**RANCANGAN PELAKSANAAN PEMBELAJARAN**

**INHERITANCE**

**UAS**

Diajukan untuk memenuhi salah satu tugas mata kuliah TELAAH KURIKULUM

Dosen Pengampu: Drs. H. Eka Fitrajaya Rahman M.T



Disusun Oleh:

1505064 Fatwa M Abdillah

**PRODI PENDIDIKAN ILMU KOMPUTER**

**DEPARTEMEN PENDIDIKAN ILMU KOMPUTER**

# FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM

**UNIVERSITAS PENDIDIKAN INDONESIA**

**2018**

**RANCANGAN PELAKSANAAN PEMBELAJARAN**

**(RPP)**

|  |  |  |
| --- | --- | --- |
| Sekolah | : | SMKN 14045 Bandung |
| Mata Pelajaran | : | Pemprograman Berorientasi Objek |
| Pokok Bahasan | : | Inheritance & Polymorphism |
| Kelas | : | 11 |
| Semester | : | 2 (Genap) |
| Alokasi | : | 1 x 45 menit |
| Tahun Pelajaran | : | 2017 / 2018 |
| Pertemuan ke | : | 9 |
| Guru Mata Pelajaran | : | Fatwa M Abdillah |

# A. Kompetensi Inti

KI-1. Menghayati dan mengamalkan ajaran agama yang dianutnya.

KI-2. Menghayati dan Mengamalkan perilaku jujur, disiplin, tanggung jawab, peduli (gotong royong, kerjasama, toleran, damai), santun, responsif dan proaktif dan menunjukan sikap sebagai bagian dari solusi atas berbagai permasalahan dalam berinteraksi secara efektif dengan lingkungan sosial dan alam serta dalam menempatkan diri sebagai cerminan bangsa dalam pergaulan dunia.

KI-3.Memahami, menerapkan dan menganalisis pengetahuan faktual, konseptual, prosedural dan metakognitif berdasarkan rasa ingin tahunya tentang ilmu pengetahuan, teknologi, seni, budaya, dan humaniora dalam wawasan kemanusiaan, kebangsaan, kenegaraan, dan peradaban terkait penyebab fenomena dan kejadian dalam bidang kerja yang spesifik untuk memecahkan masalah.

KI-4.Mengolah, menalar, menyaji dan mencipta dalam ranah konkret dan ranah abstrak terkait dengan pengembangan dari yang dipelajarinya di sekolah secara mandiri, bertindak secara efektif dan kreatif dan mampu melaksanakan tugas spesifik dibawah pengawasan langsung.

# B. Kompetensi Dasar

3.1 Membuat program OOP menggunakan Inheritance pada class

# C. Indikator Pembelajaran

Siswa mampu :

3.1.1 Mendeskripsikan pengertian dari inheritance

3.1.2 Menyebutkan manfaat dari penggunaan inheritance

3.1.3 Mendeskripsikan pengertian dari class parent

3.1.4 Mendeskripsikan pengertian dari class child

3.1.5 Mendeskripsikan pengertian dari @override

3.1.6 Membuat class yang digunakan sebagai parent.

3.1.7 Membuat class child dari class parent yang tersedia

3.1.8 Menerapkan override pada class child yang tersedia

# D. Tujuan Pembelajaran

Setelah mempelajari materi ini, diharapkan siswa dapat memahami bagaimana penggunaan inheritance pada sebuah class.

# E. Materi Pembelajaran

1. Pengenalan inheritance (Class Parent, Class Child, dan Override)
2. Penerapan inheritance dengan sebuah contoh kasus .

# F. Model Pembelajaran

Pendekatan : Saintifik

Model pembelajaran : Team Assisted Individualy

**G. Kegiatan Pembelajaran**

# a. Pendahuluan

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Guru | Siswa | Alokasi Waktu |
| 1 | Mengucapkan salam pembuka dan menyapa kabar peserta didiknya | Menjawab salam | 3 menit |
| 2 | Absen kelas | - |
| 3 | Menyampaikan tujuan pembelajaran hari ini dan apa saja yang akan dipelajari hari ini | Memperhatikan guru |

# b. Kegiatan inti

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tahapan | Guru | | Siswa | Alokasi Waktu |
| Teams | Membentuk tim terdiri atas 4 – 6 anggota dan 1 ketua kelompok | Mengkondisikan tempat duduk sesuai kelompok | | 2 menit | |
| placement test | Menyampaikan materi tentang inheritance | Mengisi LKS sesuai dengan pengamatan kelompok terhadap penyampaian materi guru | | 2 menit | |
| student creative | Guru sebagai fasilitator | Siswa mengisi lembaran perintah yang ada di LKS | | 2 menit | |
| team study | Guru memonitoring proses diskusi | Siswa melakukan diskusi hasil pengerjaan di LKS | | 2 menit | |
| team scores and team recognition | Guru mengarahkan dan menilai jawaban tiap kelompok ke arah yang benar | Siswa mengoreksi jawaban mereka | | 30 menit | |
| whole class units | Guru menyampaikan keseluruhan pembelajaran tentang inheritance | Siswa mengisi LKS dan menuliskan jawaban setelah dikoreksi | | 5 menit | | |

# c. Penutup

|  |  |  |  |
| --- | --- | --- | --- |
| No | Guru | Siswa | Alokasi Waktu |
| 1 | Guru menutup pelajaran hari dengan salam, serta menyampaikan tugas yang nanti dikumpulkan. | - | 1 menit |

**H. Sumber, media, dan alat**

Powerpoint dan modul.

# I. Penilaian

Tes tulis ( Essay ) Soal :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Indikator | Soal | Kunci Jawaban | Keterangan |
| 1. | 3.1.1 | Jelaskan pengertian dari inheritance ! | Inheritance (pewarisan) adalah mekanisme di OOP yang memungkinkan class baru dibuat berdasarkan class yang sudah ada sebelumnya dan mewarisi “sifat” (method dan atribut) dari class parent atau superclass. | Kata kunci “class yang mewarisi sifat parent” |
| 2. | 3.1.2 | Sebutkan 3 manfaat penggunaan inheritance ! | Menyederhanakan syntax  Tidak mengulang syntax yang sama  Meminimalisir kesalahan pemanggilan variable/ atribut/ method  Error handling lebih mudah |  |
| 3. | 3.1.3 | Apa yang dimaksud dengan class parent? | Class utama / class rujukan untuk class lain diambil variable/ atribut/ methodnya. Disebut juga superclass |  |
| 4. | 3.1.4 | Apa yang dimaksud dengan class child? | Class yang merujuk class parent, ditandai dengan *extends* |  |
| 5. | 3.1.5 | Apa yang dimaksud dengan override? | Fungsi yang digunakan class child untuk menimpa isi method dari class parent |  |
| 6 | 3.1.6 | Buatlah sebuah class parent tentang sekolah yang berisi atribut nim, nama, dan nilai ! | **public** **class** Sekolah {  String nim;  String nama;  **double** nilai;  **public** … {  ….  }  } | … diisi bebas |
| 7 | 3.1.7 | Buatlah sebuah class child yang ter-inheritance dengan class parent Sekolah ! | **public** **class** … extends Sekolah{  ….  }  } | … diisi bebas, kata kunci “extends Sekolah” |
| 8 | 3.1.8 | Buat sebuah class child dari class parent berikut untuk mengubah KKN kelulusan menjadi diatas 85 !  **public** **class** Kelulusan {  String nama;  **double** nilai;  String Pelajaran;  **public** String cekLulus() {  String status;  **if** (nilai>75 ) {  status = “Lulus";  } **else** {  status = “Remedial";  }  **return** status;  }  } | **public** **class** … extends Kelulusan {  @Override  public String cekLulus() {  String status;  if (nilai>85 ) {  status = “Lulus";  } else {  status = “Remedial";  }  return status;  }  } | … diisi bebas kata kunci (nilai>85) |

Pedoman Penilaian :

𝑠𝑘𝑜𝑟 𝑠𝑖𝑠𝑤𝑎

|  |  |  |
| --- | --- | --- |
| 𝑁𝑖𝑙𝑎𝑖 𝑠𝑖𝑠𝑤𝑎 =  𝑆𝑘𝑜𝑟 𝑀𝑎𝑘𝑠𝑖𝑚𝑢𝑚 𝐼𝑑𝑒𝑎𝑙 (𝑆𝑀𝐼)  Standar pencapaian indikator : | | x 100 |
| KKM = 65,00 |  |
| **Mengetahui,** |  | **Bandung, April 2018** |
| **Kepala sekolah** |  | **Guru mata pelajaran Pemprograman Berorientasi Objek** |
|  |  |  |
| ***NAMA KEPALA SEKOLAH*** |  | **Fatwa M Abdillah** |
| ***NIP KEPALA SEKOLAH*** |  | ***NIP GURU*** |

**LEMBAR KERJA SISWA**

**(LKS)**

**Mata Pelajaran : Pemrograman Berorientasi Objek**

**Materi : Inheritance**

Inheritance (pewarisan) adalah mekanisme di OOP yang memungkinkan class baru dibuat berdasarkan 32 class yang sudah ada sebelumnya dan mewarisi “sifat” (method dan atribut) dari class parent atau superclass.

Sebagai contoh, akan diturunkan kelas MahasiswaS3 dari kelas Mahasiswa. Class MahasiswaS3 yang baru ini akan mewarisi atribut dan method milik Mahasiswa, tapi juga bisa menambahkan atribut atau method yang baru. Class Mahasiswa akan disebut sebagai **superclass** dan class MahasiswaS3 adalah **subclass**.

Sekolah

Siswa

diturunkan

ditambah

di-override/ ditimpa

Pertama buat class Sekolah sebagai berikut:

**public** **class** Sekolah {

String nama;  
 **double** nilai;

**public** String cekLulus() {  
 String status;  
 **if** (nilai>75 ) {  
 status = “Lulus";  
 } **else** {  
 status = “Remedial";

}  
 **return** status;  
 }

}

Coba jalankan untuk melihat bagaimana cekLulus dihitung. Kemudian buatlah class baru, Siswa yang merupakan turunan dari Mahasiswa. Jangan lupa tambahkan keyword “extends” di code.

**public** **class** Siswa **extends** Sekolah {

**public** **static** **void** main(String[] args) {  
 Siswa sis = **new** Siswa();  
 String Pelajaran;   
 sis.nama = “Fatwa”;  
 sis.nilai = 83.6;  
 sis.pelajaran = “Bahasa Indonesia”;  
 System.*out*.println(“Nama : ” + sis.nama);  
 System.*out*.println(“Mapel: ” + pelajaran);  
 System.*out*.println(sis.cekLulus());  
 }

}

Dapat dilihat pada kode diatas bahwa atribut seperti nama, nilai tidak perlu dideklarasikan ulang. Semua atribut dan method pada superclass dapat diakses oleh subclass.

**HASIL PENGAMATAN**

Manfaat Inheritance :

……………………………………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………………………………………………………

Class Parent : ……………………………………………………………………………………………………………………………………………………………………………………………………

Class Child : ……………………………………………………………………………………………………………………………………………………………………………………………………

Override : ……………………………………………………………………………………………………………………………………………………………………………………………………

PENUGASAN KELOMPOK :

Buatlah Contoh Penggunaan Class Parent dan Class Child dalam permasalahan di sekolah :

Nama Aplikasi : ………………………………………………………………………………

Class Parent :

**public** **class** Sekolah {

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

}

Class Child :

**public** **class** …………………………… extends Sekolah {

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

…………………………………………………………………

}