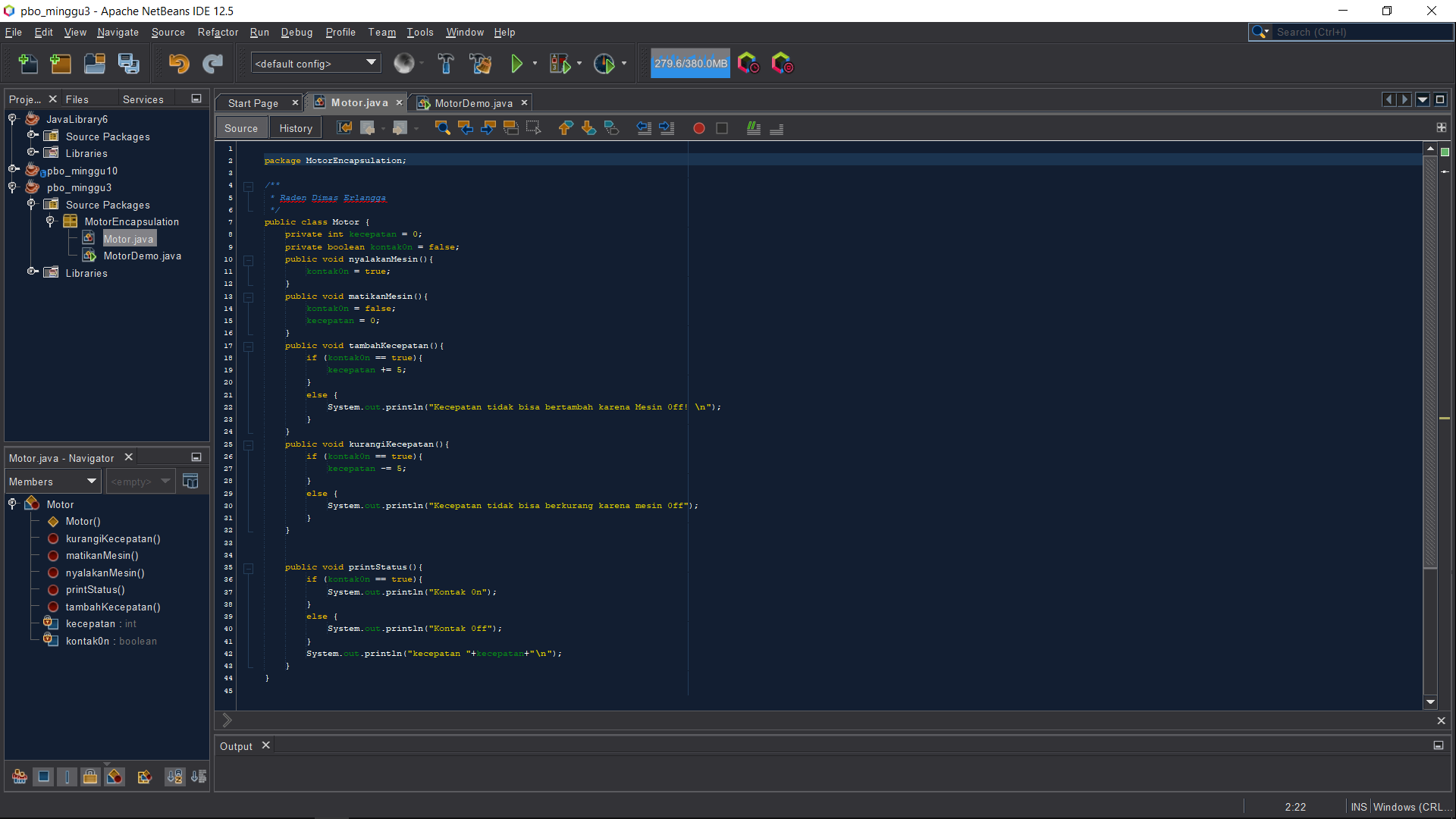
1. Ubah cara kerja class motor sesuai dengan UML class diagram berikut.



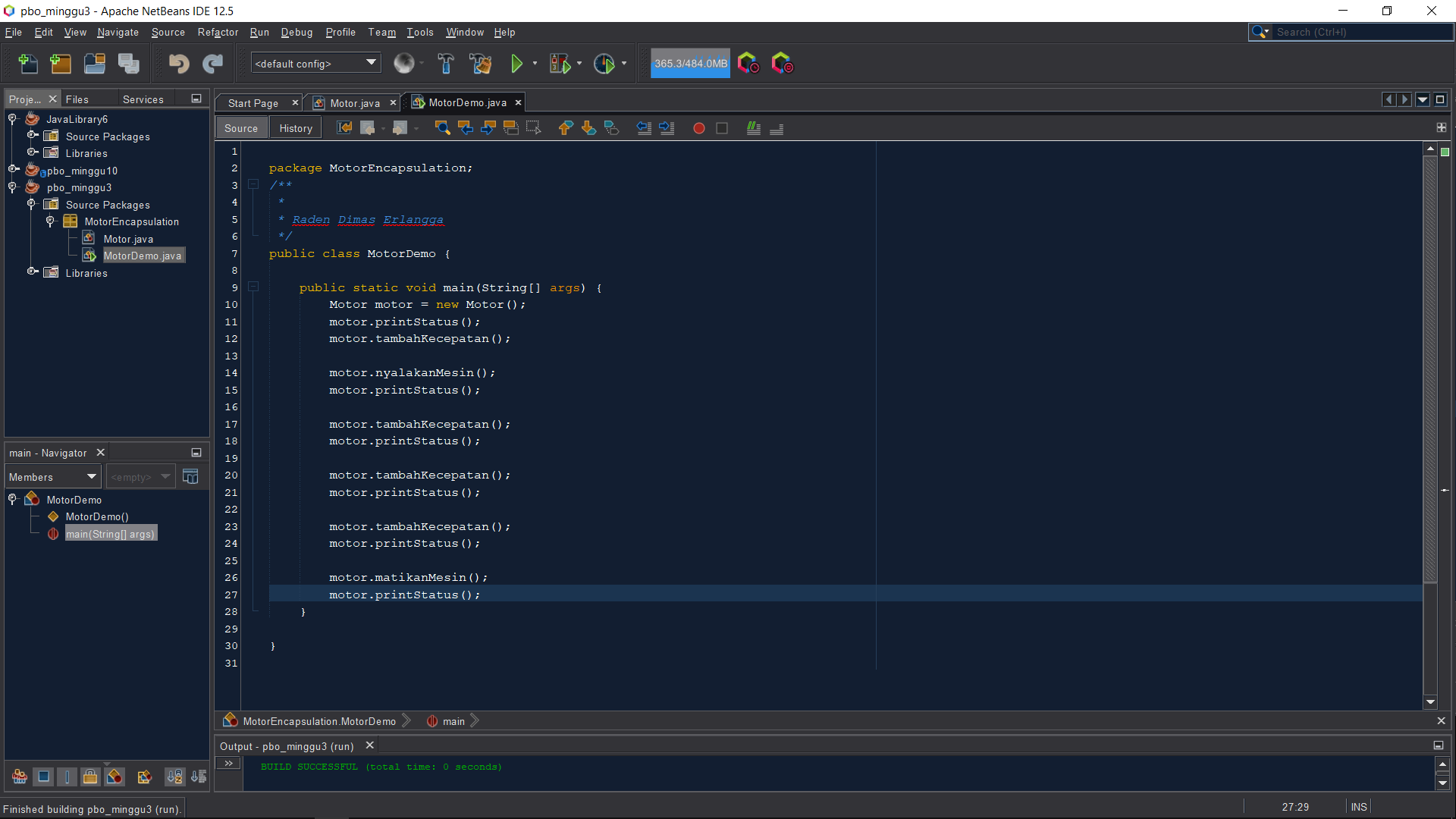
* 1. Berdasarkan UML class diagram tersebut maka class Motor terdapat perubahan, yaitu access modifier kecepatan dan kontakOn menjadi private
  2. Tambahkan method nyalakanMesin, matikanMesin, tambahKecepatan, kurangiKecepatan.

Maka :

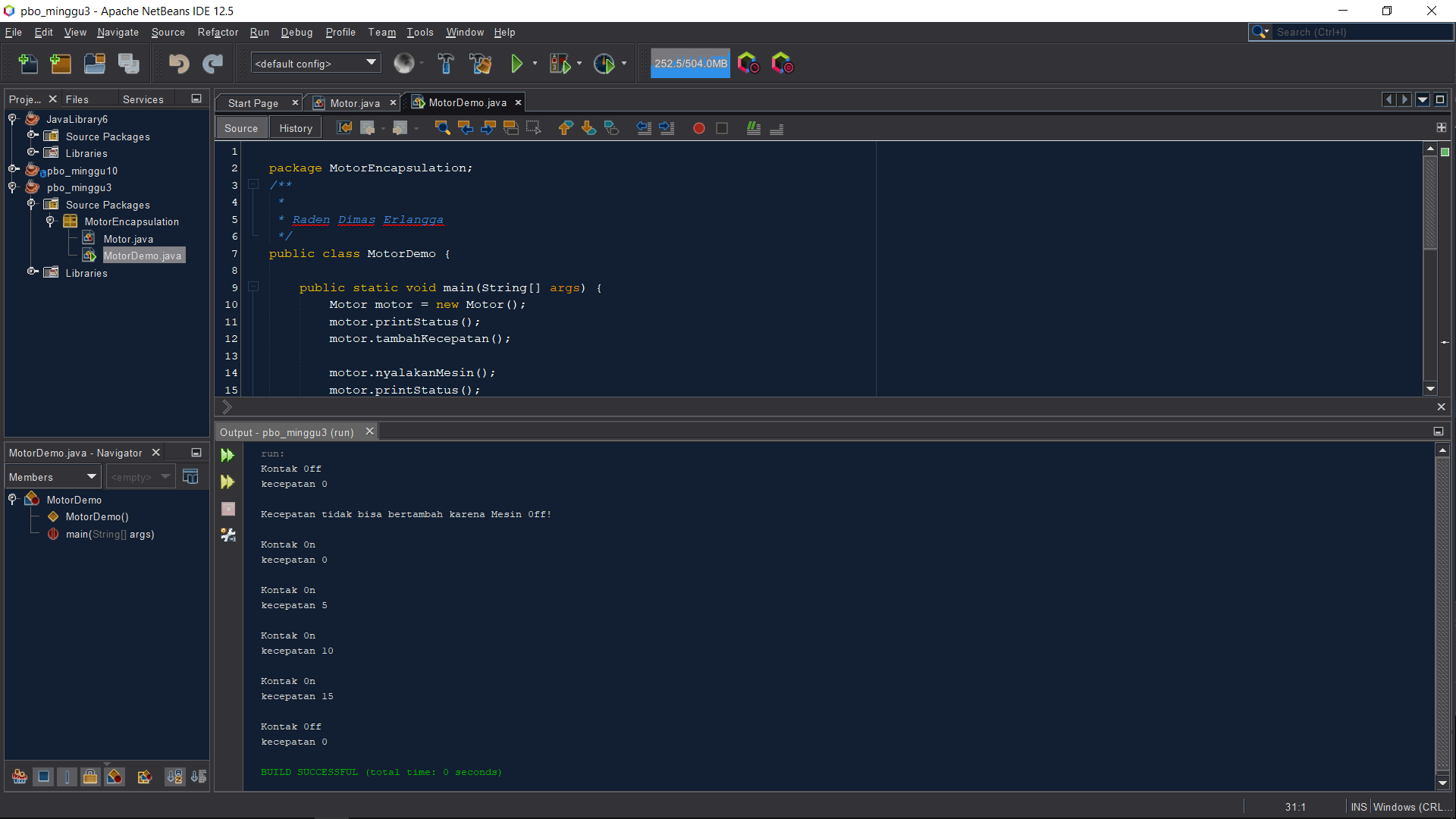
A. Class Motor



B. Class MotorDemo



C. Hasil Running



Dari percobaan diatas, dapat kita amati sekarang atribut kecepatan tidak bisa diakses oleh pengguna dan diganti nilainya secara sembarangan. Bahkan ketika mencoba menambah kecepatan saat posisi kontak masih OFF, maka akan muncul notifikasi bahwa mesin OFF. Untuk mendapatkan kecepatan yang diinginkan, maka harus dilakukan secara gradual, yaitu dengan memanggil method tambahKecepatan() beberapa kali. Hal ini mirip seperti saat kita mengendarai motor.

Pertanyaan

* + 1. Pada class TestMobil, saat kita menambah kecepatan untuk pertama kalinya, mengapa muncul peringatan “Kecepatan tidak bisa bertambah karena Mesin Off!”?

Jawaban :

Karena pada class Motor, terdapat if statement, yang mencegah kecepatan mesin bertambah jika keadaan mesin mati.

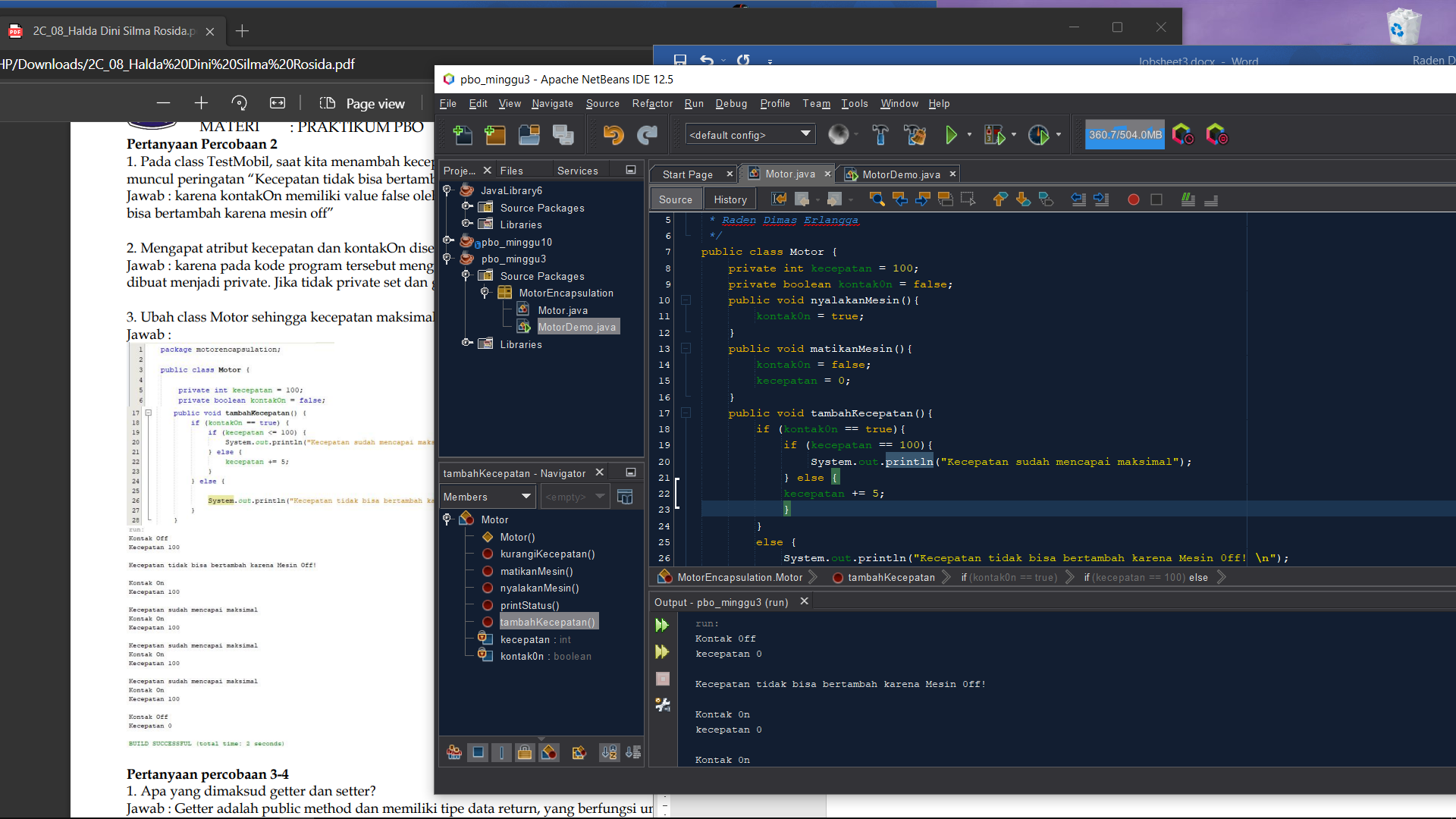
* + 1. Mengapat atribut kecepatan dan kontakOn diset private?

Jawaban :

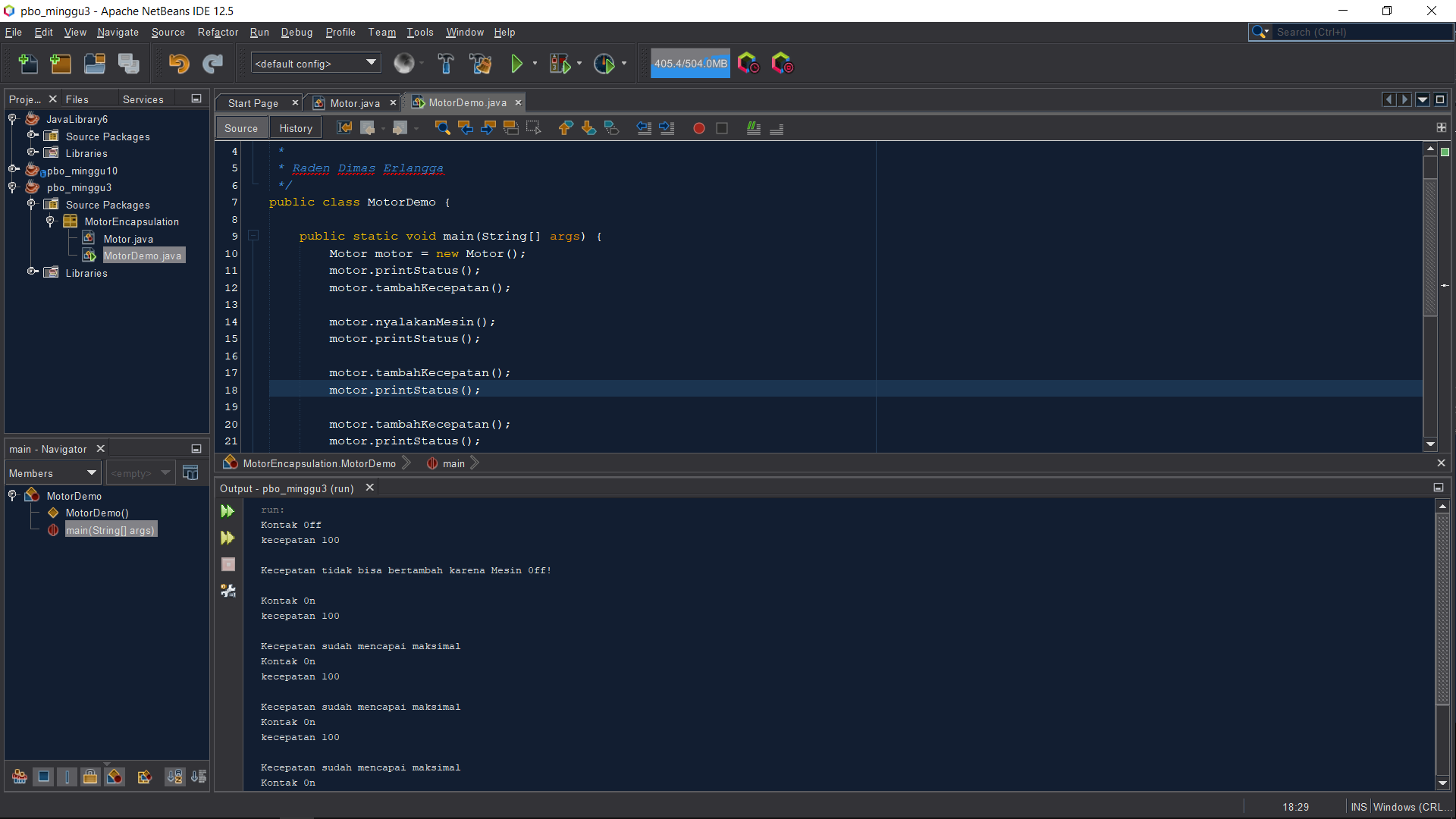
Karena class motor berada di satu package yang sama, maka bisa menggunakan atribut private untuk melakukan access modifier

* + 1. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

Jawaban :



Disini saya mendeklarasi kecepatan = 100 untuk melakukan test apakah if statement bekerja, saat di run hasil nya adalah sebagai berikut :

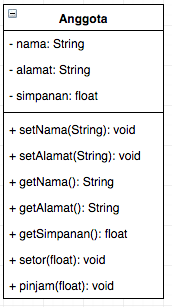


Method tambah kecepatan menampilkan print “kecepatan sudah mencapai maksimal”

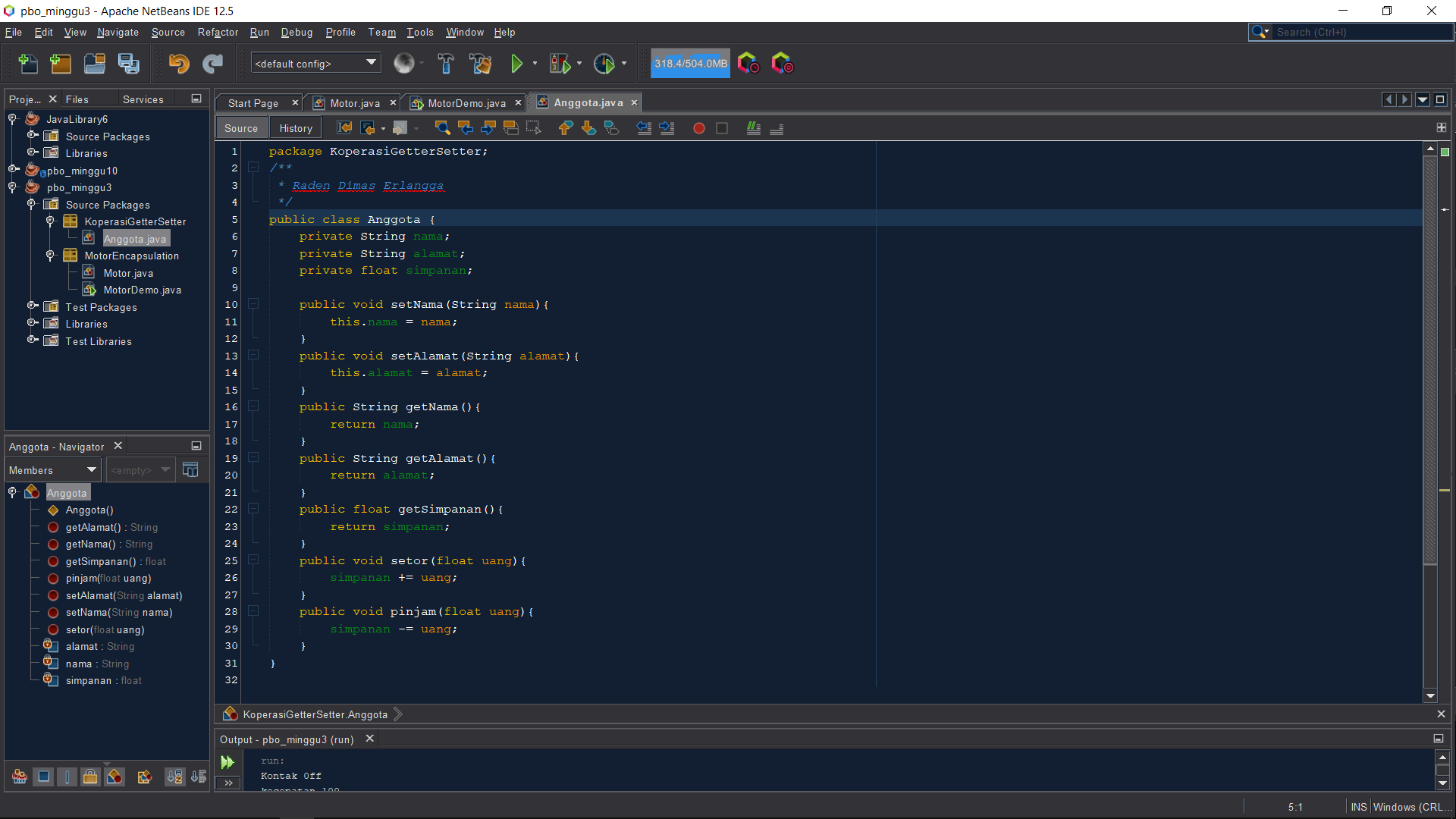
**Percobaan 3 Getter & Setter**

Misalkan di sebuah sistem informasi koperasi, terdapat class Anggota. Anggota memiliki atribut nama, alamat dan simpanan, dan method setter, getter dan setor dan pinjam. Semua atribut pada anggota tidak boleh diubah sembarangan, melainkan hanya dapat diubah melalui method setter, getter, setor dan tarik. Khusus untuk atribut simpanan tidak terdapat setter karena simpanan akan bertambah ketika melakukan transaksi setor dan akan berkurang ketika melakukan peminjaman/tarik.

1. Berikut ini UML class buatlah class Mahasiswa pada program:

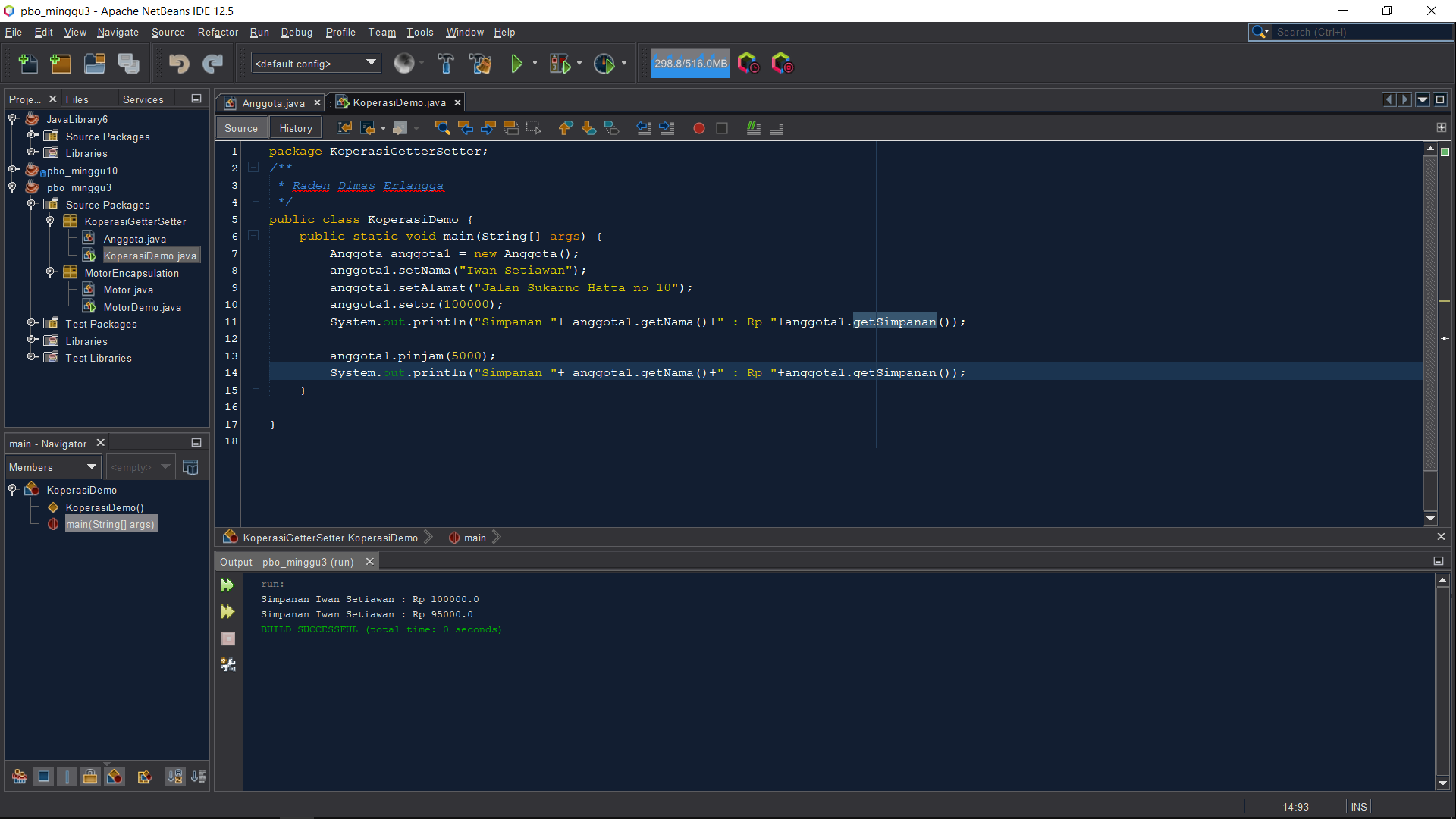


2. class Anggota :

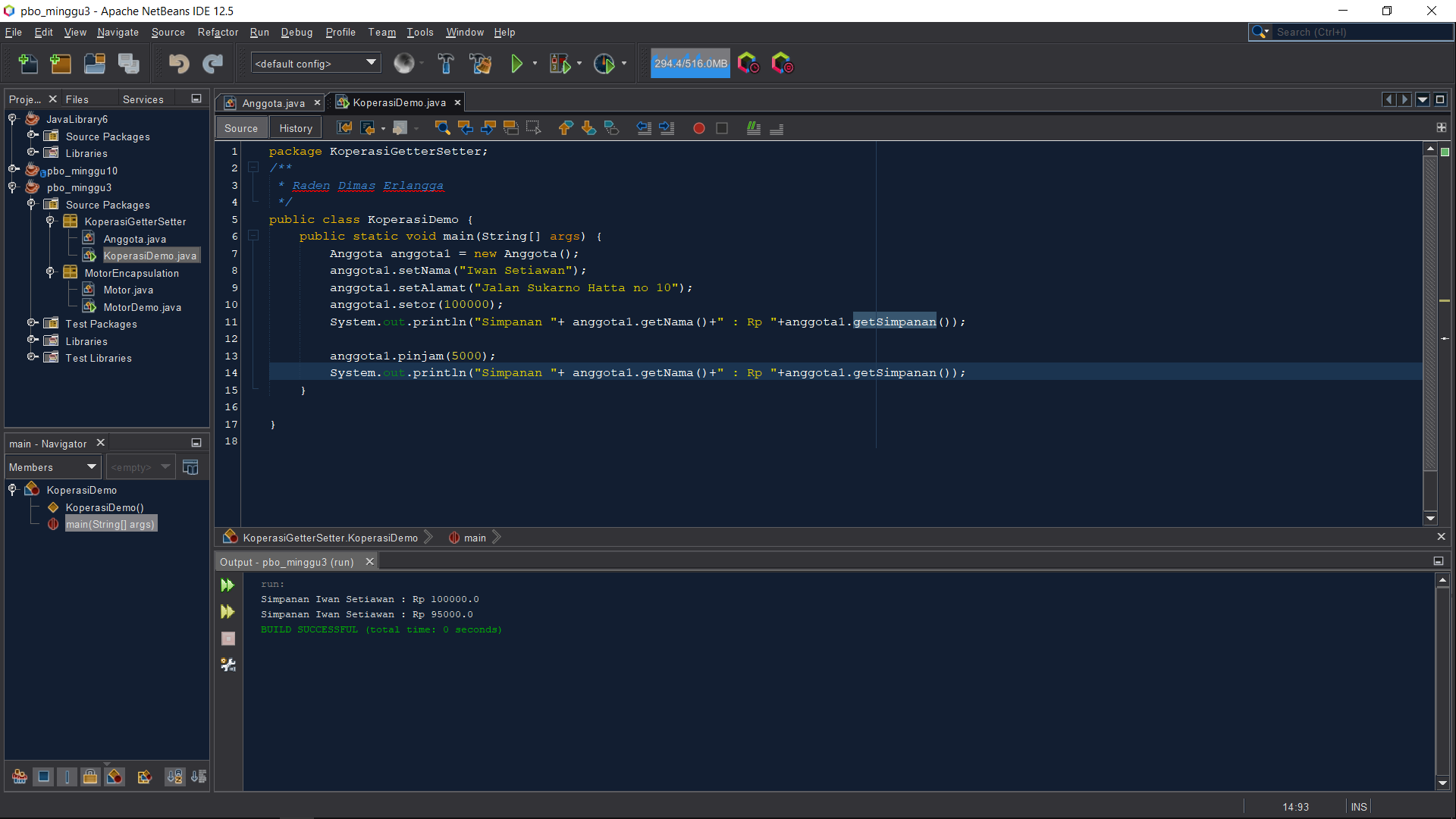


Jika diperhatikan pada class Anggota, atribut nama dan alamat memili masing-masing 1 getter dan setter. Sedangkan atribut simpanan hanya memiliki getSimpanan() saja, karena seperti tujuan awal, atribut simpanan akan berubah nilainya jika melakukan transaksi setor() dan pinjam/tarik().

3. class KoperasiDemo



Hasil Running :



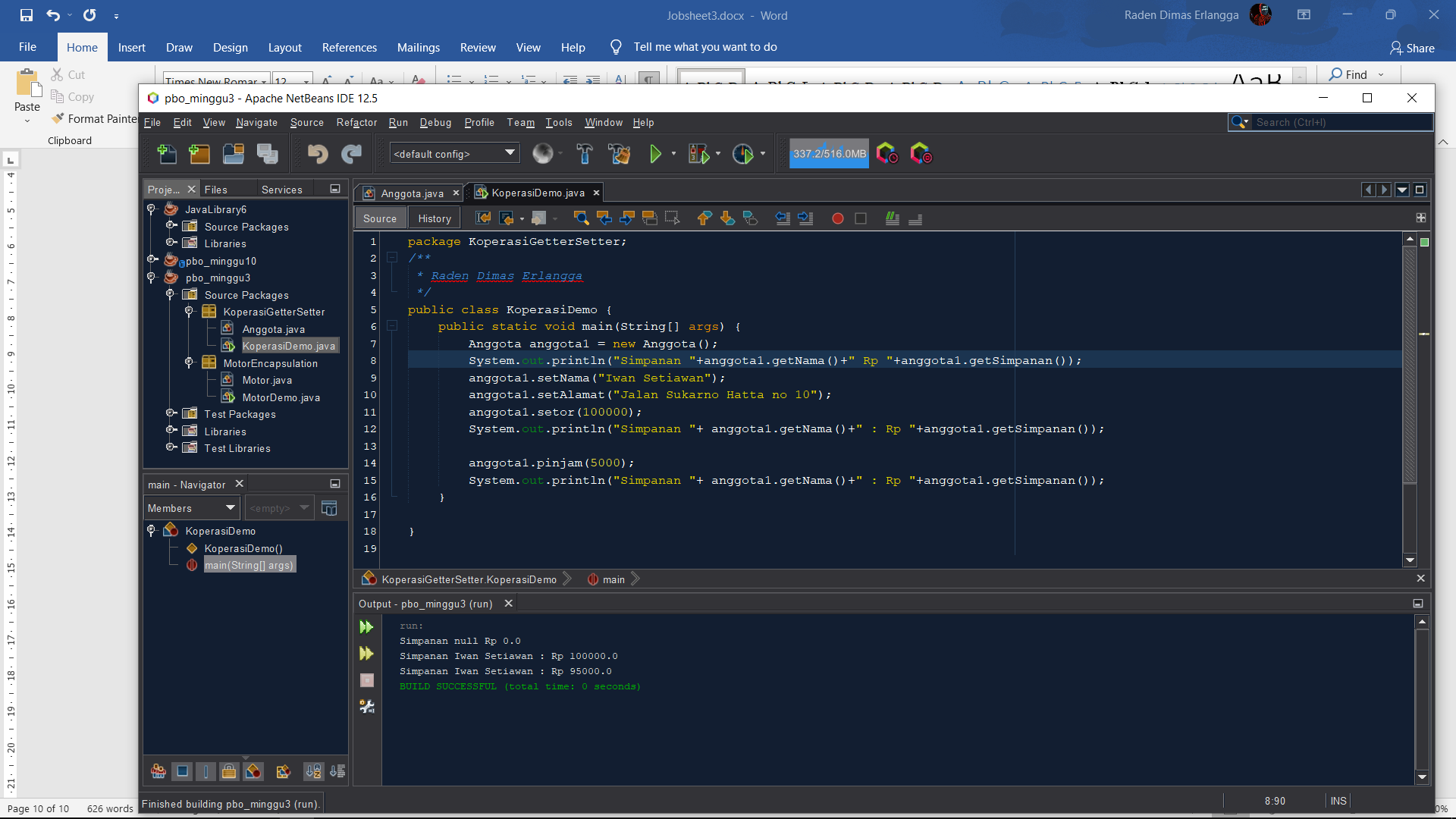
Dapat dilihat pada hasil percobaan diatas, untuk mengubah simpanan tidak dilakukan secara langsung dengan mengubah atribut simpanan, melainkan melalui method setor() dan pinjam(). Untuk menampilkan nama pun harus melalui method getNama(), dan untuk menampilkan simpanan melalui getSimpanan().

**Percobaan 4 Konstruktor, instansiasi**

1. ubah class KoperasiDemo :

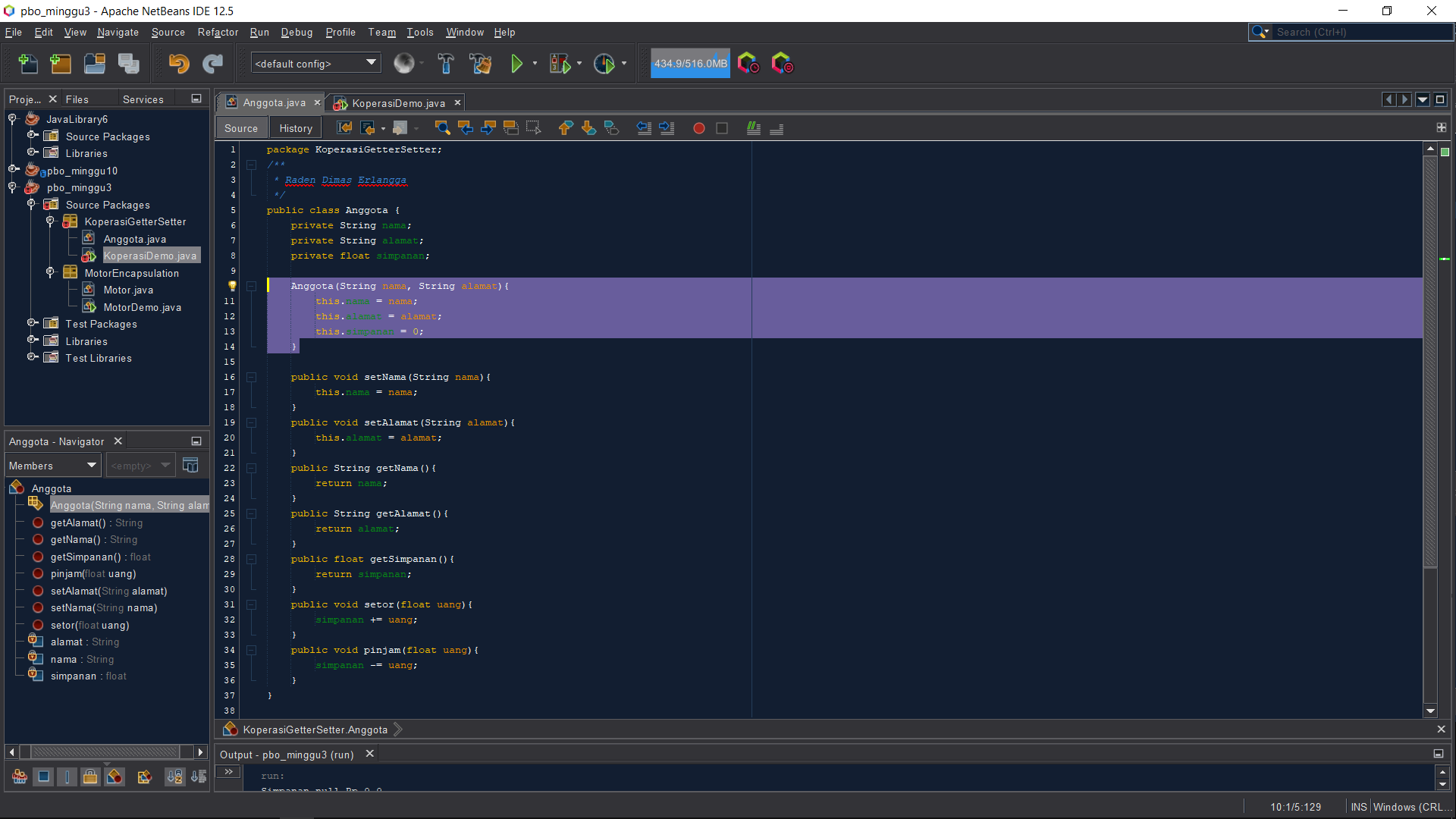


Hasil running :



Dapat dilihat hasil running program, ketika dilakukan pemanggilan method getNama() hasilnya hal ini terjadi karena atribut nama belum diset nilai defaultnya. Hal ini dapat ditangani dengan membuat kontruktor.

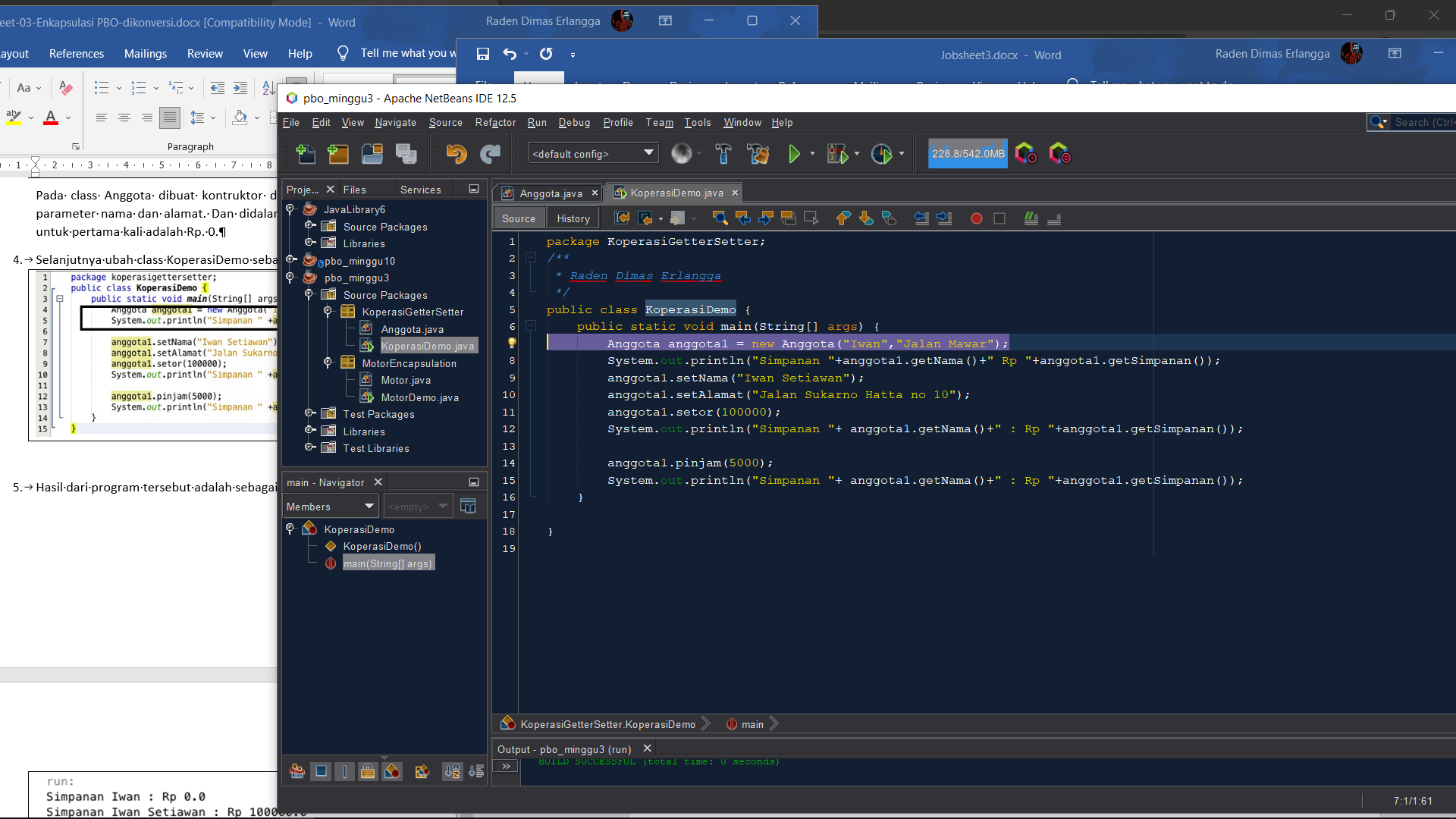
3. Ubah Class anggota menjadi seperti berikut :



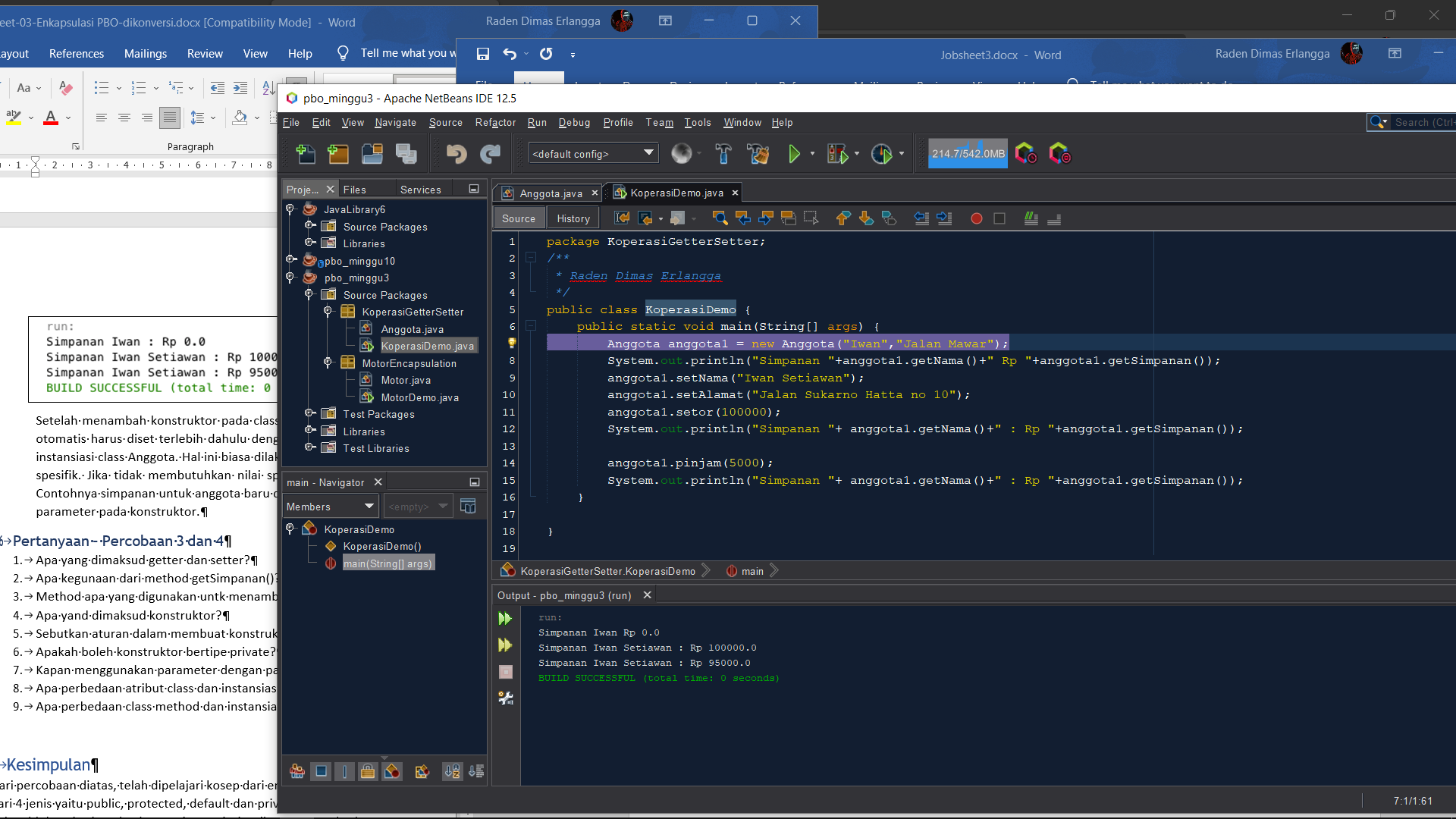
Pada class Anggota dibuat kontruktor dengan access modifier default yang memiliki 2 parameter nama dan alamat. Dan didalam konstruktor tersebut dipastikan nilai simpanan untuk pertama kali adalah

Rp. 0

4. selanjut nya ubah class koperasiDemo sebagai berikut :



5. Hasil Running



Setelah menambah konstruktor pada class Anggoata maka atribut nama dan alamat secara otomatis harus diset terlebih dahulu dengan melakukan passing parameter jika melakukan instansiasi class Anggota. Hal ini biasa dilakukan untuk atribut yang membutuhkan nilai yang spesifik. Jika tidak membutuhkan nilai spesifik dalam konstruktor tidak perlu parameter. Contohnya simpanan untuk anggota baru diset 0, maka simpanan tidak perlu untuk dijadikan parameter pada konstruktor.

Pertanyaan - Percobaan 3 dan 4

* + 1. Apa yang dimaksud getter dan setter?

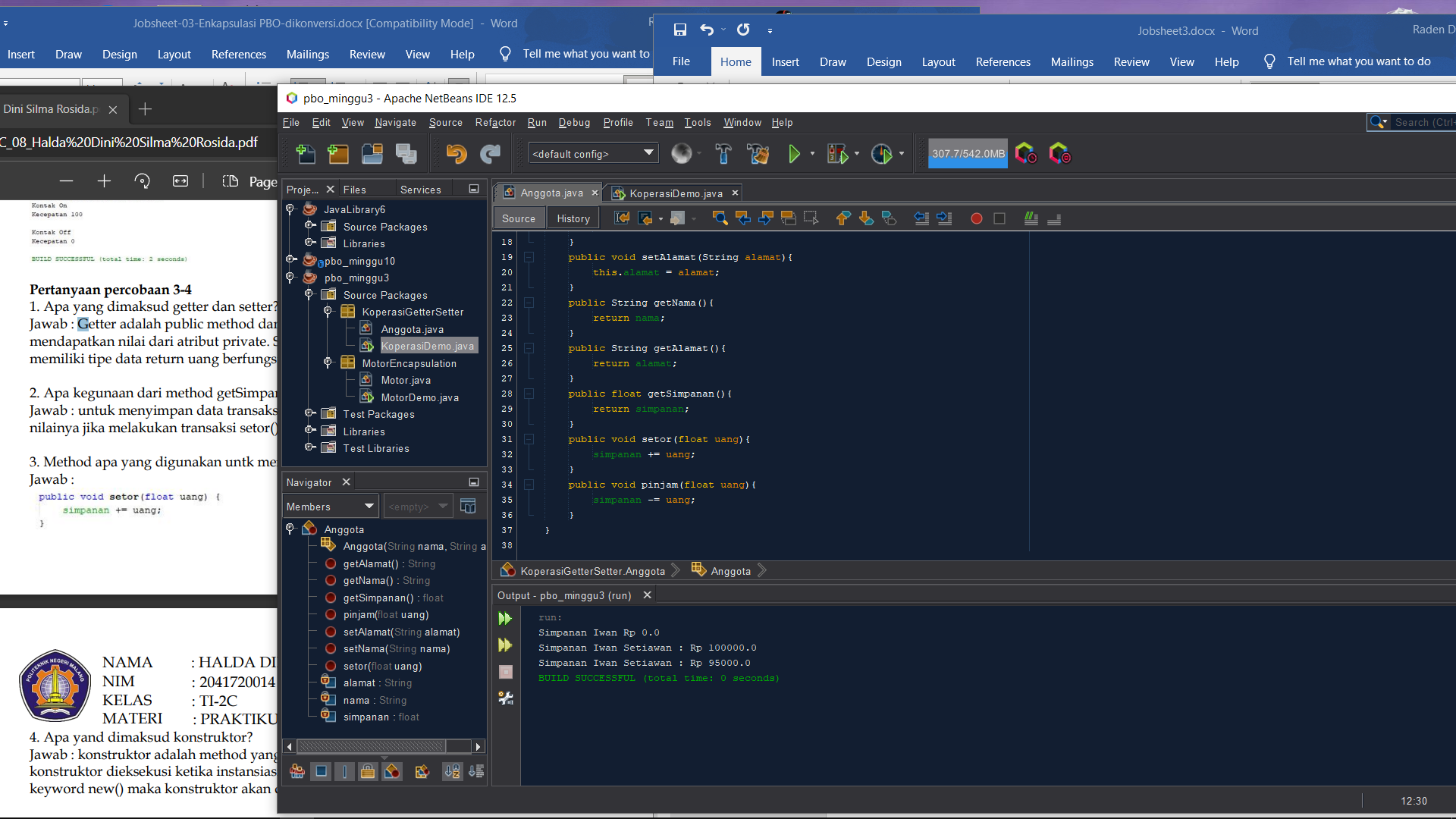
Jawaban : Getter adalah method yang memiliki return statement untuk mengambil niali dari atribut private, sedangkan setter adalah method yang tidak memiliki return, namun setter berfungsi untuk merubah / manipulasi nilai dari atribut private.

* + 1. Apa kegunaan dari method getSimpanan()?

Jawaban : untuk menyimpan data setor dan data pinjam.

* + 1. Method apa yang digunakan untk menambah saldo?

Jawaban : method setor :



* + 1. Apa yang dimaksud konstruktor?

Jawaban : Konstruktor merupakan suatu method yang tidak memiliki return statement, Konstruktor akan memberikan nilai awal pada saat suatu objek pada saat perintah new / instansiasi dijalankan untuk membuat suatu objek.

* + 1. Sebutkan aturan dalam membuat konstruktor?

Jawaban : Aturan dalam membuat konstruktor adalah : Nama konstruktor harus sama dengan class, konstruktor tidak boleh menggunakan modifier abstract, static, final dan synchronized

* + 1. Apakah boleh konstruktor bertipe private?

Jawaban : boleh jika class main dan class yang akan di konstruktor kan berada dalam satu package yang sama.

* + 1. Kapan menggunakan parameter dengan passsing parameter?

Jawaban : passing parameter digunakan untuk memanggil sebuah data Ketika menggunakan sebuah konstruktor berparameter.

* + 1. Apa perbedaan atribut class dan instansiasi atribut?

Jawaban : atribut class merupakan variable pada suatu class, sedangkan instansiasi atribut merupakan nilai yang di inputkan pada suatu input.

* + 1. Apa perbedaan class method dan instansiasi method?

Class method adalah sebuah method yang berada di dalam sebuah class, sedangkan instansiasi method adalah pemanggilan dari method yang berada di luar kelas yang memanggil method. Contoh nya instansiasi method print() yang dilakukan pada method main, namun method print ini di panggil dari class selain main.

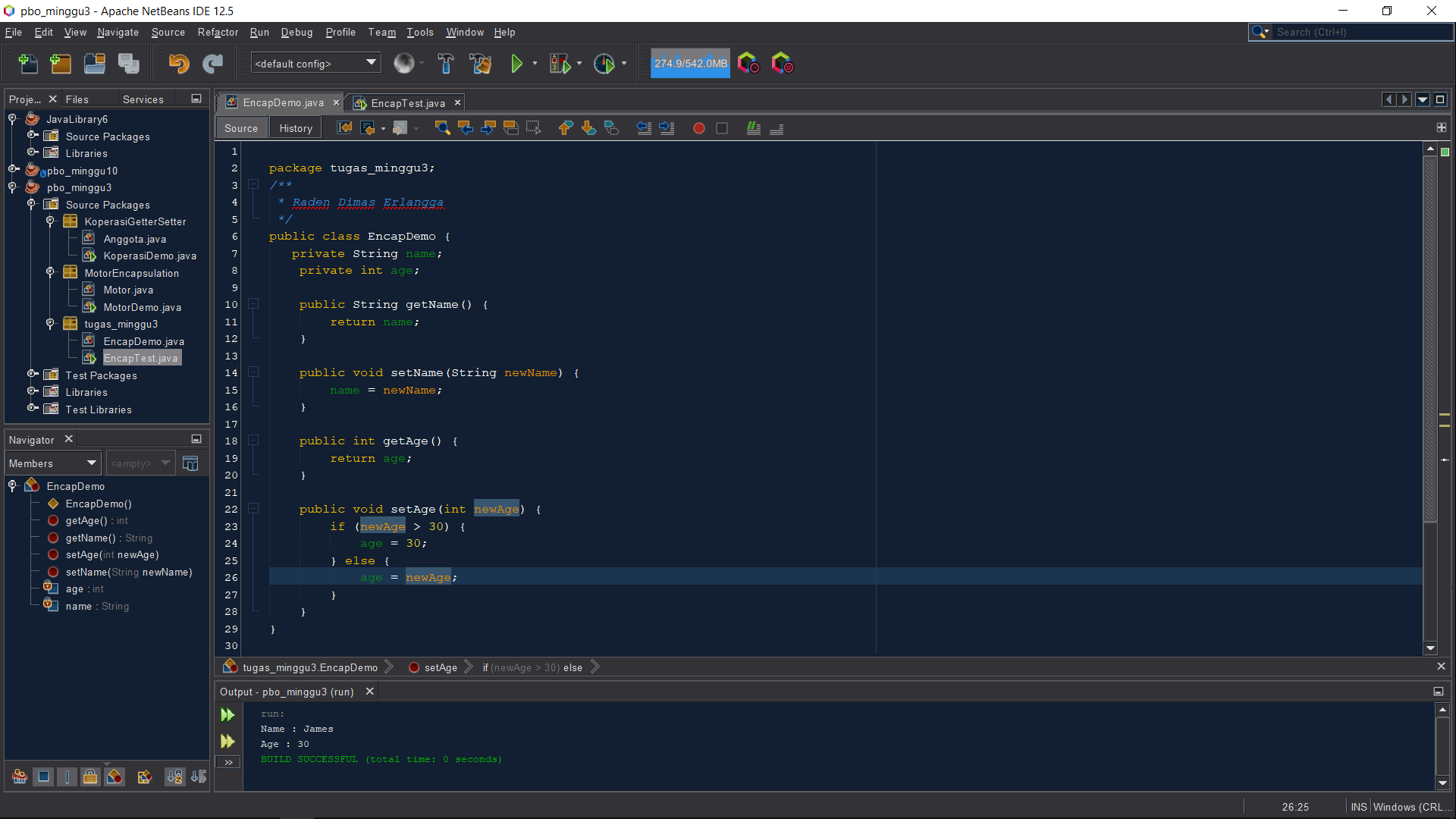
**Kesimpulan**

Dari percobaan diatas, telah dipelajari kosep dari enkapsulasi, kontruktor, access modifier yang terdiri dari 4 jenis yaitu public, protected, default dan private. Konsep atribut atau method class yang ada di dalam blok code class dan konsep instansiasi atribut atau method. Cara penggunaan getter dan setter beserta fungsi dari getter dan setter. Dan juga telah dipelajari atau memahami notasi UML

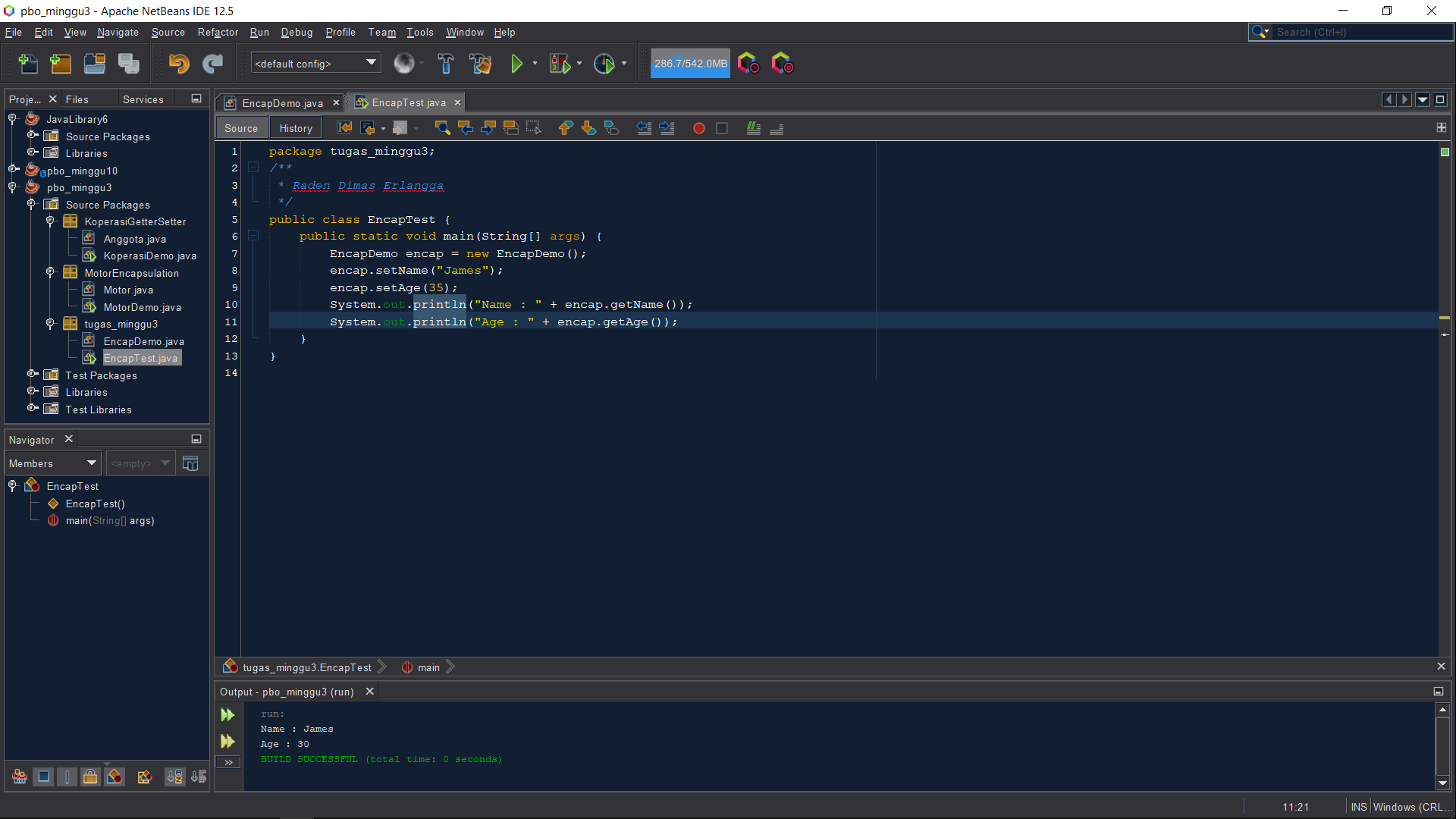
Tugas

1. Cobalah program dibawah ini dan tuliskan hasil outputnya

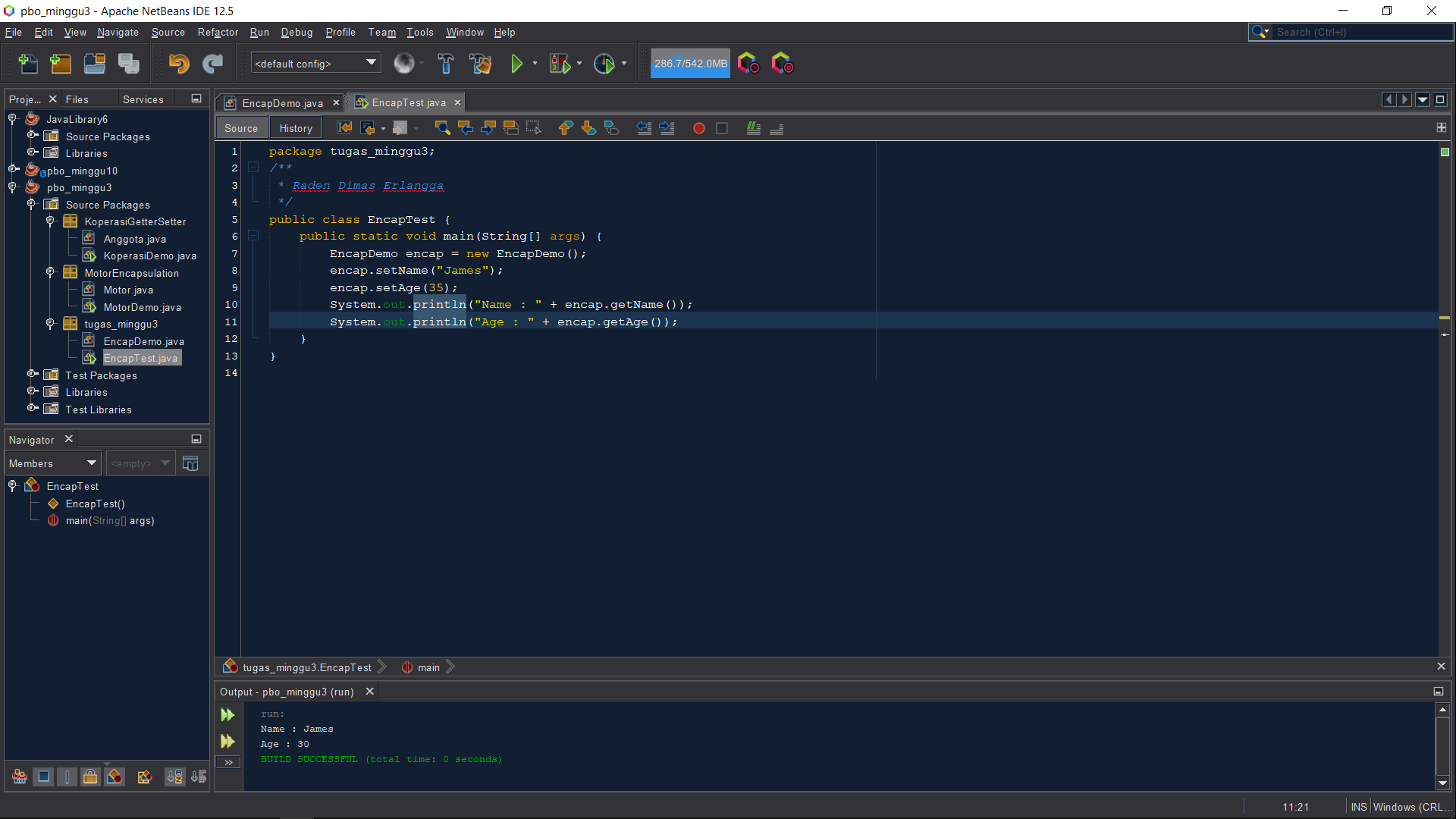
A. Class EncapDemo



B. Class EncapTest



Hasil Running :

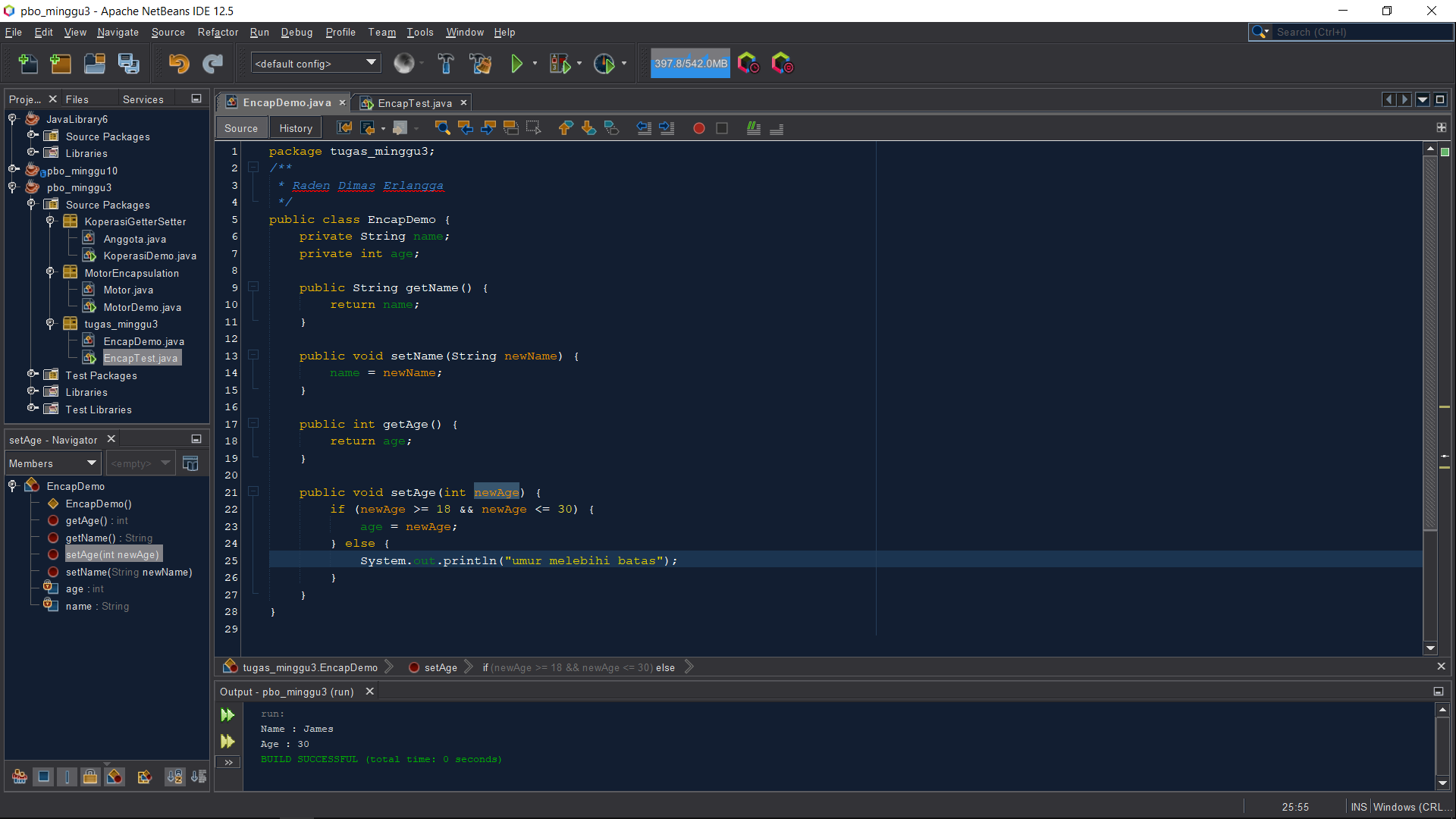


2. Pada program diatas, pada class EncapTest kita mengeset age dengan nilai 35, namun pada saat ditampilkan ke layar nilainya 30, jelaskan mengapa.

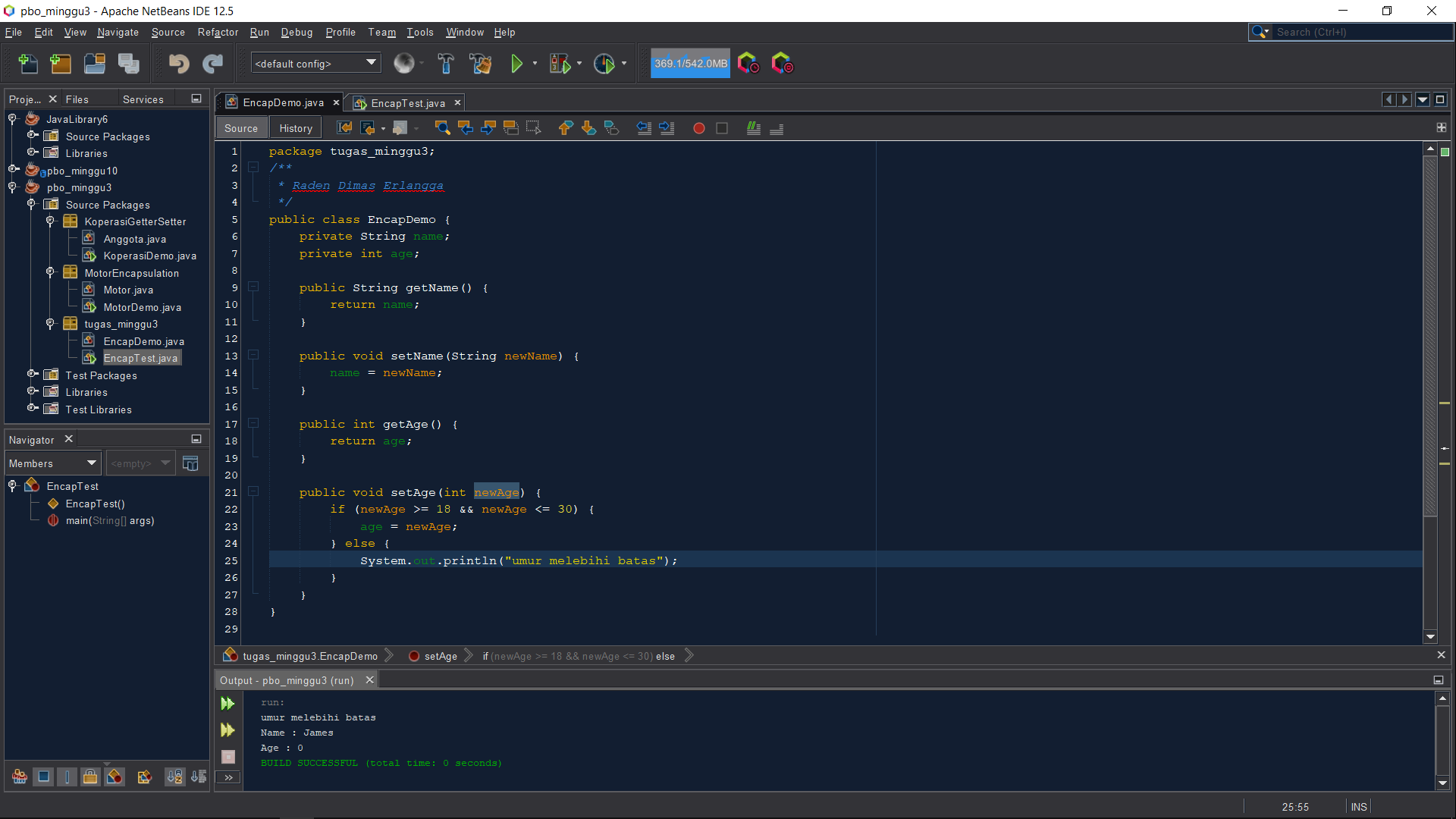
Jawaban : karena pada method setAge jika nilai yang di inputkan lebih dari 30 maka akan mendapatkan nilai age = 30.

3. Ubah program diatas agar atribut age dapat diberi nilai maksimal 30 dan minimal 18.

Jawaban :



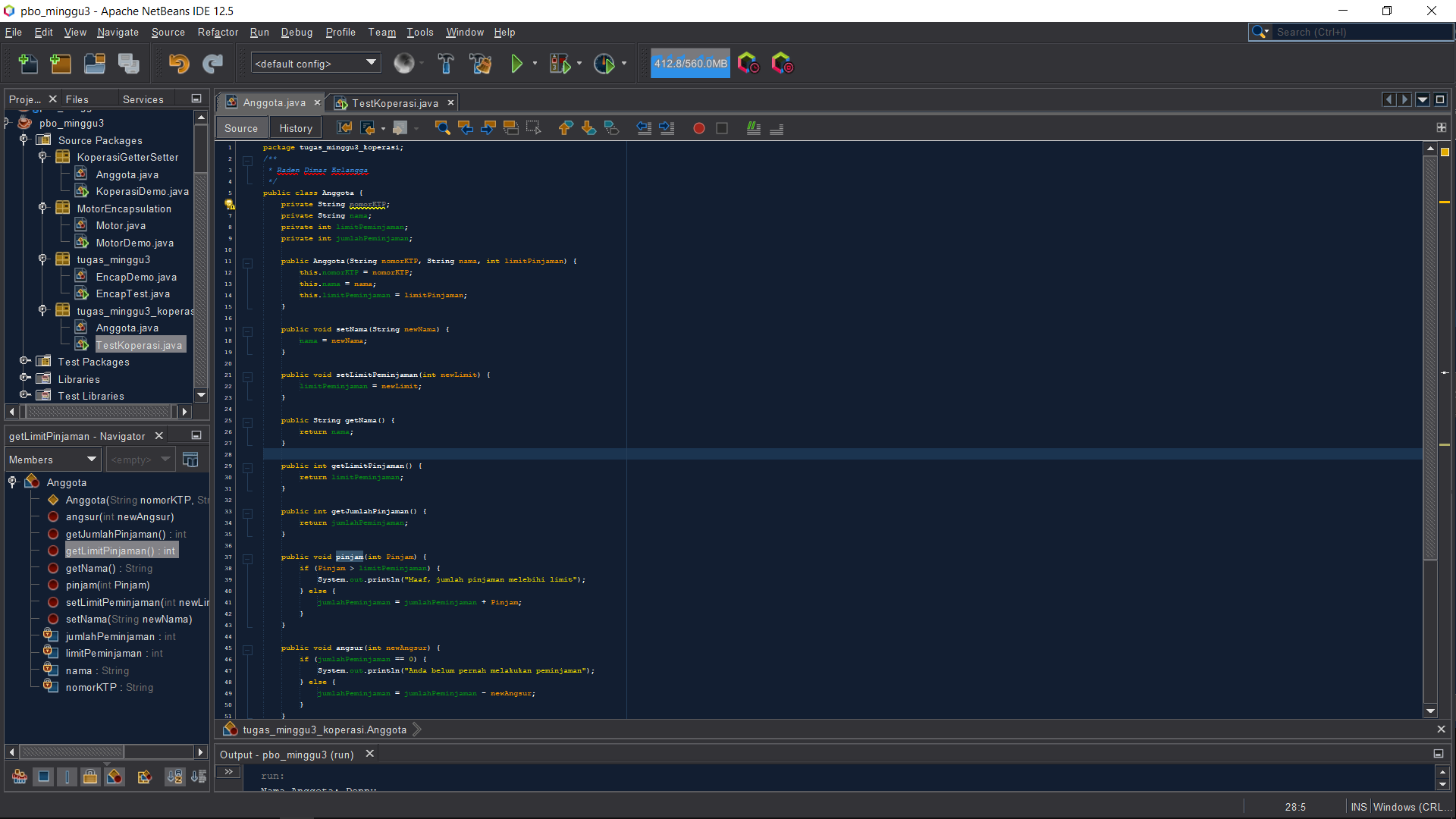
Hasil Running :



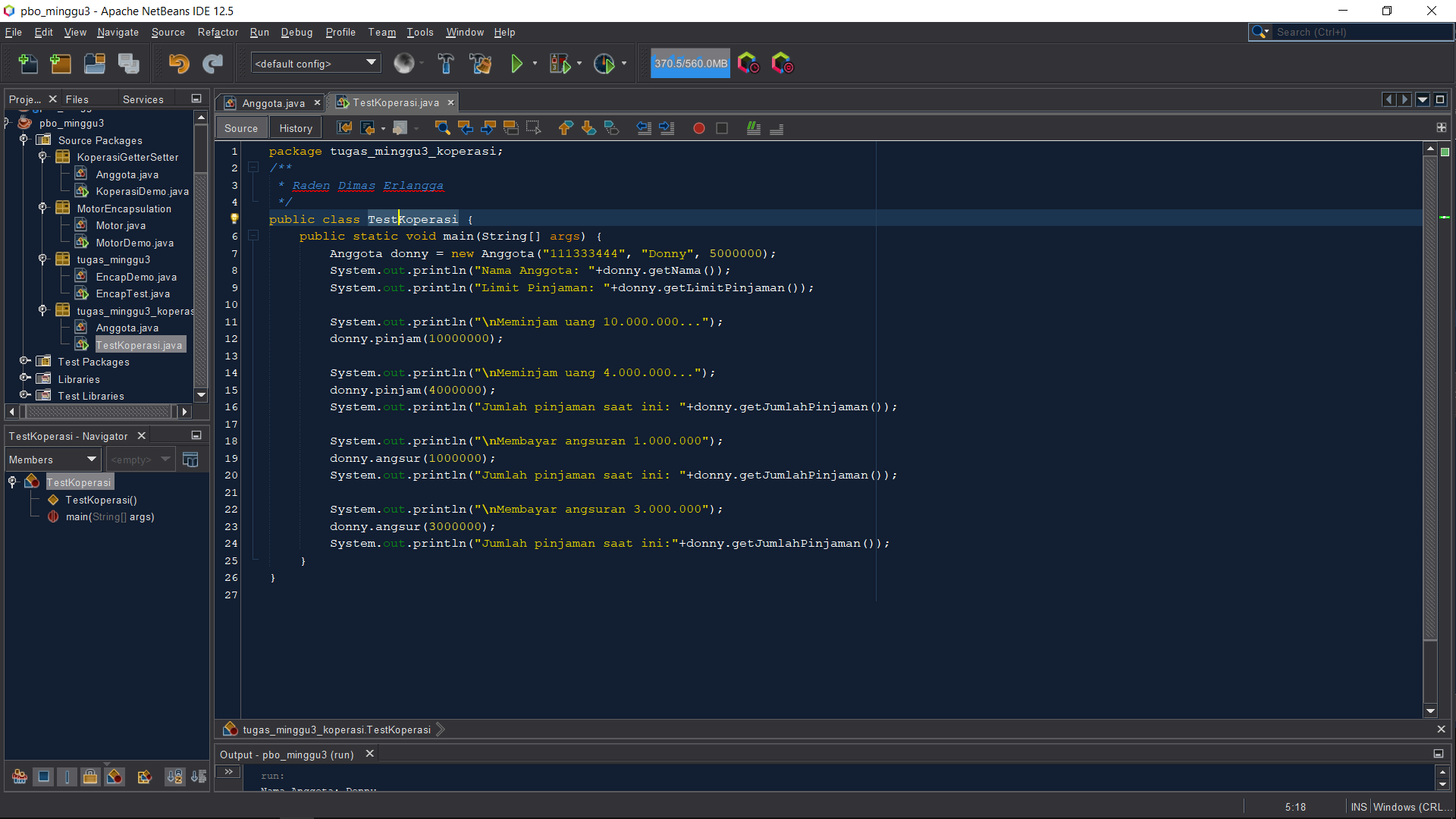
4. Pada sebuah sistem informasi koperasi simpan pinjam, terdapat class Anggota yang memiliki atribut antara lain nomor KTP, nama, limit peminjaman, dan jumlah pinjaman. Anggota dapat meminjam uang dengan batas limit peminjaman yang ditentukan. Anggota juga dapat mengangsur pinjaman. Ketika Anggota tersebut mengangsur pinjaman, maka jumlah pinjaman akan berkurang sesuai dengan nominal yang diangsur. Buatlah class Anggota tersebut, berikan atribut, method dan konstruktor sesuai dengan kebutuhan. Uji dengan TestKoperasi berikut ini untuk memeriksa apakah class Anggota yang anda buat telah sesuai dengan yang diharapkan.

Jawaban :

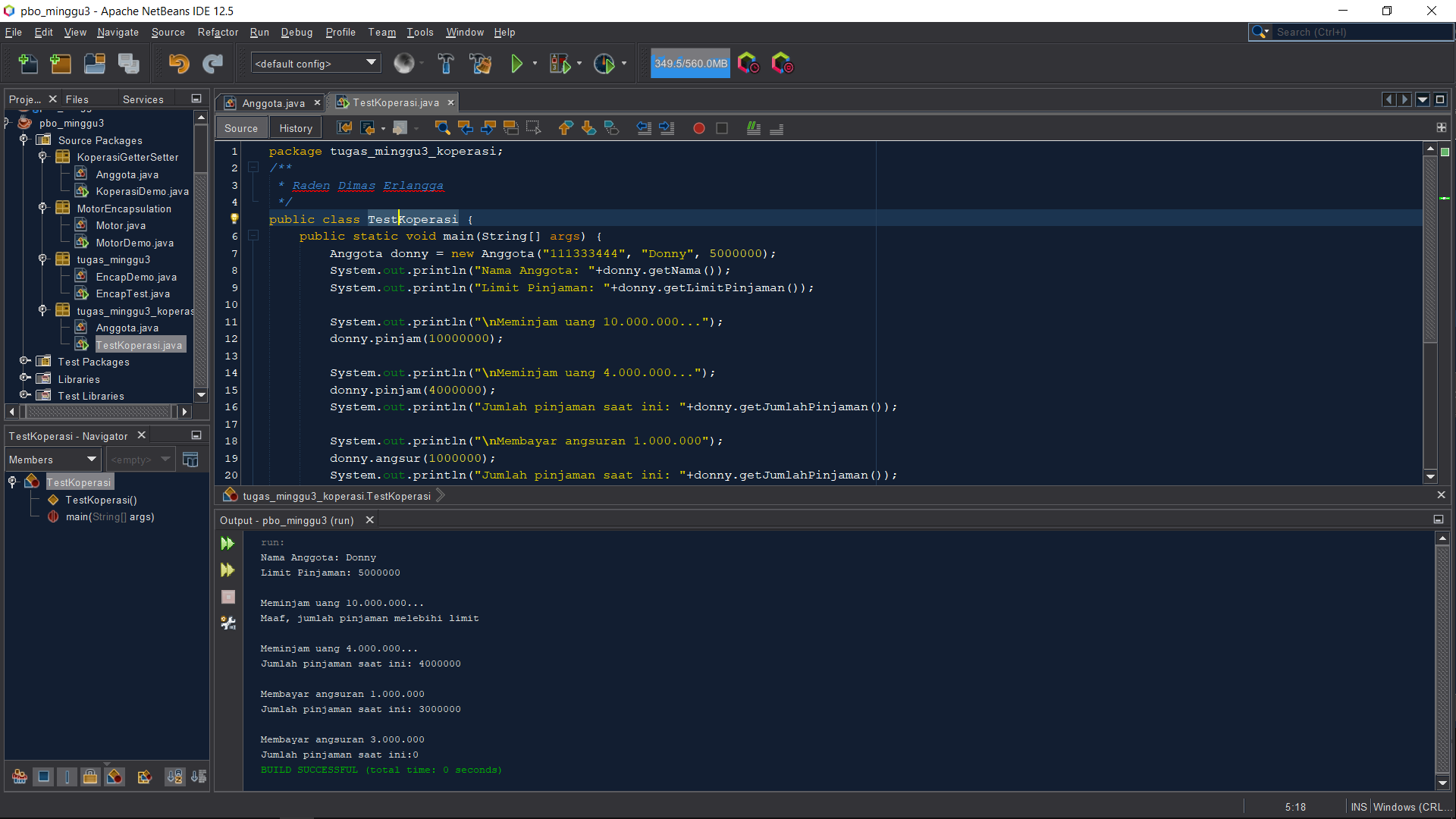
A. Class Anggota :



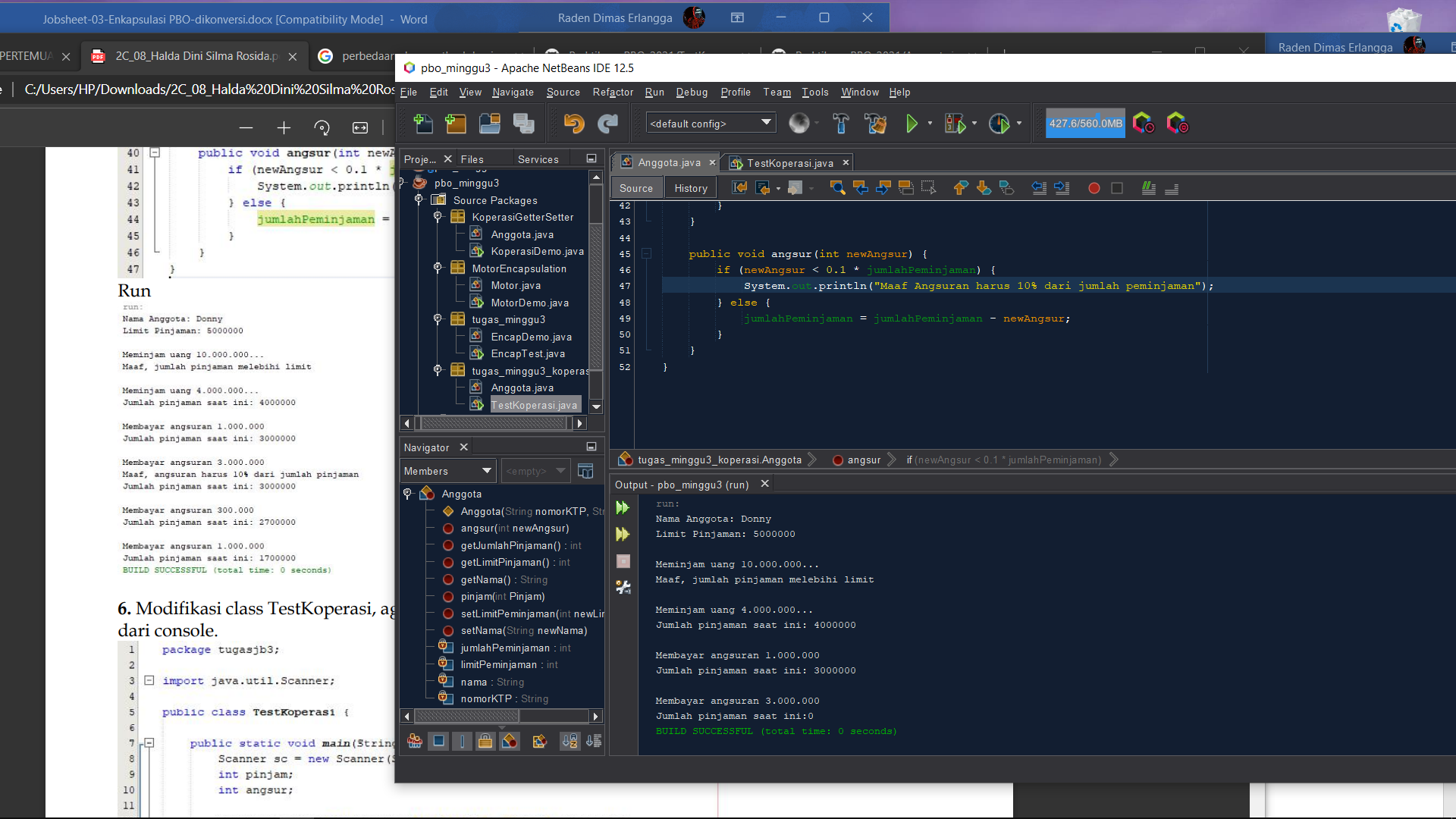
B. Class Test koperasi :



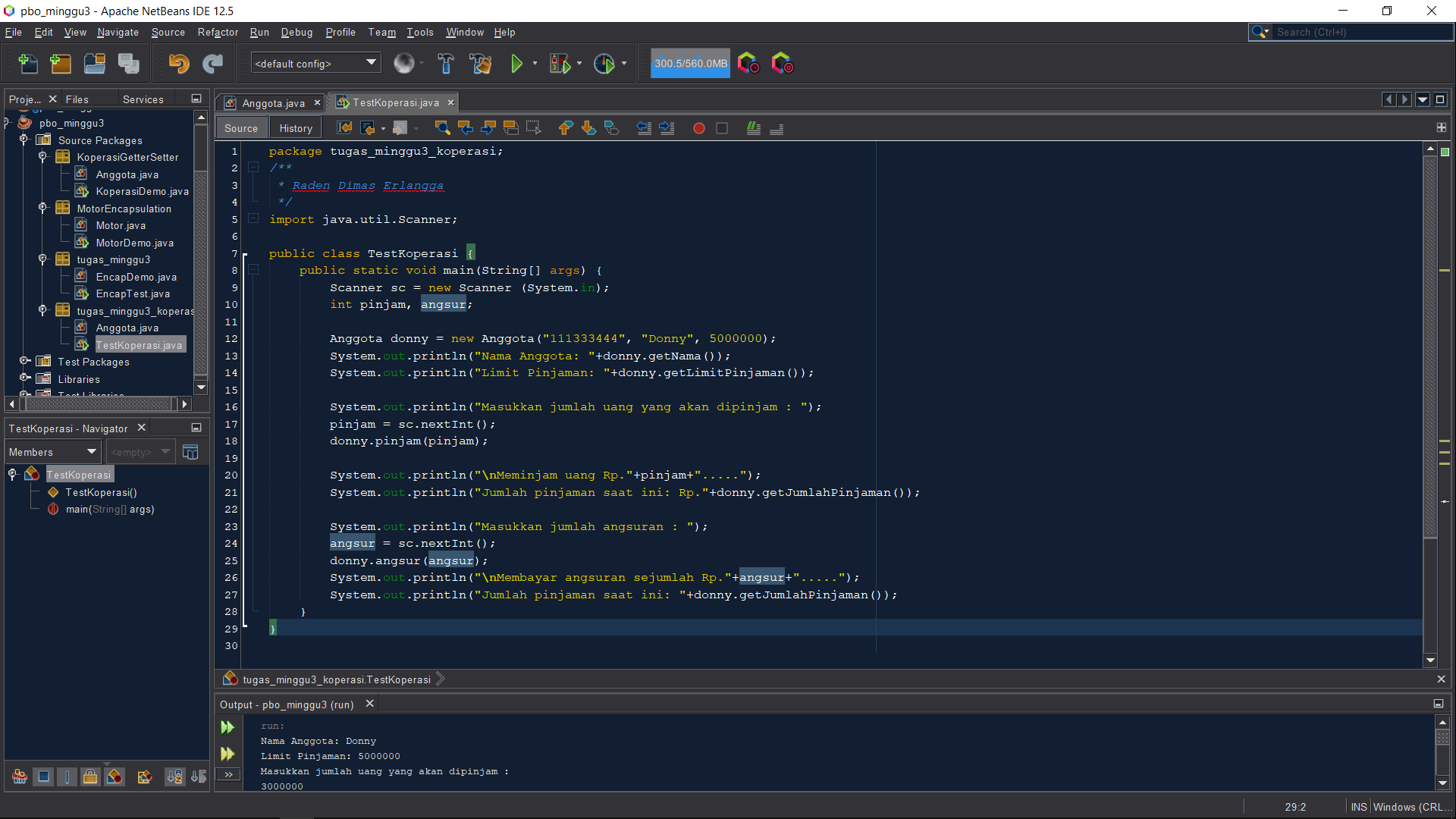
C. Hasil running :



5. Modifikasi soal no. 4 agar nominal yang dapat diangsur minimal adalah 10% dari jumlah pinjaman saat ini. Jika mengangsur kurang dari itu, maka muncul peringatan “Maaf, angsuran harus 10% dari jumlah pinjaman”.



6. Modifikasi class TestKoperasi, agar jumlah pinjaman dan angsuran dapat menerima input dari console.



Hasil running :

