|  |  |
| --- | --- |
| **Nama** | Adidarma Putra Harjuna |
| **NRP** | 5221600027 |
| **Praktikum** | Class dan |

Soal 1

|  |
| --- |
| Soal :  Petunjuk Praktek: 1. Coba contoh sederhana pembuatan kelas BankAccount pada file yang diberikan. Kemudian pelajari struktur kelas dan bagaimana membuat object dari kelas tersebut. Berikan penjelasan pemahaman kalian dari contoh program yang diberikan. |
| Source dan Output Program :   * BankAccount.h   #pragma once  class BankAccount  {  private:      // atribut no Rekening      int AccountNumber;      // atribut SaldoRekening      int AccountBalance;  public:      // constructor      BankAccount(int accNum, int accBalance);      // method untuk menyimpan uang      void Deposit(int money);      // method untuk mengambil uang      void Withdrawal(int money);      // method untuk menampilkan informasi SaldoRekening      void ShowCurrentBalance();      // method untuk memberikan current balance      int GetCurrentBalance();  };   * BankAccount.cpp   #include <iostream>  #include "BankAccount.h"  // implementasi dari constructor  // untuk inisialisasi atribut  BankAccount::BankAccount(int accNum, int accBalance) {      AccountNumber = accNum;      AccountBalance = accBalance;  }  void BankAccount::Deposit(int money) {      AccountBalance = AccountBalance + money;  }  void BankAccount::Withdrawal(int money) {      AccountBalance = AccountBalance - money;  }  void BankAccount::ShowCurrentBalance() {      std::cout << "Current Balance = " << AccountBalance << std::endl;  }  int BankAccount::GetCurrentBalance()  {      return AccountBalance;  }   * main.cpp   #include <iostream>  #include "BankAccount.h"  using namespace std;  BankAccount::BankAccount(int accNum, int accBalance) {      AccountNumber = accNum;      AccountBalance = accBalance;  }  void BankAccount::Deposit(int money) {      AccountBalance = AccountBalance + money;  }  void BankAccount::Withdrawal(int money) {      AccountBalance = AccountBalance - money;  }  void BankAccount::ShowCurrentBalance() {      std::cout << "Current Balance = " << AccountBalance << std::endl;  }  int BankAccount::GetCurrentBalance()  {      return AccountBalance;  }  int main() {      cout << "Hello World!\n";      // instantiate obyek      BankAccount accRizky(1,100000), accYudi(2,200000), accDimas(3,300000);      // tampilkan saldo awal      cout << "Saldo Awal Rizky = " << accRizky.GetCurrentBalance() << endl;      cout << "Saldo Awal Yudi = " << accYudi.GetCurrentBalance() << endl;      // dua object deposit      accRizky.Deposit(100);      accYudi.Deposit(2000);      // tampilkan saldo setelah deposit      cout << "Saldo Setelah deposit Rizky = " << accRizky.GetCurrentBalance() << endl;      cout << "Saldo Setelah deposit Yudi = " << accYudi.GetCurrentBalance() << endl;      // dua object withdrawal      accRizky.Withdrawal(5000);      accYudi.Withdrawal(20000);      // tampilkan saldo setelah withdrawal      cout << "Saldo Setelah ambil Rizky = " << accRizky.GetCurrentBalance() << endl;      cout << "Saldo Setelah ambil Yudi = " << accYudi.GetCurrentBalance() << endl;      int n;      cin >> n;  }   * Output |

Soal 2

|  |
| --- |
| Soal :  2. Buatlah kelas dan obect dari studi kasus dibawah ini: (Simpan program dengan nama MyCharacters.h dan MyCharacters.cpp) "Computer games often contain different characters or creatures. For example, you might design a game in which alien beings possess specific characteristics such as color, number of eyes, or number of lives. Design a character for a game, creating a class to hold at least three attributes for each character. Include methods to get and set each of the character’s attributes. Then write an application in which you create at least two characters each of which has a random age up to 100, a random number of eyes up to 10, and a random number of legs, up to 12. In turn, pass each character to a display method that displays the character’s attributes." |
| Source Code dan Output Program : |