**Soal Pertama**Jelaskan ap aitu Inheritance, bagaimana cara penerapannya serta aturan

Saja yang harus diperhatikan dalam menerapkan konsep ini?

**Jawaban**

Inheritance adalah suatu konsep yang mewariskan / menurunkan suatu

Karakteristik class ke class yang lain, adapun cara penerapannya adalah

* Dalam inheritance, suatu constructor tidak bisa di wariskan
* Sub-class dapat mengakses super-class dengan keyword super
* Ketika sub-class ingin mengakses atau mewarisi method dan attribute

Pada super-class maka diperlukan menulis keyword extends

**Soal Kedua**Perhatikankode-kode berikut

|  |
| --- |
| 1. Class kucing extends HewanMamalia 2. Class gajah extend HewanKarnivora 3. Class kelelawar extends HewanMamalia, HewanBersayap 4. Class Garuda extends HewanBersayap 5. Class Katak extends HewanBersayap |

Manakah penggunaan *Inheritance* yang benar dan memiliki hubungan *Generalisasi-Spesialisasi* yang masuk akal? Berikan penjelasanmu !

**Jawaban**

A dan B

Karena hewan kucing termasuk dalam hewan mamalia dan garuda juga termasuk hewan bersayap, maka masih termasuk turunannya

**Soal Ketiga**Apa itu Class Diagram dan UML ? serta dalam 1 (satu) Model UML terdapat 3 bagian, gambarkan dan jelaskan apa saja bagian-bagian tersebut

**Jawaban**

Class diagram adalah model class dalam UML yaitu berbentuk persegi dengan 3 bagian yaitu nama class, properti dan method

UML adalah salah satu tools yang paling banyak digunakan untuk membuat model struktur dan perilaku aplikasi

|  |
| --- |
| Class Name |
| s+Field : type |
| +Method (type) : type |

Nama class

Atribute

Method

Nama Class : yaitu nama class yang akan kita buat

Attribute : yaitu attribute apa saja yang akan kita gunakan didalam

Class tersebut

Method : fungsi yang ada didalam class kita

**Soal Keempat**

Gambarkan relasi-relasi berikut beserta atribut dan method-methodnya!

(pilih minimal 2) !

1. ***Association Berarah:*** Class Pesawat dengan Class Penumpang
2. ***Inheritance:*** Class Hewan dengan Class Kucing dan Anjing
3. ***Realization:*** Class IRoda dengan Class Roda
4. ***Dependency:*** Class Laptop dengan Class Mouse
5. ***Aggregation:*** Class Kamera dengan Class Lensa
6. ***Composition:*** Class Buku dengan Class Penulis

**Jawaban**

|  |
| --- |
| Penumpang |
| +No ticket : int |
| + cara lihat ticket() : void |

1. Association berarah

|  |
| --- |
| Pesawat |
| +jenis maskapai : String |
| +Lepas landas() : void |

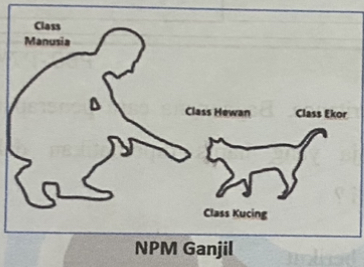
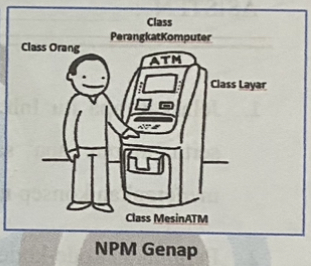
D. Depency

|  |
| --- |
| Mouse |
| +merek : String |
| + on() : void |

|  |
| --- |
| Laptop |
| +Merek : String |
| + Menyalakan() : void |

**Soal Kelima**

Perhatikan Gambar Berikut.

Sebutkan manakah Class yang memiliki relasi Association, Relasi

Aggregation/Composite dan Relasi Inheritance serta berikan contohnya !

**Jawaban**

Class manusia dan Class kucing memiliki assosiasi berarah karena manusia

Memiliki peliharaan kucing

Class hewan dan Class kucing memiliki Inheritance karena kucing termasuk

Dalam class hewan

Class ekor dan Class kucing memiliki aggregation karena ekor itu

Termasuk ada kaitannya dengan class kucing

**Soal Keenam  
*PROGRES STUDI KASUS PROJECT AKHIR :***

Berdasarkan studi kasus kalian coba buatlah Class Diagram (setidaknya

Mengandung relasi Inheritance) kemudian terapkan relasi inheritance

Ke dalam projek code kalian !

**Jawaban**

**Source Code**

|  |
| --- |
| 1. **package** Induk; 3. **import** java.text.SimpleDateFormat; 4. **import** java.util.Scanner; 5. **import** java.util.ArrayList; 6. **import** java.util.Date; 8. **public** **class** Main { 9. **static** ArrayList<AkunEntity> dataKursus = **new** ArrayList(); 10. **static** ArrayList<AkunEntity> dataAkun = **new** ArrayList(); 11. **static** Scanner input = **new** Scanner(System.in); 13. **public** **static** **void** main(String[] args) { 15. System.out.println("Menu"); **int** pil; 16. **do** { 17. System.out.println("1. Daftar Kursus"); 18. System.out.println("2. Login Kursus"); 19. System.out.println("3. Edit Kursus"); 20. System.out.println("4. Hapus Kursus"); 21. System.out.print("PILIH = "); 22. pil = input.nextInt(); **switch** (pil) 23. { 24. **case** 1: 25. daftarAkun(); 26. **break**; 27. **case** 2: 28. **break**; 30. } 32. } **while** (pil != 5); 33. } 35. **static** **void** daftarAkun() { 36. System.out.print("Nama = "); 37. String nama = input.next(); 38. System.out.print("NoTelepon = "); 39. String notelp = input.next(); 40. System.out.print("Alamat = "); 41. String alamat = input.next(); 42. System.out.print("Password = "); 43. String password = input.next(); 44. dataAkun.add(**new** AkunEntity(nama, notelp, alamat, password)); 45. }     51. **static** **void** view(String nama, String password) {  54. **for** (**int** i = 0; i > dataAkun.size(); i++) { 55. **if** (nama.equals(dataAkun.get(i).getNama()) 56. &&password.equals(dataAkun.get(i).getPassword())) { 57. System.out.println("Nama = " + 58. dataAkun.get(i).getNama()); 59. System.out.println("Password = " + 60. dataAkun.get(i).getPassword()); 61. } **else** { 62. System.out.println("Nama atau Password Salah"); 63. } 64. } 65. } 66. } 67. **package** Induk; 69. **public** **class** AkunEntity **extends** projek { 71. **public** AkunEntity(String nama, String notelp, String alamat, String password) { 73. **super**.name = nama; 74. **super**.address = alamat; 75. **super**.noTelepon = notelp; 76. **super**.pw = password; 77. }  80. **public** String getNama() { 81. **return** name; 82. } 84. **public** **void** setNama(String nama) { 85. **this**.name = name; 86. } 88. **public** String getAlamat() { 89. **return** address; 90. } 92. **public** **void** setAlamat(String alamat) { 93. **this**.address = alamat; 94. } 96. **public** String getNotelp() { 97. **return** noTelepon; 98. } 100. **public** **void** setNotelp(String notelp) { 101. **this**.noTelepon = notelp; 102. } 104. **public** String getPassword() { 105. **return** pw; 106. } 108. **public** **void** setPassword(String password) { 109. **this**.pw = password; 110. } 111. } 113. **class** projek{ 115. **public** String name, address, noTelepon, pw ; 116. } |

**Output Program**

|  |
| --- |
|  |