

	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	1 dari 17

## Rencana Pembelajaran Semester (RPS)

Prodi D4 Teknik Informatika  
Jurusan Teknik Komputer dan Informatika

1. Nama Matakuliah : Teknik Pemrograman
2. Kode/ Beban SKS : 16TIN2054/ 4 SKS
3. Semester : Genap / II
4. Prasyarat : Dasar-dasar Pemrograman
5. Status Matakuliah : Mata Kuliah Program Studi D4
6. Bentuk Pembelajaran : Kelas / ~~Seminar~~ / Praktikum
7. Dosen : Ghifari Munawar, S.Kom., M.T

### 1. Deskripsi Mata Kuliah / Course (catalog) description :

Matakuliah Teknik Pemrograman pada program studi D4 Teknik Informatika Politeknik Negeri Bandung merupakan matakuliah jurusan Teknik Komputer dan Informatika. Matakuliah Teknik Pemrograman ini berfokus/menekankan pada teknik pemrograman (sebagai lanjutan dari DDP), oo fundamental, serta merancang program berbasis objek (fan-in, fan-out, kohesi, kopling, dll). Matakuliah ini akan dibagi dalam beberapa kajian, yaitu fundamental oo, prinsip desain oo, *collection framework*, *defensive programming*, *assertion & unit testing*, serta *refactoring*. Matakuliah Teknik Pemrograman ini akan dilakukan selama 14 minggu, dengan bentuk pembelajaran **di kelas (kuliah, responsi, dan tutorial)** dan **di laboratorium (praktikum)** yang meliputi tatap muka, penugasan terstruktur serta belajar mandiri. Mahasiswa akan mendapatkan pengalaman belajar antara lain : menyimak – memahami konsep – diskusi – studi referensi, penelusuran informasi dari sumber referensi lain (website/blog/jurnal), serta praktik pemrograman di laboratorium.

### 2. Prasyarat / prerequisites:

Mahasiswa sudah pernah mengikuti dan lulus matakuliah Dasar-dasar Pemrograman (DDP).

	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	2 dari 17

### 3. Referensi / Textbook(s) and/or other required materials:

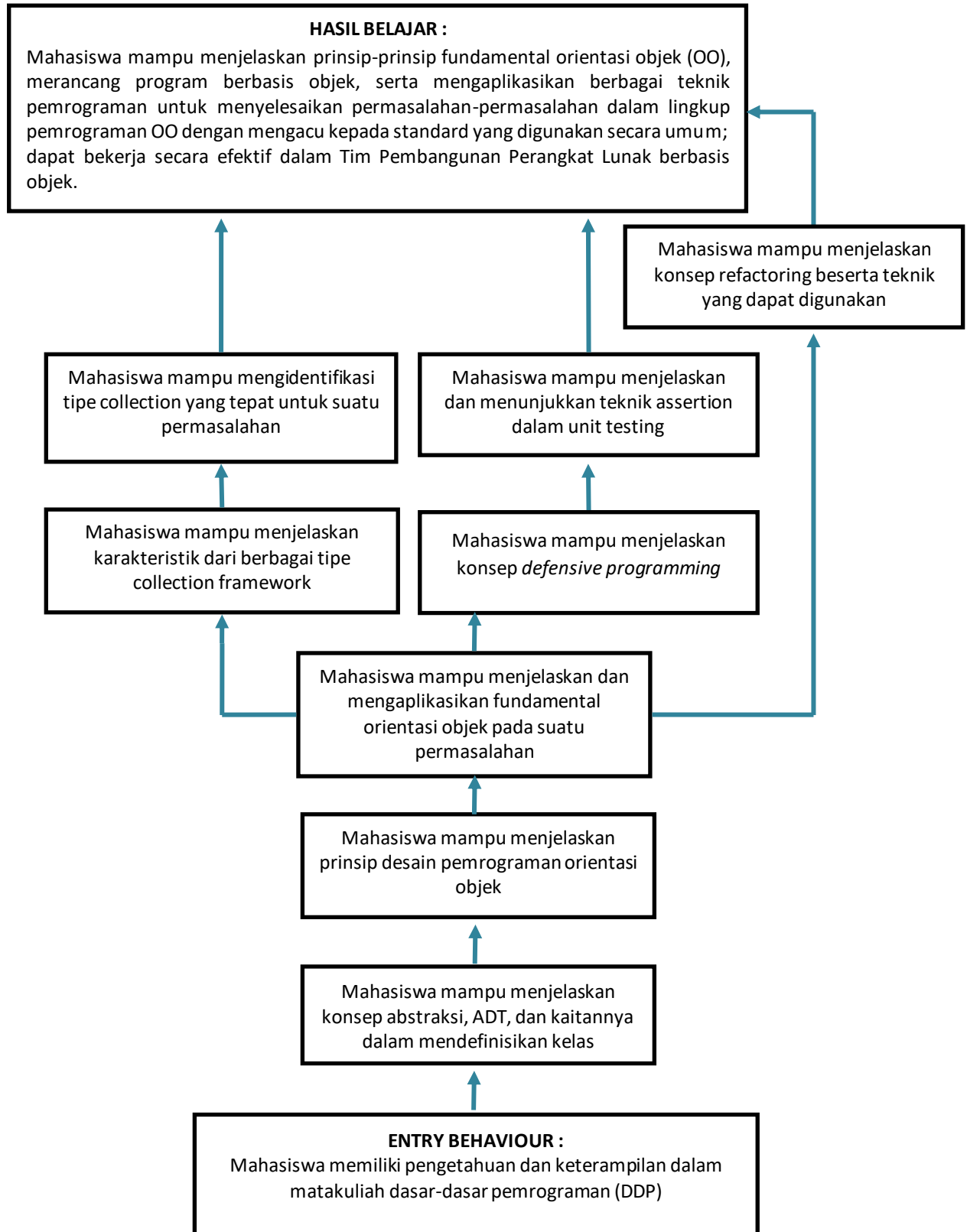
1. Van Roy, Peter, dkk. "Concepts, Techniques, and Models of Computer Programming". 2003.
2. McConnell, Steven C. "Code Complete 2nd Edition", Microsoft Press. 2004.
3. Sierra, Khaty, dkk. "Head First Java Second Edition", O'Reilly. 2004
4. Poo, Danny, dkk. "Object Oriented Programming and Java Second Edition", Springer. 2007.
5. Martin, Robert C. "Clean Code – A Handbook of Agile Software Craftmanship", Prentice Hall. 2008.
6. Freeman, Eric, dkk. "Head First Design Patterns", O'Reilly. 2004

### 4. Capaian Pembelajaran Lulusan / Learning Outcomes :

Capaian Pembelajaran Lulusan	
KU1	Mampu menerapkan pemikiran logis, kritis, inovatif, bermutu, dan terukur dalam melakukan pekerjaan yang spesifik di bidang pengembangan perangkat lunak serta sesuai dengan standar kompetensi kerja bidang yang bersangkutan.
KU2	Mampu menunjukkan kinerja mandiri, bermutu dan terukur.
KK1	Mampu menerapkan matematika, sains, dan prinsip rekayasa dalam pengembangan perangkat lunak memanfaatkan prinsip desain dalam menyelesaikan masalah di bidang teknologi dan komputasi.
KK2	Mampu mengidentifikasi dan menerapkan prinsip-prinsip komputasi dalam menganalisa dan menghasilkan alternatif solusi yang efektif menggunakan perangkat lunak yang mengacu standar pengembangan perangkat lunak berbasis standar ISO 12207 (Software Development Process).
PP4	Menguasai pengetahuan tentang codes dan standard dalam menjaga kualitas pengembangan perangkat lunak.
PP7	Menguasai pengetahuan tentang perkembangan teknologi actual

	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	3 dari 17

## 5. Peta Konsep :



	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	4 dari 17

## 6. Hasil Belajar / Course objectives:

Mahasiswa mampu menjelaskan prinsip-prinsip fundamental orientasi objek (OO), merancang program berbasis objek, serta mengaplikasikan berbagai teknik pemrograman untuk menyelesaikan permasalahan-permasalahan dalam lingkup pemrograman OO dengan mengacu kepada standard yang digunakan secara umum; dapat bekerja secara efektif dalam Tim Pembangunan Perangkat Lunak berbasis objek.

Setelah mengikuti matakuliah Teknik Pemrograman (16TIN2054), mahasiswa mampu :

- a. menjelaskan konsep abstraksi, *abstract data type* (ADT), dan kaitannya dalam mendefinisikan kelas.
- b. menjelaskan fundamental pemrograman berorientasi objek;
- c. menunjukkan dan mengaplikasikan teknik pemrograman berorientasi objek pada suatu kasus sederhana;
- d. menjelaskan prinsip desain : fan-in, fan-out, cohesion dan coupling;
- e. menjelaskan karakteristik dari berbagai tipe *collection*;
- f. mengidentifikasi tipe *collection* yang tepat untuk digunakan pada suatu kasus sederhana;
- g. menjelaskan konsep *defensive programming* beserta teknik yang dapat digunakan;
- h. menjelaskan dan menunjukkan teknik *assertion* dalam *unit testing*
- i. menjelaskan konsep *refactoring* beserta teknik yang dapat digunakan.

## 7. Bahan Kajian / Topics covered :

1. OOP Fundamental;
  - a. Abstraction Concept & ADT
  - b. Object & Class
  - c. Classification : Generalization & Specialization
  - d. Inheritance
  - e. Abstract Class & Interface

	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	5 dari 17

- f. Polymorphism
- 2. OO Design Principles;
  - a. Fan in / Fan out
  - b. Cohesion / Coupling
- 3. Generic Programming & Collection Framework (List, Set, Map);
- 4. Defensive Programming;
  - a. Error handling
  - b. Exception
- 5. Assertion & Unit Testing;
- 6. Refactoring
  - a. Clean Code
  - b. Introduction to Design Pattern

## 8. Metode Pembelajaran dan Kode Etik Perkuliahan:

**Perkuliahan** Perkuliahan meliputi kuliah, responsi dan tutorial yang dititikberatkan pada peran aktif mahasiswa. Perkuliahan didesain dalam 3 bagian yang melibatkan peran aktif mahasiswa yaitu :

**Tatap muka** berupa ceramah tentang bahan kajian, tanya-jawab antara dosen-mahasiswa, diskusi (mahasiswa secara aktif melakukan diskusi) dan presentasi hasil diskusi;

**Penugasan Terstruktur** berupa kegiatan yang berpusat pada mahasiswa untuk mendapatkan kemampuan yang disasar dengan arahan dosen dan diberikan umpan-balik. Pengalaman belajar mahasiswa dapat berupa latihan soal, tugas (mandiri dan/atau kelompok), pengamatan di lapangan dan bentuk terstruktur lainnya);

**Belajar mandiri** berupa kegiatan mandiri yang dilakukan mahasiswa untuk mencapai hasil belajar mata kuliah

	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	6 dari 17

dengan/tanpa arahan dosen.

**Praktikum** Praktikum dilakukan mahasiswa di kelas dengan lingkungan terkondisi seperti laboratorium. Praktikum bertujuan untuk mengenalkan dan membiasakan mahasiswa atas proses/perilaku /urutan kerja/tahapan produksi dan lainnya.


**Tugas** Tugas akan diberikan setiap pertemuan sebagai latihan dalam memahami bahasan teori yang telah diberikan selama perkuliahan berlangsung. Tugas terbagi menjadi individu, dan kelompok. Tugas kelompok akan diberikan beberapa kali selama satu semester untuk menumbuhkan semangat belajar dan berkembang bersama.

**Kuis** Selama satu semester dapat diadakan beberapa kali tes/kuis secara mendadak, dengan bahan kuis berupa bahan kajian yang telah dibahas pada kuliah sebelumnya maupun bahan kajian yang akan dibahas pada saat kuliah. Dalam rencana pembelajaran semester, kuis disebutkan secara *tentative*.

**Kehadiran** Mahasiswa wajib hadir di kegiatan belajar selama satu semester minimal 85% dari total kehadiran sebagai syarat mengikuti UAS. Apabila mahasiswa tidak hadir, harus ada keterangan yang menjelaskan ketidakhadirannya tersebut. Keterlambatan mahasiswa hanya bisa ditolerir selama 15 menit, keterlambatan setelah itu mahasiswa dinyatakan tidak masuk.

**Kode etik** Segala kecurangan (*cheating*, plagiat, *copy-paste* dan sejenisnya) tidak akan ditoleransi. Apabila mahasiswa terbukti melakukan perbuatan tersebut, maka akan didiskualifikasi dari kelas dengan nilai maksimal D.

**Evaluasi Kualitas** Kualitas perkuliahan akan diukur dengan menggunakan

	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	7 dari 17

**Kuliah** kuesioner yang akan dibagikan ke mahasiswa pada awal dan akhir perkuliahan. Kuesioner didesain untuk mengetahui apakah mahasiswa mempunyai perbedaan persepsi sebelum dan setelah mengikuti perkuliahan ini.

### **9. Jadwal Perkuliahan / Class/laboratory schedule (number of sessions per week and duration of session):**

Jumlah perkuliahan selama 16 minggu (termasuk UTS, dan UAS), dengan jumlah jam perkuliahan sebagai berikut :

- Tatap muka : 3 x 50 menit (3 sks teori)
- Belajar mandiri : 2 x 60 menit
- Praktikum dan Penugasan Terstruktur : 1 x 170 menit (1 sks praktek)

### **10. Rencana Pembelajaran Semester / .....**

	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	8 dari 17

## RENCANA PEMBELAJARAN SEMESTER

**Nama Mata Kuliah :** Teknik Pemrograman      **Kode MK** : 16TIN2054    **Semester :** 2      **SKS :** 4  
**Jurusan** : Teknik Komputer dan Informatika      **Prodi** : D-IV Teknik Informatika

**Dosen Pengampu** : Ghifari Munawar, S.Kom., M.T.

### Capaian Pembelajaran Lulusan yang dibebankan :

1. Mampu menerapkan pemikiran logis, kritis, inovatif, bermutu, dan terukur dalam melakukan pekerjaan yang spesifik di bidang pengembangan perangkat lunak serta sesuai dengan standar kompetensi kerja bidang yang bersangkutan [KU1].
2. Mampu menunjukkan kinerja mandiri, bermutu dan terukur [KU2].
3. Mampu menerapkan matematika, sains, dan prinsip rekayasa dalam pengembangan perangkat lunak memanfaatkan prinsip desain dalam menyelesaikan masalah di bidang teknologi dan komputasi [KK1].
4. Mampu mengidentifikasi dan menerapkan prinsip-prinsip komputasi dalam menganalisa dan menghasilkan alternatif solusi yang efektif menggunakan perangkat lunak yang mengacu standar pengembangan perangkat lunak berbasis standar ISO 12207 (Software Development Process) [KK2].
5. Menguasai pengetahuan tentang codes dan standard dalam menjaga kualitas pengembangan perangkat lunak [PP4].
6. Menguasai pengetahuan tentang perkembangan teknologi actual [PP7].

### Hasil Belajar Mata Kuliah Yang Diharapkan :

Mahasiswa mampu menjelaskan prinsip-prinsip fundamental orientasi objek (OO), merancang program berbasis objek, serta mengaplikasikan berbagai teknik pemrograman untuk menyelesaikan permasalahan-permasalahan dalam lingkup pemrograman OO dengan mengacu kepada standard yang digunakan secara umum; dapat bekerja secara efektif dalam Tim Pembangunan Perangkat Lunak berbasis objek.



	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	9 dari 17

Minggu ke	Kemampuan Akhir Yang Diharapkan	Bahan Kajian	Bentuk Pembelajaran	Alokasi Waktu	Kriteria (Indikator Capaian)	Instrumen Penilaian	Bobot	Referensi
1	2	3	4	5	6	7	8	9
1	<ul style="list-style-type: none"> <li>Mahasiswa dapat mengetahui RPS matakuliah Teknik Pemrograman</li> <li>Mahasiswa dapat mengetahui e-learning matakuliah Teknik Pemrograman</li> <li>Mahasiswa dapat menjelaskan konsep abstraksi dan kaitannya dalam mendefinisikan kelas</li> </ul>	<ul style="list-style-type: none"> <li>RPS matakuliah Teknik Pemrograman.</li> <li>E-learning matakuliah Teknik Pemrograman</li> <li>Konsep Abstraksi</li> <li>Abstract Data Type (ADT)</li> <li>Kaitan ADT dalam pendefinisian kelas</li> </ul>	<b>Tatap muka :</b> Ceramah;	3 x 50 menit	Tidak ada	-	-	RPS, dan [1]
			<b>Praktikum dan Penugasan Terstruktur :</b> <ul style="list-style-type: none"> <li>Menginstall dan mengeksplorasi perangkat lunak bantu pemrograman berbasis objek</li> <li>Melakukan registrasi e-learning matakuliah Teknik Pemrograman</li> <li>Mengeksplorasi <i>resources</i> e-learning matakuliah Teknik Pemrograman</li> </ul>	1 x 170 menit	Mahasiswa dapat menunjukkan hasil eksplorasi			
2	<ul style="list-style-type: none"> <li>Mahasiswa dapat menjelaskan konsep objek</li> <li>Mahasiswa dapat menjelaskan konsep kelas.</li> <li>Mahasiswa dapat merancang kelas pada suatu permasalahan</li> <li>Mahasiswa dapat mengaplikasikan kelas pada bahasa pemrograman objek</li> </ul>	<ul style="list-style-type: none"> <li>Classes as a complete ADT</li> <li>Konsep objek &amp; kelas</li> <li>Desain Kelas (attributes, operations, message)</li> </ul>	<b>Tatap muka :</b> Ceramah;	3 x 50 menit	Mahasiswa dapat menjelaskan konsep kelas & objek	Rubrik portofolio	5%	[1], dan [4]
			<b>Praktikum dan Penugasan Terstruktur :</b> <ul style="list-style-type: none"> <li>Mengeksplorasi bahasa pemrograman objek</li> <li>Mengesplorasi sintak dalam pembuatan kelas &amp; objek</li> </ul>	1 x 170 menit	Mahasiswa dapat merancang dan mengaplikasikan kelas pada suatu permasalahan			

	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	10 dari 17

			<ul style="list-style-type: none"> <li>Mengerjakan kasus yang dituliskan pada e-learning</li> </ul>					
3	<ul style="list-style-type: none"> <li>Mahasiswa dapat menjelaskan prinsip-prinsip desain pemrograman orientasi objek</li> </ul>	Prinsip desain orientasi objek yang mencakup : <ul style="list-style-type: none"> <li>Cohesion</li> <li>Coupling</li> <li>Fan-in/ out</li> </ul>	<b>Tatap muka :</b> Ceramah;  <b>Praktikum dan Penugasan Terstruktur :</b> Mahasiswa mengeksplorasi permasalahan pemrograman oo terkait high cohesion, low cohesion, high coupling, dan low coupling.	2 x 3 x 50 menit  2 x 1 x 170 menit	Mahasiswa dapat menjelaskan prinsip desain oo : high cohesion dan low coupling	Rubrik holistik	5%	[2]
4	<ul style="list-style-type: none"> <li>Mahasiswa dapat menjelaskan konsep classification.</li> <li>Mahasiswa dapat menjelaskan konsep generalisasi.</li> <li>Mahasiswa dapat menjelaskan konsep spesialisasi.</li> <li>Mahasiswa dapat mendefinisikan konsep generalisasi, spesialisasi pada suatu permasalahan.</li> </ul>	<ul style="list-style-type: none"> <li>Classification : Generalization &amp; Specialization</li> </ul>	<b>Tatap muka :</b> Ceramah;  <b>Praktikum dan Penugasan Terstruktur :</b> <ul style="list-style-type: none"> <li>Mendefinisikan konsep generalisasi &amp; spesialisasi pada suatu permasalahan.</li> <li>Mengerjakan kuis online terkait "classification" pada e-learning</li> </ul>	3 x 50 menit  1 x 170 menit	Mahasiswa dapat menjelaskan konsep classification, generalisasi, dan spesialisasi  Mahasiswa dapat mendefinisikan konsep generalisasi & spesialisasi pada suatu permasalahan.	Rubrik holistik	5%	[3], dan [4]
5	<ul style="list-style-type: none"> <li>Mahasiswa dapat menjelaskan konsep</li> </ul>	<ul style="list-style-type: none"> <li>Inheritance</li> <li>Constructor</li> </ul>	<b>Tatap muka :</b> Ceramah;	3 x 50 menit	Mahasiswa dapat menjelaskan konsep inheritance	Rubrik portofolio	5%	[3], dan [4]

	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	11 dari 17

	<i>inheritance</i> . <ul style="list-style-type: none"> <li>Mahasiswa dapat menjelaskan teknis pembuatan <i>constructor</i>.</li> <li>Mahasiswa dapat menjelaskan teknis pembuatan <i>method overloading</i>.</li> <li>Mahasiswa dapat mendefinisikan konsep inheritance pada suatu permasalahan</li> </ul>	<ul style="list-style-type: none"> <li>Method overloading</li> </ul>	<b>Praktikum dan Penugasan Terstruktur :</b> <ul style="list-style-type: none"> <li>Mengeksplorasi kasus inheritance</li> <li>Mengerjakan soal latihan inheritance pada e-learning</li> </ul>	1 x 170 menit	Mahasiswa dapat mendefinisikan konsep inheritance pada suatu permasalahan			
6	<ul style="list-style-type: none"> <li>Mahasiswa dapat menjelaskan dan menunjukkan konsep abstract class</li> <li>Mahasiswa dapat menjelaskan dan menunjukkan konsep interface</li> <li>Mahasiswa dapat menjelaskan konsep multiple inheritance</li> </ul>	<ul style="list-style-type: none"> <li>Abstract Class</li> <li>Interface</li> </ul>	<b>Tatap muka :</b> Ceramah;	3 x 50 menit	Mahasiswa dapat menjelaskan konsep multiple inheritance	Rubrik holistik	5%	[3], dan [4]
			<b>Praktikum dan Penugasan Terstruktur :</b> <ul style="list-style-type: none"> <li>Mahasiswa mengeksplorasi implementasi abstract class &amp; interface</li> <li>Mengerjakan soal latihan abstract class &amp; interface pada e-learning</li> </ul>	1 x 170 menit	Mahasiswa dapat menunjukkan konsep abstract class dan interface pada suatu permasalahan			
7	<ul style="list-style-type: none"> <li>Mahasiswa dapat menjelaskan konsep polymorphism</li> <li>Mahasiswa dapat menjelaskan teknik pembuatan method</li> </ul>	<ul style="list-style-type: none"> <li>Polymorphism</li> <li>Method overriding</li> <li>Polymorphism via inheritance</li> <li>Polymorphism via interface</li> </ul>	<b>Tatap muka :</b> Ceramah;	3 x 50 menit	Mahasiswa dapat menjelaskan konsep polymorphism	Rubrik holistik	5%	[3], dan [4]
			<b>Praktikum dan Penugasan Terstruktur :</b>	1 x 170 menit	Mahasiswa dapat menunjukkan			

	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	12 dari 17

	overriding • Mahasiswa dapat menunjukkan konsep polymorphism via inheritance • Mahasiswa dapat menunjukkan konsep polymorphism via interface		• Mahasiswa mengeksplorasi kasus polymorphism • Mahasiswa mengeksplorasi teknik polymorphism via inheritance dan teknik polymorphism via interface • Mengerjakan soal latihan “polymorphism” pada e-learning		konsep polymorphism via inheritance dan polymorphism via interface			
8	<b>UTS</b>					Naskah UTS	20%	[1], [2], [3], dan [4]
9	• Mahasiswa dapat menjelaskan konsep generic programming • Mahasiswa dapat menunjukkan teknik generic programming	Generic programming	<b>Tatap muka :</b> Ceramah; <b>Praktikum dan Penugasan Terstruktur :</b> <ul style="list-style-type: none"> <li>Mengeksplorasi implementasi generic programming</li> <li>Mengerjakan soal latihan “generic programming” pada e-learning</li> </ul>	3 x 50 menit 1 x 170 menit	Mahasiswa dapat mengaplikasikan generic programming pada suatu permasalahan	Rubrik portofolio	5%	[2], [3]
10	• Mahasiswa dapat menjelaskan konsep java collection framework • Mahasiswa dapat menjelaskan berbagai karakteristik collection framework • Mahasiswa dapat memilih collection framework yang tepat untuk suatu	Java Collection Framework : <ul style="list-style-type: none"> <li>List</li> <li>Set</li> <li>Map</li> </ul>	<b>Tatap muka :</b> Ceramah; <b>Praktikum dan Penugasan Terstruktur :</b> <ul style="list-style-type: none"> <li>Mahasiswa mengeksplorasi teknis pembuatan list, set, map pada bahasa pemrograman objek</li> <li>Mahasiswa mengidentifikasi collection framework yang</li> </ul>	3 x 50 menit 1 x 170 menit	Mahasiswa dapat menunjukkan collection framework yang tepat pada suatu permasalahan	Rubrik portofolio	5%	[3]

	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	13 dari 17

	permasalahan		tepat pada suatu permasalahan • Mengerjakan soal latihan “collection framework” pada e-learning					
11	<ul style="list-style-type: none"> <li>• Mahasiswa dapat menjelaskan konsep defensive programming</li> <li>• Mahasiswa dapat menjelaskan teknik assertion</li> <li>• Mahasiswa dapat menjelaskan teknik error handling</li> <li>• Mahasiswa dapat menjelaskan teknik exceptions</li> </ul>	Defensive programming • Assertion • Error handling • Exceptions	<b>Tatap muka :</b> Ceramah;	3 x 50 menit	Mahasiswa dapat menjelaskan dan menunjukkan teknik defensive programming serta implementasinya pada suatu permasalahan	Rubrik portofolio	5%	[2], dan [3]
			<b>Praktikum dan Penugasan Terstruktur :</b> • Mahasiswa mengeksplorasi teknis assertion, error handling, exceptions • Mengerjakan soal latihan “defensive programming” pada e-learning	1 x 170 menit				
12	<ul style="list-style-type: none"> <li>• Mahasiswa dapat menjelaskan konsep unit testing</li> <li>• Mahasiswa dapat menunjukkan penggunaan junit</li> <li>• Mahasiswa dapat menunjukkan hasil unit testing</li> </ul>	Unit Testing dengan Junit	<b>Tatap muka :</b> Ceramah;	3 x 50 menit	Mahasiswa dapat menjelaskan dan menunjukkan konsep unit testing dengan junit	-	-	[2], dan [3]
			<b>Praktikum dan Penugasan Terstruktur :</b> Mahasiswa mengeksplorasi unit testing menggunakan junit	1 x 170 menit				
13	<ul style="list-style-type: none"> <li>• Mahasiswa dapat menjelaskan konsep</li> </ul>	Refactoring, dan Teknik Code Tuning	<b>Tatap muka :</b> Ceramah;	3 x 50 menit	Mahasiswa dapat mengidentifikasi teknik refactoring	Rubrik holistik	5%	[2]

	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	14 dari 17

	<ul style="list-style-type: none"> <li>refactoring</li> <li>Mahasiswa dapat menjelaskan strategi dalam refactoring</li> <li>Mahasiswa dapat menjelaskan teknik code tuning</li> </ul>		<b>Praktikum dan Penugasan Terstruktur :</b> <ul style="list-style-type: none"> <li>Mahasiswa mengeksplorasi konsep refactoring dan code tuning</li> <li>Mengerjakan quiz “refactoring &amp; code tuning” pada e-learning</li> </ul>	1 x 170 menit	dan code tuning dalam suatu permasalahan			
14	<ul style="list-style-type: none"> <li>Mahasiswa dapat menjelaskan konsep design pattern</li> <li>Mahasiswa dapat menyebutkan jenis-jenis design pattern (structural, behavioral dan creational)</li> <li>Mahasiswa dapat mengidentifikasi design pattern yang sesuai untuk digunakan pada suatu permasalahan</li> </ul>	Pengenalan Design pattern : <ul style="list-style-type: none"> <li>Structural</li> <li>Creational</li> <li>Behavioral</li> </ul>	<b>Tatap muka :</b> Ceramah;	3 x 50 menit	Mahasiswa dapat mengidentifikasi design pattern yang sesuai untuk permasalahan yang dihadapi	Rubrik holistik	5%	[6]
			<b>Praktikum dan Penugasan Terstruktur :</b> <ul style="list-style-type: none"> <li>Mahasiswa mempelajari buku “head first design pattern”</li> <li>Mahasiswa mencoba mengaplikasikan berbagai jenis design pattern</li> <li>Mempelajari contoh kasus penerapan design pattern pada e-learning</li> </ul>	1 x 170 menit				
15	<ul style="list-style-type: none"> <li>Mahasiswa dapat menjelaskan konsep clean code</li> <li>Mahasiswa dapat menjelaskan kemungkinan permasalahan yang timbul ketika tidak menerapkan clean code</li> <li>Mahasiswa dapat</li> </ul>	Clean code	<b>Tatap muka :</b> Ceramah;	3 x 50 menit	Mahasiswa dapat menjelaskan beberapa teknik clean code yang dapat diterapkan	Rubrik holistik	5%	[5]
			<b>Praktikum dan Penugasan Terstruktur :</b> <ul style="list-style-type: none"> <li>Mahasiswa mempelajari buku “clean code”</li> <li>Mengerjakan soal “clean code” pada e-learning</li> </ul>	1 x 170 menit				

	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	15 dari 17

	menjelaskan teknik clean code							
16	UAS					Naskah UAS	20%	[2], [3], [5], dan [6]

Daftar Pustaka	
Utama	Pendukung
1. Van Roy, Peter, dkk. "Concepts, Techniques, and Models of Computer Programming". 2003. 2. McConnell, Steven C. "Code Complete 2nd Edition", Microsoft Press. 2004. 3. Sierra, Khaty, dkk. "Head First Java Second Edition", O'Reilly. 2004 4. Poo, Danny, dkk. "Object Oriented Programming and Java Second Edition", Springer. 2007. 5. Martin, Robert C. "Clean Code – A HandBook of Agile Software Craftmanship", Prentice Hall. 2008. 6. Freeman, Eric, dkk. "Head First Design Patterns", O'Reilly. 2004	1. Java Note 2. Contoh-contoh kasus untuk latihan yang diambil dari berbagai sumber (web, forum, online resources, dll)
Sarana Pembelajaran	
Hardware	Software
1. LCD projector 2. Laptop / Komputer 3. VGA to HDMI Converter 4. Pen Pointer	1. Netbeans / Eclipse 2. JDK 8 3. JUnit
Prasarana Pembelajaran	
1. Kelas 2. Laboratorium Rekayasa Perangkat Lunak	

	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	16 dari 17

3. Papan Tulis

OTORISASI	Koordinator Mata Kuliah	Koordinator RPS	Ketua Prodi D4 TI
Revisi Tanggal : 25 Oktober 2018	GHIFARI MUNAWAR, S.KOM.,M.T.		SANTISUNDARI, M.T



	<b>RENCANA PEMBELAJARAN SEMESTER</b>	No. Dokumen	RPS/R1/16TIN3063/2016
		Berlaku Sejak	1 September 2016
	<b>PROGRAM SARJANA TERAPAN TEKNIK INFORMATIKA</b>	Revisi	01
		Halaman	17 dari 17

## RUBRIK YANG DIGUNAKAN UNTUK ASESMEN

### A. RUBRIK HOLISTIK

	Sangat Baik	Baik	Cukup	Kurang
Dimensi	Skor 81-100	Skor 71-80	Skor 61-70	Skor 60
<b>Penguasaan Materi</b>	Permasalahan dan solusi saling memenuhi dan diungkap dengan jelas	Permasalahan jelas, namun solusi kurang sesuai untuk menyelesaikan masalah	Permasalahan kurang jelas, solusi dapat ditangkap sebagai penyelesaian beberapa masalah	Permasalahan kurang jelas demikian pula solusinya
<b>Ketepatan menyelesaikan masalah</b>	Ide, jelas, inovatif, dan mampu menyelesaikan masalah dengan cakupan luas	Ide yang dikemukakan jelas, mampu menyelesaikan masalah, inovatif, cakupan tidak terlalu luas	Ide yang dikemukakan jelas dan sesuai, namun kurang inovatif	Ada ide yang dikemukakan, namun kurang sesuai dengan permasalahan
<b>Kemampuan menghadapi Pertanyaan</b>	Menangkap pertanyaan dan memberi jawaban yang tepat	Menangkap pertanyaan, namun jawaban yang diberikan kurang tepat	Menangkap pertanyaan, namun tidak dapat memberikan jawaban	Kurang dapat menangkap pertanyaan dan tidak dapat merespon

### B. PORTOFOLIO

Dimensi	Sangat Baik Skor 81-100	Baik Skor 71-80	Cukup Skor 61-70	Kurang Skor 60
<b>Laporan Hasil Pekerjaan</b>	Isi mampu menjawab permasalahan yang dihadapi dengan solusi yang tepat dan efektif	Isi mampu menjawab permasalahan yang dihadapi dengan solusi yang tepat, namun belum efektif	Isi menjawab permasalahan dengan solusi yang kurang tepat	Isi tidak mampu menjawab permasalahan yang ada
<b>Source code Aplikasi</b>	Source code terorganisasi dengan baik dengan algoritma yang sesuai dengan persoalan dan mampu menampilkan hasil screenshot aplikasi dengan menyusun skenario pengujiannya	Source code belum terorganisasi dengan baik, namun algoritma mampu menyelesaikan persoalan, dan hasil screenshot dapat ditampilkan	Source code tidak terorganisasi dengan baik dan solusi yang digunakan kurang tepat, hasil screenshot ditampilkan apa adanya	Source code tidak dapat menjawab persoalan, masih muncul bug (kesalahan kode), hasil pekerjaan tidak dapat ditampilkan