BUKU RANCANGAN PENGAJARAN (BRP) MATA KULIAH

PEMROGRAMAN FUNGSIONAL

oleh

Dr. Ade Azurat

Program Studi Ilmu Komputer

Fakultas Ilmu Komputer
Universitas Indonesia Depok, Juli 2020

UNIVERSITAS INDONESIA
FAKULTAS ILMU KOMPUTER
PROGRAM STUDI SARJANA ILMU KOMPUTER

BUKU RANCANGAN PENGAJARAN

MATA KULIAH (MK)	Pemrograman Fungsional	**BOBOT (sks)**	**MK yang menjadi prasyarat**	**Menjadi prasyarat untuk MK**	**Integrasi Antar MK**
KODE	CSCE604123	4 sks	Matematika Diskrit 1, Struktur Data dan Algoritma	-	Dasar-dasar Pemrograman, Pemrograman Lanjut, Proyek Perangkat Lunak
Rumpun MK	-				
Semester	5				
Dosen Pengampu	Dr. Ade Azurat				
Deskripsi Mata Kuliah	Mata kuliah ini mengajarkan paradigma pemrograman Fungsional dan melatih ketrampilan pemrograman secara fungsional. Dengan pemahaman ini dan wawasan paradigma pemrograman yang lebih luas, seseorang dapat memberikan penyelesaian pemrograman yang lebih efektif dan efisien disesuaikan dengan karakter permasalahannya. Ruang lingkup bahan kajian meliput konsep-konsep dasar pemrograman fungsional yang memberikan dasar kompetensi untuk menyelesaikan permasalahan nyata dan juga beberapa konsep-konsep lanjutan bagi peserta yang berminat memperdalam *software technology* seperti membuat bahasa pemrograman baru dan penelitian dibidang bahasa pemrograman lebih khusus teori Pemrograman Fungsional. Mata kuliah ini disajikan dengan metode pembelajaran jarak jauh, baik dilakukan secara formal terjadwal atau secara mandiri informal. Materi akan disajikan melalui berbagai media video, teks dan forum diskusi serta alat bantu lain. Praktikum dilakukan menggunakan fasilitas online baik sinkronus atau asinkronus.				
Tautan Kelas Daring	scele.cs.ui.ac.id atau funpro.cs.ui.ac.id (direncanakan) dapat juga diakses melalui emas.ui.ac.id				

CPL-PRODI yang dibebankan pada MK	
CPL-1	Mampu mengembangkan perangkat lunak dengan bahasa fungsional
CPL-2	Familiar dengan konsep-konsep pemrograman dalam paradigma rekursif dan fungsional.
CPL-3	Dapat menerapkan type- checking untuk membuat program yang lebih jelas dan dapat diverifikasi.
Capaian Pembelajaran Mata Kuliah (CPMK)	
CPMK-1	Memahami dan dapat menerapkan konsep paradigma pemrograman fungsional, dalam hal penyusunan algoritma dan struktur data.
CPMK-2	Memahami dan dapat memanfaatkan kaidah strongly-type untuk pengembangan aplikasi yang lebih efektif dan berkualitas dalam arti dapat menghindari dan mendeteksi kesalahan saat run-time.
CPMK-3	Mampu mengembangkan perangkat lunak menggunakan paradigma dan bahasa pemrograman fungsional sesuai kaidah.
CPMK-4	Memahami dan memiliki perspektif paradigma pemrograman dan mampu menerapkannya pada permasalahan pengembangan perangkat lunak secara tepat guna.

Sub-CPMK	
Sub- CPMK 1.1	Memahami paradigma deklaratif pada pemrograman fungsional
Sub- CPMK 1.2	Memahami konsep variable, ekspresi dan definisi fungsi
Sub- CPMK 1.3	Memahami konsep evaluasi dan dapat menerapkan evaluasi pada ekspresi secara teoritis dan tertulis
Sub- CPMK 1.4	Dapat menerapkan konsep variable, ekspresi, fungsi dan evaluasi dalam bahasa pemrograman fungsional seperti Haskell
Sub- CPMK 1.5	Memahami dan dapat menerapkan konsep *Higher Order Function*
Sub- CPMK 1.6	Memahami dan dapat menerapkan konsep *Partial Evaluation*
Sub- CPMK 1.7	Memahami dan dapat menerapkan konsep *Lazy Evaluation*
Sub- CPMK 2.1	Mengenal Lambda Calculus sebagai salah satu model komputasi
Sub- CPMK 2.2	Mengenal sejarah dan kategori type-system dalam bahasa pemrograman
Sub- CPMK 2.3	Memahami peranan type system dalam pemrograman
Sub- CPMK 2.4	Memahami definisi karakteristik dari *Typed Lambda Calculus*
Sub- CPMK 2.5	Mampu menuliskan ekspresi lambda calculus secara simbolik matematis dan mengevaluasinya secara teoritis dan tertulis manual
Sub- CPMK 2.6	Mampu menuliskan ekspreksi lambda calculus dalam bahasa pemrograman fungsional seperti Haskell
Sub- CPMK 2.7	Memahami salah satu algoritma type inferencing dengan mensimulasikan langkah kerja sebuah algoritma bila diberikan sebuah ekspresi dan type nya
Sub- CPMK 2.8	Mampu menganalisa ketidaksesuaian type dengan ekspresi
Sub- CPMK 3.1	Mampu menerapkan konsep-konsep pemrograman fungsional pada kasus-kasus klasik pemrograman seperti, search, sorting
Sub- CPMK 3.2	Memahami konsep Algebraic Data Type

Sub- CPMK 3.3	Dapat menerapkan Algebraic Data Type dalam sebuah permasalahan
Sub- CPMK 3.4	Memahami konsep modul
Sub- CPMK 3.5	Mampu memahami library lanjutan yang spesifik memanfaatkan kaida pemrograman fungsional seperti parser kombinator dan dapat menggunakan nya secara tepat sesuai kebutuhan
Sub- CPMK 3.6	Mengenal konsep lanjutan pemrograman fungsional sepertin *functor* dan *monad*
Sub- CPMK 4.1	Mengenal berbagai paradigma pemrograman dan keterhubungannya dalam sebuah produk perangkat lunak
Sub- CPMK 4.2	Memahami bagaimana paradigma pemrograman tertentu memiliki kelebihan dan kekurangan dibanding paradigma lain
Sub- CPMK 4.3	Mampu menerapkan pengetahuan paradigma pemrograman dalam memilih teknologi maupun bahasa pemrograman yang sesuai permasalahan
Sub- CPMK 4.4	Mampu menggabungkan penyelesaian yang diselesaikan dengan paradigma pemrograman yang berbeda untuk menyelesaikan permasalahan pengembangan perangkat lunak
Sub- CPMK 4.5	Mampu menerapkan best-practice terkini terkait bahasa pemrograman fungsional
Sub- CPMK 4.6	Memahami sejarah paradigma pemrograman dan keterkaitannya dengan perkembangan trend bahasa pemrograman terkini

Bahan Kajian: Materi pembelajaran	Course Overview; Introduction to programming paradigms and functional programming Pengenalan Haskel Ekspression dan Evaluation Lambda Calculus, Type System Partial Evaluation, Currying Higher Order Function and Lazy Evaluation Functional Component, React Functional Component: Paradigma Functional dalam React Hook Erlang Fault Tolerance system Erlang OTP: Process and Concurrency Advanced Topic: Implementation of Type System
Daftar Pustaka	**Wajib:** (1) Haskell: The Craft of Functional Programming, Simon Thompson, Addison-Wesley, 1999 (2) Learning React: Functional Web Development with React and Redux, Alex Banks, Eve Porcello, O'Reilly Media, 2018 (3) A Tutorial Introduction of Lambda Calculus, Lecture Notes by Ra´ul Rojas, FU Berlin. **Tambahan:** (4) Haskell School of Expression, Paul Hudak, Cambridge University Press, 2000 (5) Functional Programming in JavaScript, Dan Mantyla, Packt Publishing, March 2015 (6) Learn You Some Erlang for Great Good! A Beginner's Guide by Fred Hebert, No Starch Press, January 2013 Available online (free) at: http://learnyousomeerlang.com/content (7) Quick Introduction to Type Systems, Lecture Notes by David MacQueen, Univ. Princeton

RENCANA PEMBELAJARAN

(Makna Kode Indikator Khusus bisa dilihat di Lampiran)

Pekan Ke-	Sub CPMK/Sub CLO (Kemampuan pada setiap Akhir Tahap Pembelajaran)	Bahan Kajian (Materi Pembelajaran) [Rujukan]	Metode Pembelajaran	Moda Pembelajaran	Pengalaman Belajar (*O-L-U)		Indikator Pencapaian sub-CPMK	Bobot Penerapan sub CPMK pada MK
					Daring (Online)	Luring (Offline)	Indikator Umum; Khusus	Bobot Penerapan sub CPMK pada MK
1	**1.1 - 1.3**	Course Overview; Introduction to programming paradigms and functional programming. Rujukan [1 ,2, 5] dan sumber lain terkini	**120 menit Video**	**Asinkronus: Video Kuliah Tutorial dan Latihan Sinkronus: Diskusi, Tanya jawab**	**Orientasi:** Peserta diberikan arahan dan motivasi pemrograman fungsional dan arahan serta contoh awal	**Orientasi:** Peserta membaca buku.	**Umum: Peserta dapat merefleksikan pemahaman materi. Khusus: 1.1.1 - 1.3.4**	**5%**
			240 menit praktikum		**Latihan: -**	**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri contoh yang diberikan, berlatih mengerjakan latihan		

			120 menit aktif mandiri (membaca, tanya jawab, menulis, diskusi dll)		**Umpan Balik:** Peserta memaparkan pengalaman programming sebelumnya, memberikan perbandingan, refleksi. Termasuk juga pada awal kuliah peserta memaparkan kesulitan yang dihadapi sebelumnya. peserta memberikan refleksi dan umpan balik dari praktikum untuk mengukur pemahaman serta memastikan siap untuk materi selanjutnya.			
			60 menit soal latihan (kuis)					
2	**1.4**	Pengenalan Haskel [1]	**120 menit Video**	**Asinkronus: Video Kuliah Tutorial dan Praktikum Sinkronus: Diskusi, Tanya jawab**	**Orientasi:** Peserta diberikan arahan dan motivasi pemrograman fungsional dan arahan serta contoh awal	**Orientasi:** Peserta membaca buku.	**Umum: Peserta dapat merefleksikan pemahaman materi, menjalankan program**	**10%**

			240 menit praktikum		**Latihan:** Praktikum 1 dilakukan secara online dan dimonitor, dapat dilakukan asinkronus.	**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri contoh yang diberikan, berlatih mengerjakan latihan	**haskell sederhana dan menjelaskannya.** **Khusus: 1.4.1 - 1.4.6**	
			120 menit aktif mandiri (membaca, tanya jawab, menulis, diskusi dll)		**Umpan Balik:** Peserta mencoba dan melaporkan hasil percobaannya, dalam arti sharing pengalaman dan refleksi atau tanya jawab diskusi.	**Umpan Balik: -**		
			60 menit soal latihan (kuis)					
3	**1.4**	Ekspression dan Evaluation [1,2]	**120 menit Video**	**Asinkronus: Video Kuliah Tutorial dan Praktikum Sinkronus: Diskusi, Tanya jawab**	**Orientasi:** Peserta diberikan arahan dan motivasi pemrograman fungsional dan arahan serta contoh awal	**Orientasi:** Peserta membaca buku.	**Umum: Peserta dapat merefleksikan pemahaman materi. memperlihatkan dan menjelaskan cara mengevaluasi ekspresi dalam functional programming**	**10%**
			240 menit praktikum		**Latihan:** Praktikum 2 dilakukan secara online dan dimonitor, dapat dilakukan asinkronus.	**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri contoh yang diberikan, berlatih mengerjakan latihan		

			120 menit aktif mandiri (membaca, tanya jawab, menulis, diskusi dll)		**Umpan Balik:** Peserta mencoba dan melaporkan hasil percobaannya, dalam arti sharing pengalaman dan refleksi atau tanya jawab diskusi.	**Umpan Balik: -**	**Khusus: 1.4.1 - 1.4.6**	
			60 menit soal latihan (kuis)					
4	**2.1-2.6**	Lambda Calculus, [3] Type System [3,7]	**120 menit Video**	**Asinkronus: Video Kuliah Tutorial dan Praktikum Sinkronus: Diskusi, Tanya jawab**	**Orientasi:** Peserta diberikan arahan dan motivasi pemrograman fungsional dan arahan serta contoh awal	**Orientasi:** Peserta membaca buku.	**Umum: Peserta dapat merefleksikan pemahaman materi. Peserta dapat membuat ekspresi typed lambda calculus, menjelaskan dan mengevaluasinya** **Khusus: 2.1.1-2.6.4**	**10%**
			240 menit praktikum		**Latihan:** Praktikum 3 dilakukan secara online dan dimonitor, dapat dilakukan asinkronus.	**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri contoh yang diberikan, berlatih mengerjakan latihan		
			120 menit aktif mandiri (membaca, tanya jawab, menulis, diskusi dll)		**Umpan Balik:** Peserta mencoba dan melaporkan hasil percobaannya, dalam arti sharing pengalaman dan refleksi atau tanya jawab diskusi.	**Umpan Balik: -**		
			60 menit soal					

			latihan (kuis)					
5	**1.6**	Partial Evaluation, Currying [1,3,5]	**120 menit Video**	**Asinkronus: Video Kuliah Tutorial dan Praktikum Sinkronus: Diskusi, Tanya jawab**	**Orientasi:** Peserta diberikan arahan dan motivasi pemrograman fungsional dan arahan serta contoh awal	**Orientasi:** Peserta membaca buku.	**Umum: Peserta dapat merefleksikan pemahaman materi. Peserta dapat memahami dan membuat program yang menerapkan Partial Evaluation and Currying. Dapat mengubah uncurrying menjadi currying. Khusus: 1.6.1-1.6.4**	**5%**
			240 menit praktikum		**Latihan:** Praktikum 4 dilakukan secara online dan dimonitor, dapat dilakukan asinkronus.	**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri contoh yang diberikan, berlatih mengerjakan latihan		
			120 menit aktif mandiri (membaca, tanya jawab, menulis, diskusi dll)		**Umpan Balik:** Peserta mencoba dan melaporkan hasil percobaannya, dalam arti sharing pengalaman dan refleksi atau tanya jawab diskusi.	**Umpan Balik: -**		
			60 menit soal latihan (kuis)					
6	**1.5 dan 1.7**	Higher Order Function and Lazy Evaluation [1,3,5]	**120 menit Video**	**Asinkronus: Video Kuliah Tutorial dan Praktikum Sinkronus:**	**Orientasi:** Peserta diberikan arahan dan motivasi pemrograman fungsional dan	**Orientasi:** Peserta membaca buku.	**Umum: Peserta dapat merefleksikan pemahaman materi. Peserta**	**10%**

				Diskusi, Tanya jawab	arahan serta contoh awal		**dapat memahami program yang menerapkan Higher order function dan dapat merefactor program bisa agar menerapkan higherorder function. Peserta memahami penerapan HOF bisa membuat program menjadi lebih modular. Peserta juga bisa memperlihatkan lazy evaluation. Khusus: 1.5.1 - 1.5.4 , 1.7.1 - 1.7.4**	
			240 menit praktikum		**Latihan:** Praktikum 5 dilakukan secara online dan dimonitor, dapat dilakukan asinkronus.	**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri contoh yang diberikan, berlatih mengerjakan latihan		
			120 menit aktif mandiri (membaca, tanya jawab, menulis, diskusi dll)		**Umpan Balik:** Peserta mencoba dan melaporkan hasil percobaannya, dalam arti sharing pengalaman dan refleksi atau tanya jawab diskusi.	**Umpan Balik: -**		
			60 menit soal latihan (kuis)					
7	**1.1- 1.3 dan 3.1-4.2**	Initiation group project [2], Quiz dan Review	**120 menit Video**	**Asinkronus: Video Kuliah Tutorial dan Praktikum Sinkronus: Diskusi, Tanya jawab**	**Orientasi:** Peserta diberikan arahan terkait materi perkembangan terkini dan penerapan functional programming	**Orientasi:** Peserta membaca buku.	**Umum: Peserta dapat memperlihatkan pemahaman materi selama ini dan dapat mengusulkan**	**5%**

					dalam permasalahan nyata		**topik permasalahan yagn bisa diselesaikan dengan pendekatan functional programming. Khusus: 1.1.1 - 1.3.4 - 4.1.1 - 4.2.5**	
			240 menit praktikum		**Latihan:** Inisiasi pengerjaan project dilakukan secara online dan dimonitor, dapat dilakukan asinkronus.	**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri contoh yang diberikan, berlatih mengerjakan latihan		
			120 menit aktif mandiri (membaca, tanya jawab, menulis, diskusi dll)		**Umpan Balik:** Peserta mencoba dan melaporkan hasil percobaannya, dalam arti sharing pengalaman dan refleksi atau tanya jawab diskusi.	**Umpan Balik: -**		
			60 menit soal latihan (kuis)					
8		Mid Term Exam		**Sinkronus: Individual review (Ujian lisan) Asinkronus: Ujian tertulis online, dan tugas**				
9	**3.1 - 4.4**	Functional Component, React [2,5]	**120 menit Video**	**Asinkronus: Video Kuliah Tutorial dan Praktikum Sinkronus:**	**Orientasi:** Peserta diberikan arahan dan motivasi pemrograman fungsional dan	**Orientasi:** Peserta membaca buku.	**Umum: Peserta dapat merefleksikan pemahaman materi. Peserta**	**10%**

				Diskusi, Tanya jawab	arahan serta contoh awal		**memhami functinoal component dan penerapan functional programming pada javascript Khusus: 4.4.1 - 4.5.3**	
			240 menit praktikum		**Latihan:** Pengerjaan project dilakukan secara online, berkelompok dan dimonitor, dapat dilakukan asinkronus.	**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri contoh yang diberikan, berlatih mengerjakan latihan		
			120 menit aktif mandiri (membaca, tanya jawab, menulis, diskusi dll)		**Umpan Balik:** Peserta mencoba dan melaporkan hasil percobaannya, dalam arti sharing pengalaman dan refleksi atau tanya jawab diskusi.	**Umpan Balik: -**		
			60 menit soal latihan (kuis)					
10	**3.1 - 4.4**	Functional Component: Paradigma Functional dalam React Hook [2,5] dan sumber terkini lainnya.	**120 menit Video**	**Asinkronus: Video Kuliah Tutorial dan Praktikum Sinkronus: Diskusi, Tanya jawab**	**Orientasi:** Peserta diberikan arahan terkait materi, dan arahan ntuk latihan penerapan pemrograman fungsional dan arahan serta contoh awal	**Orientasi:** Peserta membaca buku.	**Umum: Peserta dapat merefleksikan pemahaman materi. Peserta memhami functinoal component dan penerapan functional programming pada javascript**	**5%**
			240 menit praktikum		**Latihan:** Pengerjaan project dilakukan secara online,	**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri		

					berkelompok dan dimonitor, dapat dilakukan asinkronus.	contoh yang diberikan, berlatih mengerjakan latihan	**Khusus: 4.4.1 - 4.5.3**	
			120 menit aktif mandiri (membaca, tanya jawab, menulis, diskusi dll)		**Umpan Balik:** Peserta mencoba dan melaporkan hasil percobaannya, dalam arti sharing pengalaman dan refleksi atau tanya jawab diskusi.	**Umpan Balik: -**		
			60 menit soal latihan (kuis)					
11	**3.1 - 4.4**	Erlang Fault Tolerance system [1,6]	**120 menit Video**	**Asinkronus: Video Kuliah Tutorial dan Praktikum Sinkronus: Diskusi, Tanya jawab**	**Orientasi:** Peserta diberikan arahan terkait materi, dan arahan ntuk latihan penerapan pemrograman fungsional dan arahan serta contoh awal	**Orientasi:** Peserta membaca buku.	**Umum: Peserta dapat merefleksikan pemahaman materi. Peserta memhami functinoal component dan penerapan functional programming pada Erlang Khusus: 4.4.1 - 4.5.3**	**5%**
			240 menit praktikum		**Latihan:** Pengerjaan project dilakukan secara online, berkelompok dan dimonitor, dapat dilakukan asinkronus.	**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri contoh yang diberikan, berlatih mengerjakan latihan		

			120 menit aktif mandiri (membaca, tanya jawab, menulis, diskusi dll)		**Umpan Balik:** Peserta mencoba dan melaporkan hasil percobaannya, dalam arti sharing pengalaman dan refleksi atau tanya jawab diskusi.	**Umpan Balik: -**		
			60 menit soal latihan (kuis)					
12	**3.1 - 4.4**	Erlan OTP: Process and Concurrency 1[,6]	**120 menit Video**	**Asinkronus: Video Kuliah Tutorial dan Praktikum Sinkronus: Diskusi, Tanya jawab**	**Orientasi:** Peserta diberikan arahan terkait materi, dan arahan ntuk latihan penerapan pemrograman fungsional dan arahan serta contoh awal	**Orientasi:** Peserta membaca buku.	**Umum: Peserta dapat merefleksikan pemahaman materi. Peserta memhami functinoal component dan penerapan functional programming pada Erlang Khusus: 4.4.1 - 4.5.3**	**5%**
			240 menit praktikum		**Latihan:** Pengerjaan project dilakukan secara online, berkelompok dan dimonitor, dapat dilakukan asinkronus.	**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri contoh yang diberikan, berlatih mengerjakan latihan		
			120 menit aktif mandiri (membaca, tanya jawab,		**Umpan Balik:** Peserta mencoba dan melaporkan hasil percobaannya, dalam arti sharing pengalaman dan	**Umpan Balik: -**		

			menulis, diskusi dll)		refleksi atau tanya jawab diskusi.			
			60 menit soal latihan (kuis)					
13	**2.7 2.8**	Advanced Topic: Implementation of Type System [3,7]	**120 menit Video**	**Asinkronus: Video Kuliah Tutorial dan Praktikum Sinkronus: Diskusi, Tanya jawab**	**Orientasi:** Peserta diberikan arahan terkait materi, dan arahan ntuk latihan penerapan pemrograman fungsional dan arahan serta contoh awal	**Orientasi:** Peserta membaca buku.	**Umum: Peserta dapat merefleksikan pemahaman materi. Peserta mengenal topik lanjutan dari Functional Programming yang masih terus berkembang baik secara pragmatis untuk industri maupun dalam dunia riset. Khusus: 2.7.1 - 2.8.3**	**10%**
			240 menit praktikum		**Latihan:** Pengerjaan project dilakukan secara online, berkelompok dan dimonitor, dapat dilakukan asinkronus.	**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri contoh yang diberikan, berlatih mengerjakan latihan		
			120 menit aktif mandiri (membaca, tanya jawab, menulis, diskusi dll)		**Umpan Balik:** Peserta mencoba dan melaporkan hasil percobaannya, dalam arti sharing pengalaman dan refleksi atau tanya jawab diskusi.	**Umpan Balik: -**		

			60 menit soal latihan (kuis)					
14	**1.1 - 4.4**	Project Presentation	**120 menit Video**	**Asinkronus: Diskusi, tanya jawab Sinkronus: Presentation, Diskusi, Tanya jawab**	**Orientasi:** Peserta diberikan arahan terkait materi perkembangan terkini dan penerapan functional programming dalam permasalahan nyata	**Orientasi:** Mempersiapkan presentasi dan demo tugas project	**Umum: Peserta dapat merefleksikan pemahaman materi. Peserta dapet mempresentasikan hasil kerja dan menjelaskan penerapan Functional Programming untuk penyelesaian permasalahan Khusus: seluruh indikator list**	**5%**
			60 menit Presentation		**Latihan:** Inisiasi pengerjaan project dilakukan secara online dan dimonitor, dapat dilakukan asinkronus.	**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri contoh yang diberikan, berlatih mengerjakan latihan		
			120 menit aktif mandiri (membaca, tanya jawab, menulis, diskusi dll)		**Umpan Balik:** Peserta mencoba dan melaporkan hasil percobaannya, dalam arti sharing pengalaman dan refleksi atau tanya jawab diskusi.	**Umpan Balik: -**		
			60 menit soal latihan (kuis)					

15	**1.1 - 4.4**	Project Presentation	**120 menit Video**	**Asinkronus: Diskusi, tanya jawab Sinkronus: Presentation, Diskusi, Tanya jawab**	**Orientasi:** Peserta diberikan arahan terkait materi perkembangan terkini dan penerapan functional programming dalam permasalahan nyata	**Orientasi:** Mempersiapkan presentasi dan demo tugas project	**Umum: Peserta dapat merefleksikan pemahaman materi. Peserta dapet mempresentasikan hasil kerja dan menjelaskan penerapan Functional Programming untuk penyelesaian permasalahan Khusus: seluruh indikator list**	**5%**
			60 menit Presentation		**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri contoh yang diberikan, berlatih mengerjakan latihan	**Latihan:** Peserta mencoba sendiri, mengulang kembali secara mandiri contoh yang diberikan, berlatih mengerjakan latihan		
			120 menit aktif mandiri (membaca, tanya jawab, menulis, diskusi dll)		**Umpan Balik:** Peserta mencoba dan melaporkan hasil percobaannya, dalam arti sharing pengalaman dan refleksi atau tanya jawab diskusi.	**Umpan Balik: -**		
			60 menit soal latihan (kuis)					
16		Final Term		**Sinkronus: Individual review (Ujian lisan) Asinkronus:**				

				Ujian tertulis online, dan tugas				

*) Mg: Minggu
**) Sinkronus: interaksi pembelajaran antara dosen dan mahasiswa dilakukan pada waktu yang bersamaan, menggunakan teknologi *video conference* atau *chatting*.
Asinkronus: interaksi pembelajaran dilakukan secara fleksibel dan tidak harus dalam waktu yang sama, misalkan menggunakan forum diskusi atau belajar mandiri/penugasan mahasiswa.

RANCANGAN TUGAS DAN LATIHAN

Pekan Ke	Nama Tugas	Sub CPMK	Penugasan	Ruang Lingkup	Cara Pengerjaan	Batas Waktu	Luaran Tugas yang Dihasilkan
1	Praktikum 1	1.1 – 1.4	Soal Programing	Terbatas sesuai acuan literatur	Individu, dibantu tutor	1 pekan	Source code
2	Praktikum 2	2.1 – 2.4	Soal Programing	Terbatas sesuai acuan literatur	Individu, dibantu tutor	1 pekan	Source code
3	Praktikum 3	3.1 – 3.4	Soal Programing	Terbatas sesuai acuan literatur	Individu, dibantu tutor	1 pekan	Source code
4	Praktikum 4	1.7, 3.1-3.4	Soal Programing	Terbatas sesuai acuan literatur	Individu, dibantu tutor	1 pekan	Source code
5	Praktikum 5	1.5,1.6,1.7,	Soal Programing	Terbatas sesuai acuan literatur	Individu, dibantu tutor	2 pekan	Source code

6-9	Project Kelompok	1.1 – 1.7, 3.1 - 3.5, 4.1-4.3	Pembuatan project product software	Bebas, peserta diharapkan mengembangkan kreativitas dan inovasinya.	Kerja kelompok, dengan milestone bertahap	4 pekan	Source code, presentation dan dokumentasi

KRITERIA PENILAIAN (EVALUASI HASIL PEMBELAJARAN)
Pada bagian ini dituliskan

Bentuk Evaluasi	**Sub-CPMK**	**Instrumen/ Jenis Asesmen**	**Frekuensi**	**Bobot Evaluasi (%)**
Kuis rutin tertulis	Sesuai pekan	Tertulis sesuai kunci jawaban	11	10%
Praktikum	Sesuai modul praktikum	Review source code dan demo	5	25%
Ujian	Keseluruhan	Tertulis dan individual review, ujian lisan	2	50%
Project	Keseluruhan	Observasi, review source code	1	15%
Total				**100%**

Pedoman Kriteria Penilaian

Konversi nilai akhir mahasiswa berdasarkan ketentuan yang berlaku di Universitas Indonesia. Konversi nilai tersebut adalah:

Nilai Angka	**Nilai Huruf**	**Bobot**
85—100	A	4,00
80—<85	A-	3,70
75—<80	B+	3,30

70—<75	B	3,00
65—<70	B-	2,70
60—<65	C+	2,30
55—<60	C	2,00
40—<55	D	1,00
<40	E	0,00

Rubrik Penilaian:
Rubrik ini digunakan sebagai pedoman untuk menilai atau memberi tingkatan dari hasil kinerja mahasiswa. Rubrik biasanya terdiri dari kriteria penilaian yang mencakup dimensi/aspek yang dinilai berdasarkan indikator capaian pembelajaran. Rubrik penilaian ini berguna untuk memperjelas dasar dan aspek penilaian sehingga mahasiswa dan dosen bisa berpedoman pada hal yang sama mengenai tuntutan kinerja yang diharapkan. Dosen dapat memilih jenis rubrik yang sesuai dengan asesmen yang diberikan. Rubrik ini dapat digunakan melalui fitur *Assignment* dalam EMAS UI, dengan mengaktifkan fitur *Rubric* pada bagian *Grading Method*.

Rubrik dipisahkan untuk jenis evaluasi yang relatif ada faktor subjectif. Untuk evaluasi yang objectif, ditentukan langsung dari penilaian secara otomatis sesuai jawaban benar salah peserta.

Rubrik Project

	A - Excelence	B - Expected	C – Minimal Competence	D – Unaccepted
Presentation	Presentasi sangat menyakinkan, ilustrasi yang bagus, media yang menarik, jelas dan menarik serta sesuai.	Presentasi dipaparkan dengan lancar dan jelas. Artikulasi sangat baik slide dibuat menarik	Presentasi seadanya hanya membaca slide dan memperlihatkan source code serta demo	Presentasi kurang disiapkan, slide tidak tersedia baik dan demo tidak berjalan baik.
Documentation	Dokumentasi dibuat sangat menarik, mudah dibaca, teratur sangat memudahkan orang lain	Dokumentasi lengkap, pembaca bisa meredo pekerjaan nya	Dokumentasi yang paling penting tersedia. Pembaca masih mungkin untuk dapat mengulang	Informasi yang paling penting, tidak tersedia.

	untuk melanjutkan dan meredo pekerjaannya.		pekerjaan tapi butuh pengetahuan ekstra	
Source Code	Menerapkan lebih dari 60% dengan kualitas kerja setara dengan anggota team dalam arti tim mengeluarkan potensi maksimal mereka.	Menerapkan minimal 60% dari materi yang telah dipelajari dengan tepat guna. Kualitas kerja setara dengan jumlah tim.	Menerapkan kurang dari 60% namun dapat menunjukan beberapa penerapan yang tepat. Program tepat bisa berjalan baik.	Program tidak bisa berjalan baik.

Rubrik untuk praktikum akan dirinci dimasing-masing modul praktikum disesuaikan dengan tugas terkait.

Rubrik untuk individual Review

	A - Excelence	B - Expected	C – Minimal Competence	D – Unaccepted
Demo Programming	Peserta dapat memperlihatkan penerapan skill functional programming dengan baik dan bisa coding lebih cepat dari rata-rata. Sudah sangat trampil, sehingga tidak perlu lihat literatur lagi	Peserta bisa menerapkan semua materi dengan baik, bila diberikan waktu yang cukup dan akses ke literatur	Peserta bisa implementasi lebih dari 50 % materi.	Peserta tidak bisa menerapkan materi sebanyak 50% dalam bentuk programming
Tanya Jawab	Peserta menjawab lebih lancar, dan tepat	Peserta dapat menjawab namun sekitar 30% pertanyaan perlu sedikit arahan atau koreksi	Peserta menjawab baik kurang dari 70% pertanyaan /pemahaman	Peserta menjawab baik kurang dari 50% pemahaman

Pengembangan Diri	Peserat mengembangkan potensi dirinya dengan maksimal, mempelajari hal lain yang tidak dibahas dikelas.	Peserta mempelajari materi ekstra sebatas yang diarahkan saja.	Peserta hanya memperlajari materi seperti yang diajarkan saja.	Peserta memahami materi atau mengikuti kegiatan kurang dari 60% (misalnya ada 2 praktikum yang tidak dikerjakan)

Lampiran CPMK, Sub CPMK dan Indikator

Penulisan CPMK, Sub-CPMK dan Indikatornya mengikuti penomoran level yang dipisahkan dengan titik. Contoh kode 3.2.4, menyatakan CPMK nomor, dengan sub CMPK nomor 2 dan indikator ke-4 dari sub-CPMK nomor 2 tersebut.

1. Memahami dan dapat menerapkan konsep paradigma pemrograman fungsional, dalam hal penyusunan algoritma dan struktur data.

1.1. Memahami paradigma deklaratif pada pemrograman fungsional

1.1.1. Peserta dapat menguraikan minimal 3 paradigma pemrograman

1.1.2. Peserta memahami ilustrasi sederhana yang memperlihatkan perbedaan masing-masing paradigman

1.1.3. Peserta dapat menyebutkan contoh bahasa pemrograman pada masing-masing paradigma

1.1.4. Peserta dapat menyebutkan kelebihan paradigma deklaratif pada pemrograman fungsional

1.2. Memahami konsep variable, ekspresi dan definisi fungsi

1.2.1. Peserta dapat membedakan variable dalam functional dan dalam imperatif

1.2.2. Peserta dapat menuliskan eksepresi arithmatika dalam bahasa pemrograman functional seperti haskell

1.2.3. Peserta memahami dan dapat memperlihatkan bahwa variable dalam funtional programming adalah immutable atau tidak bisa diubah, dengan kata lain adalah alias dari sebuah data.

1.2.4. Peserta dapat menyusun definisi fungsi sederhana yang merupakan representasi dari ekspressi yang dibungkus atau diwakili oleh fungsi tersebut

1.3. Memahami konsep evaluasi dan dapat menerapkan evaluasi pada ekspresi secara teoritis dan tertulis.

1.3.1. Peserta mengetahui bahwa pola evaluasi pada functional programming adalah seperti pembuktian persamaan matematika

1.3.2. Peserta dapat mencontohkan menguraikan definisi dalam proses evaluasi

1.3.3. Bila diberikan sebuah ekspresi, peserta dapat menguraikan apakah ekspresi tersebut dievaluasi secara benar atau tidak.

1.3.4. Bila diberikan definisi fungsi, peserta dapat menguraikan evaluasi dari fungsi-fungsi tersebut dan dapat menentukan apakah properti yang diharapkan dari fungsi tersebut adalah benar. Misalnya fungsi untuk menentukan akan triple segitiga membentuk phytagoras atau bukan.

1.4. Dapat menerapkan konsep variable, ekspresi, fungsi dan evaluasi dalam bahasa pemrograman fungsional seperti Haskell

1.4.1. Peserta dapat membuat ekspresi pada bahasa pemrograman dan melihat evaluasi ekspresi tersebut oleh interpreter

1.4.2. Peserta mengetahui dan dapat menggunakan beberapa tipe dan struktur data dasar dalam Haskell seperti list comprehension.

1.4.3. Peserta dapat membuat fungsi dan mengeksekusinya pada interpreter

1.4.4. Bila diberikan fungsi yang keliru, dan interpreter yang memberikan pesan error terkait ekspresi, peserta dapat menganalisa pesan error tersebut dan melakukan perbaikan

1.4.5. Bila diberikan sebuah definisi fungsi, Peserta bisa menjelaskan dan menguraikan elemen-elemen dari fungsi tersebut seperti apa yang dilakukan, apa yang jadi parameter dan hasil dari fungsi tersebut.

1.4.6. Peserta bisa membuat program sederhana (tanpa rekursif), kurang dari 10 baris yang melakukan perhitungan sederhana seperti menentukan jenis segitiga yang bisa dibentuk dari tiga sisi yang merupakan paramater dari fungsi.

1.5. Memahami dan dapat menerapkan konsep *Higher Order Function*.

1.5.1. Bila diberikan program yang menerapkan HOF, peserta dapat menjelaskan cara kerja program tersebut

1.5.2. Peserta dapat membuat fungsi yang menerima fungsi sebagai parameter

1.5.3. Peserta dapat membuat fungsi yang menghasilkan fungsi sebagai outputnya

1.5.4. Peserta dapat mencontohkan penerapan dan evaluasi dari fungsi yang menerapakan HOF.

1.5.5. Peserta dapat menjelaskan manfaat penerapan HOF dibanding bahasa pemrograman lain yang tidak memiliki HOF.

1.6. Memahami dan dapat menerapkan konsep *Partial Evaluation*.

1.6.1. Peserta dapat menjelaskan evaluasi program yang menerapkan *Partial Evaluation*.

1.6.2. Peserta dapat membuat program yang didalamnya ada beberapa penerapan fungsi yang partial, tidak langsung dievaluasi menjadi value, tapi masih berupa fungsi yang bermanfaat untuk proses selanjutnya.

1.6.3. Peserta dapat memahami bahwa partial evaluation adalah salah satu bentuk upaya parameterisasi dan modularisasi program yang lebih tinggi dari sekedar mengganti value parameter. Peserta dapat menunjukkan dalam bentuk perbandingan dengan program lain yang tidak menerapkan partial evaluation.

1.6.4. Bila diberikan program dengan partial evaluation yang keliru dan pesan error dari interpreter, peserta dapat menganalisa dan memperbaikinya.

1.7. Memahami dan dapat menerapkan konsep *Lazy Evaluation*.

1.7.1. Diberikan program yang menerapkan *Lazy evaluation*, peserta dapat menjelaskan cara kerja dan evaluasinya. Misalnya seperti algoritam *sieve of erasthothenes* dalam menggenerate bilangan prima.

1.7.2. Peserta memahami struktur data stream dan list of comprehension dalam bahasa pemrograman seperti haskell yang menerapkan lazy evaluationa.

1.7.3. Peserta dapat membuat program menggunakan list comprehension yang menerapkan lazy evaluation.

1.7.4. Peserta memahami bagaimana secara konseptual, lazy evaluation bekerja di bahasa pemrograman seperti Haskell. Bila diberikan sebuah ekspressi yang di evaluasi secara lazy, peserta dapat mengilustrasikan dengan bahasa nya sendiri.

2. Memahami dan dapat memanfaatkan kaidah strongly-type untuk pengembangan aplikasi yang lebih efektif dan berkualitas dalam arti dapat menghindari dan mendeteksi kesalahan saat run-time.

2.1. Mengenal *Lambda Calculus* sebagai salah satu model komputasi

2.1.1. Peserta mengetahui apa yang dimaksud dengan model komputasi secara sederhana (bukan mendalam) dan dapat menguraikan nya dengan bahasa sendiri atau ilustrasi lain.

2.1.2. Peserta dapat menguraikan maksud *Lambda Calculus* sebagai model komputasi dengan bahasa nya sendiri

2.1.3. Peserta dapat membandingkan dengan model komputasi sederhana lain nya seperti *von-neumaan machine*.

2.2. Mengenal sejarah dan kategori type-system dalam bahasa pemrograman.

2.2.1. Peserta dapat menyebutkan kategori type-system dan karakteristiknya. Misalnya *Strongly Type,Weekly Type, Static Type, Dinamic Type*.

2.2.2. Peserta dapat menyebutkan beberapa bahasa pemrograman yang memiliki type system sesuai kategori tersebut.

2.2.3. Peserta dapat menjelaskan karakteristik masing-masing kategory type system tersebut. Peserta dapat memberikan contoh perbedaan dari masing-masing bahasa pemrograman terkait.

2.3. Memahami peranan type system dalam pemrograman

2.3.1. Peserta menyadari manfaat dari type system. Peserta dapat merangkum dari pengalaman profesional programmer yang bisa dicari dari literatur ataupun profesional blog.

2.3.2. Peserta dapat memberikan contoh kasus pemanfaatan type system. Contoh kasus bisa diperoleh dari praktikum yang dilakukan sendiri atau bisa didapat juga dari sumber lain atau pengalaman lain yang dipahami dan bisa diuraikan dengan kalimat sendiri.

2.4. Memahami definisi karakteristik dari *Typed Lambda Calculus*

2.4.1. Peserta mengenal dua karakteristik utama dari yaitu alpha-convention dan beta-reduction, (additional point untuk peserta yang berminat bisa lebih lanjut mempelajari eta-expansion)

2.4.2. Bila diberikan ekspresi dalam lambda calculus, peserta bisa memperlihatkan penerapan karakteristik tersebut.

2.4.3. Bila diberikan evaluasi yang keliru, peserta dapat menjelaskan kekeliruannya.

2.4.4. Peserta mengetahui isu terkait capturing free-variable.

2.4.5. Peserta dapat menambahkan informasi tentang tipe pada ekspressi lambda calculus.

2.5. Mampu menuliskan ekspresi lambda calculus secara simbolik matematis dan mengevaluasinya secara teoritis dan tertulis manual

2.5.1. Diberikan sebuah soal cerita atau ekspresi dalam bahasa manusia, peserta dapat menuliskan ekspresi tersebut dalam lambda calculus.

2.5.2. Peserta dapat menguraikan ekspresi lambda calculus dan mengevaluasi sesuai aturan rule yang telah dipelajari.

2.5.3. Peserta bisa memberikan contoh ekspresi dan evaluasi yang mengalami capturing free-variable.

2.6. Mampu menuliskan ekspreksi lambda calculus dalam bahasa pemrograman fungsional seperti Haskell.

2.6.1. Diberikan sebuah ekspresi dalam lambda calculus, peserta dapat menuliskan nya dalam bentuk bahasa pemrograman Haskell

2.6.2. Peserta dapat membandingkan hasil evaluasi manual dengan evaluasi oleh interpreter haskell.

2.6.3. Ketika ada kesalahan penulisan lambda calculus secara manual yang dipetakan kedalam bahasa pemrograman haskell, pesan error yang diberikan Haskell dapat dianalisa oleh peserta dan memberikan saran perbaikan.

2.6.4. Bila ada kesalahan terkait penetapan tipe, peserta dapat menganalisa pesan error dari haskell dan melakukan perbaikan.

2.7. Memahami salah satu algoritma type inferencing dengan mensimulasikan langkah kerja sebuah algoritma bila diberikan sebuah ekspresi dan type nya.

2.7.1. Peserta mengetahui algoritme type inferencing misalnya seperti algoritma yang di susun oleh Curry dan Feys serta di modifikasi oleh Hindley.

2.7.2. Peserta mengenal terminologi, tata-cara, notasi dalam type theory

2.7.3. Peserta mengetahui cara penulisan type judgement

2.7.4. Diberikan sebuah type judgement, peserta dapat menjelaskan dengan kalimat nya sendiri

2.8. Mampu menganalisa ketidaksesuaian type dengan ekspresi

2.8.1. Peserta dapat menentukan apakah sebuah ekspressi ill-typed atau tidak

2.8.2. Diberikan sebuah ekspresi, peserta dapat menguraikan type judgemen mengikuti pola salah satu algoritma.

2.8.3. Diberikan sebuah ekspresi dalam haskell, peserta dapat menguraikan type judgement nya.

2.8.4. Diberikan semua pesan error haskell terkait type-error, peserta dapat menjelaskan bagaiamana type-judgement dilakukan.

3. Mampu mengembangkan perangkat lunak menggunakan paradigma dan bahasa pemrograman fungsional sesuai kaidah.

3.1. Mampu menerapkan konsep-konsep pemrograman fungsional pada kasus-kasus klasik pemrograman seperti, search, sorting

3.1.1. Peserta dapat memahami cara kerja algoritma searching dan sorting serta beberapa algoritma dasar dalam functional programming

3.1.2. Peserta dapat memodifikasi algoritma sederhana terkait

3.1.3. Peserta dapat membuat program dalam bahasa fungsional yang tingkat kesulitan algoritmanya setara dengan searching sorting.

3.2. Memahami konsep Algebraic Data Type.

3.2.1. Peserta memahami terminologi Algebraic Data Type

3.2.2. Diberikan sebuah ADT, peserta dapat menjelas dengan kalimat nya sendiri

3.2.3. Diberikan sebuah kasus, peserta dapat membuat ADT yang berkesesuaian

3.3. Dapat menerapkan Algebraic Data Type dalam sebuah permasalahan

3.3.1. Peserta memahami peranan ADT dalam sebuah penyelesaian masalah.

3.3.2. Peserta dapat memetakan persama ADT dalam pemrograman fungsional dibanding paradigma lain.

3.3.3. Peserta dapat menyusun penyelesaian dengan merancang beberapa ADT

3.3.4. Peserta dapat membuat program yang memanfaatkan ADT

3.4. Memahami konsep modul.

3.4.1. Peserta mengenal terminologi Modul

3.4.2. Peserta dapat membuat program dalam modul

3.4.3. Peserta dapat menggunakan modul lain dalam programnya.

3.4.4. Peserta bisa menerapkan standard best practice dalam penggunaan modul termasuk terkait isu fully qualified name, atau restriction import.

3.5. Mampu memahami library lanjutan yang spesifik memanfaatkan kaida pemrograman fungsional seperti parser kombinator dan dapat menggunakan nya secara tepat sesuai kebutuhan.

3.5.1. Peserta memahami bahwa banyak library functional programming telah dikembangka dan tidak perlu dibuat ulang.

3.5.2. Peserta dapat menggunakan library yang telah tersedia.

3.5.3. Peserta dapat membuat library sendiri sesuai standard best practice yang ada.

3.6. Mengenal konsep lanjutan pemrograman fungsional sepertin *functor* dan *monad*

3.6.1. Peserta mengenal adanya konsep lanjutan yang bisa memberikan manfaat lebih seperti *functor* dan *monad*.

3.6.2. Peserta memahami terminology dan bisa memberikan contoh

3.6.3. Diberikan sebuah program yang menerapkan konsep tersebut, peserta dapat menjelaskan dengan kalimat nya sendiri.

3.6.4. Peserta memahami dan dapat menerapkan functor untuk fleksibilitas pemrograman yang lebih baik lain. Peserta dapat menunjukkan nya dengan membuat dua versi dan memperlihat manfaat pada versi yang menerapkan. (ekstra)

3.6.5. Peserta memahami dan dapat menerapkan monad untuk fleksibilitas pemrograman yang lebih baik lain. Peserta dapat menunjukkan nya dengan membuat dua versi dan memperlihat manfaat pada versi yang menerapkan. (ekstra)

4. Memahami dan memiliki perspektif paradigma pemrograman dan mampu menerapkannya pada permasalahan pengembangan perangkat lunak secara tepat guna.

4.1. Mengenal berbagai paradigma pemrograman dan keterhubungannya dalam sebuah produk perangkat lunak

4.1.1. Bila diberikan sebuah project, peserta dapat menunjukkan paradigma yang digunakan pada masing-masing bagian.

4.1.2. Peserta dapat memperlihatkan bagaimana keterhubungan masing-masing paradigma tersebut dalam pengembangan perangkat lunak.

4.2. Memahami bagaimana paradigma pemrograman tertentu memiliki kelebihan dan kekurangan dibanding paradigma lain.

4.2.1. Peserta dapat menunjukkan contoh kelebihan sebuah paradigma dalam suatu bagian tugas pengembangan perangkat lunak.

4.2.2. Peserta dapat menunjukkan contoh kekurangan sebuah paradigma dalam suatu bagian tugas pengembangan perangkat lunak.

4.2.3. Peserta dapat membuat tabel perbandingan

4.3. Mampu menerapkan pengetahuan paradigma pemrograman dalam memilih teknologi maupun bahasa pemrograman yang sesuai permasalahan

4.3.1. Diberikan sebuah project, peserta dapat mem-break-down pekerjaan dan mengusulkan paradigma yang sesuai untuk masing-masing bagian

4.3.2. Peserta dapat melakukan analisa kebutuhan paradigma yang sesuai untuk diterapkan.

4.3.3. Diberikan pembagian pekerjaan, peserta dapat menerapkan paradigma functional programming secara tepat guna

4.4. Mampu menggabungkan penyelesaian yang diselesaikan dengan paradigma pemrograman yang berbeda untuk menyelesaikan permasalahan pengembangan perangkat lunak

4.4.1. Diberikan sebuah project yang sebelumnya tidak menerapkan functional programming, peserta dapat me-refactoring bagian tertentu yang cocok untuk diimplementasikan secara functional.

4.4.2. Peserta dapat membuat bagian modul baru secara functional yang bisa dihubungkan dengan program yang dibuat dengan cara lain.

4.4.3. Peserta mengetahui beberapa cara untuk bisa mengkomunikasikan program yang dibuat dengan beberapa bahasa pemrograman berbeda, misalnya melalui API call, RPC atau sejenisnya.

4.5. Mampu menerapkan best-practice terkini terkait bahasa pemrograman fungsional.

4.5.1. Mengenal principle umum dalam functional programming seperti *point-free style* (komposisi fungsi tanpa parameter)

4.5.2. Menerapkan bestpractice pada suatu kasus

4.5.3. Diberikan source code yang belum menerapkan best-practice, peserta dapat merefactor dengan menerapkan best practice tersebut.

4.6. Memahami sejarah paradigma pemrograman dan keterkaitannya dengan perkembangan trend bahasa pemrograman terkini.

4.6.1. Mengenal sejarah bahasa pemrograman dan paradigma terkait

4.6.2. Mengenal penerapan best-practice functional dalam bahasa pemrograman populer misalnya sejak 2018 sudah diterapkan Hook pada react, atau Swift dari Apple dan Kotlin dari Google sudah menginclude functional dalam bahasa baru tersebut.

4.6.3. Berdasar informasi terkini, peserta mengindikasikan perkembangan terkini terkait dengan penerapan paradigma functional programming dalam pengembangan perangkat lunak dan perkembangan bahasa pemrograman.

___ Akhir Dokumen BRP Online Pemrograman Fungsional 2020 __