**STUDENT LEARNING JOURNAL 6**

Nama : Nara Atthama

NIM : 162012233034

Course Code : SIR201

Department : FTMM/TRKB

I. OBJECTIVE

|  |
| --- |
| 1. Mahasiswa memahami definisi dari UML Class Diagram  2. Mahasiswa memahami penggunaaan Class Diagram untuk pemrograman berbasis  objek |

II. ACTIVITIES

|  |
| --- |
| 1. Belajar Mandiri mengenai UML Class Diagram 2. Mengerjakan Soal Latihan 3. Mengerjakan Learning Journal |

III. RESULT AND EVALUATION

|  |
| --- |
| UML Class Diagram adalah suatu tools untuk melakukan spesifikasi, visualisasi dari komponen – komponen *software.* Pemodelan menggunakan UML dapat diartikan  dengan menggambarkan bagian-bagian pada dunia nyata yang diterjemahkan menjadi  software menggunakan notasi standart UML.  Tipe UML:   1. Use Case Diagram 2. Class Diagram 3. Object Diagram 4. State Diagram 5. Activity Diagram 6. Sequence Diagram 7. Collaboration Diagram 8. Component Diagram 9. Deployment Diagram   Notasi akses Spesifikasi sebagai berikut :  untuk atribut public/ method   * untuk atribut private / method   Notasi tipe data sebagai berikut :   * Panjang : int * Lebar : int * Jumlah : float   Notasi untuk data parameter, yaitu dengan menambahkan nama dan tipe datanya :   * SetPanjang(P : int) * SetLebar(L : float)   Notasi untuk set data output dari member function adalah dengan menambahkan tipe data setelah data parameter:   * SetPanjang(P : int) : void * SetLebar(L : float) : void * getLuas () : float   **Soal Latihan :**    **Code :**  #include <iostream>  using namespace std;  class Bentuk {      float Lebar;      float Panjang;      float Luas;      public:          Bentuk(){}          Bentuk(float, float);          void setLebar(float);          void setPanjang(float);          float getLebar();          float getPanjang();          float getLuas();  };  Bentuk::Bentuk(float l, float p) {      Lebar = l;      Panjang = p;  }  void Bentuk::setLebar(float L) {      Lebar = L;  }  void Bentuk::setPanjang(float P) {      Panjang = P;  }  float Bentuk::getLebar() {      return Lebar;  }  float Bentuk::getPanjang() {      return Panjang;  }  float Bentuk::getLuas() {      return Lebar \* Panjang;  }  int main(){      Bentuk b(3,4);      Bentuk a;      a.setLebar(5);      a.setPanjang(10);      cout << "Luas a : " << a.getLuas() << endl;      cout << "Luas b : " << b.getLuas() << endl;      return 0;  }  **Hasil Run:** |