

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Dalam sebuah penelitian, membuat dokumentasi perlu dilakukan. Dokumentasi yang dibuat bisa dalam bentuk *hardcopy* atau *softcopy*, tergantung kebutuhannya. Dokumentasi adalah kegiatan untuk mencatat suatu peristiwa atau aktifitas yang dianggap berharga atau penting. Dokumentasi yang sudah dibuat dapat menjadi referensi untuk memandu dalam melakukan sebuah aktifitas.

Dalam bidang Teknologi Informasi, dokumentasi kode program java umumnya ditulis dalam format *Javadoc*. *Javadoc* adalah sebuah *tools* yang dimiliki oleh Java yang berguna untuk mengekstrak informasi dari sebuah *file* java menjadi sebuah dokumentasi. Umumnya digunakan untuk mendokumentasikan sebuah nama kelas, *interface*, *method* dan *custom tag*. Oleh karena itu, *Javadoc* sangatlah penting karena dapat memuat berbagai informasi dari sebuah *file* java. Informasi tersebut dapat menjelaskan sebuah kelas yang dibuat dalam sebuah dokumentasi perangkat lunak.

Skripsi mahasiswa Program Studi Teknik Informatika Fakultas Teknologi Informasi dan Sains (FTIS) Universitas Katolik Parahyangan (Unpar) adalah membuat perangkat lunak. Perangkat lunak yang dibuat umumnya menggunakan bahasa pemrograman *java*. Seperti yang sudah dijelaskan, bahasa pemrograman *java* memiliki *Javadoc* sebagai informasi dari kelas, *interface*, *method* dan juga *custom tag* yang dibuat, sehingga informasi tersebut dapat digunakan sebagai penjelasan perangkat lunak pada dokumentasi perangkat lunak. Untuk mendokumentasikan perangkat lunak yang dibuat, seluruh mahasiswa diwajibkan untuk menggunakan  $\text{\LaTeX}$  dalam pembuatan sebuah dokumentasi Skripsi.  $\text{\LaTeX}$  merupakan bahasa *markup* untuk menyusun sebuah dokumentasi.  $\text{\LaTeX}$  membuat apa yang ditampilkan sama seperti apa yang ditulis. Umumnya bentuk akhir dari dokumen yang dibuat oleh  $\text{\LaTeX}$  biasanya berupa sebuah *file* PDF

Pada salah satu bab dokumentasi Skripsi, terdapat penjelasan dari setiap kelas pada perangkat lunak yang dibuat. Penjelasan tersebut sebenarnya dapat diambil dari *Javadoc* yang telah dibuat pada kelas *java*, namun saat ini berdasarkan pengamatan tersebut masih diketik secara manual dari *Javadoc* ke dalam format  $\text{\LaTeX}$ , sehingga membutuhkan lebih banyak waktu untuk mendokumentasikan setiap kelas pada perangkat lunak yang dibuat.

Oleh karena itu, perlu dikembangkan sebuah perangkat lunak yang dapat mengekstraksi informasi pada *Javadoc* ke format  $\text{\LaTeX}$  secara otomatis. Perangkat lunak ini mengimplementasikan sebuah *Application Programming Interface* (API) yang digunakan untuk mengambil informasi berupa nama kelas, *interface*, *method* dan juga *custom tag* yang terdapat pada sebuah *file* *java*

### 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disebutkan di atas, maka dihasilkan beberapa poin yang menjadi rumusan masalah dari masalah ini. Rumusan masalah yang akan dibangun antara lain sebagai berikut:

1. Bagaimana membuat perangkat lunak yang dapat mengonversikan format *Javadoc* ke dalam format  $\text{\LaTeX}$  secara otomatis?

2. Bagaimana antarmuka yang baik untuk perangkat lunak yang akan dibuat?

### 1.3 Tujuan

Adapun tujuan yang ingin dicapai dari penelitian ini adalah menjawab rumusan masalah di atas, yaitu:

1. Membuat perangkat lunak yang dapat mengonversikan format *Javadoc* ke format  $\text{\LaTeX}$  secara otomatis.
2. Mempelajari antarmuka yang baik untuk perangkat lunak yang akan dibuat.

### 1.4 Batasan Masalah

Agar pembahasan masalah tidak terlalu luas, masalah yang akan dikaji di dalam penelitian ini memiliki batasan, yaitu:

1. Perangkat lunak yang dikembangkan menggunakan bahasa pemrograman *Java*.
2. Perangkat lunak hanya dapat menerima masukan data berupa sekumpulan *file java*.
3. Perangkat lunak hanya menghasilkan *output* berupa format  $\text{\LaTeX}$  yang selanjutnya akan dimasukkan ke dalam file  $\text{\LaTeX}$ .

### 1.5 Metodologi

Untuk menyelesaikan penelitian ini disusunlah tahap-tahap tugas yang perlu dilakukan. Tahap-tahap yang dimaksud adalah sebagai berikut:

1. Melakukan studi literatur untuk mengetahui *syntax* yang terdapat pada  $\text{\LaTeX}$  dan mengetahui apa saja isi dari dokumentasi Javadoc Doclet API.
2. Melakukan survei terhadap format penulisan pada suatu bab pada skripsi yang berisi tentang dokumentasi perangkat lunak yang dibuat. Membutuhkan minimal 3 dokumen skripsi sebagai panduan format penulisan.
3. Mengimplementasikan langkah-langkah untuk mengkonversi *Javadoc* ke format  $\text{\LaTeX}$ .
4. Melakukan pengujian terhadap perangkat lunak yang telah diimplementasi.
5. Menarik kesimpulan berdasarkan hasil pengujian.

### 1.6 Sistematika Pembahasan

1. Bab 1 Pendahuluan  
Bab ini akan membahas mengenai latar belakang, rumusan masalah, tujuan, batas masalah, metodologi penelitian dan sistematika penulisan.
2. Bab 2 Dasar Teori  
Bab ini akan membahas mengenai pengertian *Javadoc*, Doclet dan  $\text{\LaTeX}$ .
3. Bab 3 Analisis  
Bab ini akan membahas mengenai analisis struktur  $\text{\LaTeX}$  dan analisis program sejenis TeXDoclet.

---

4. Bab 4 Perancangan

Bab ini akan membahas mengenai tahap-tahap perancangan dan penjelasan perangkat lunak.

5. Bab 5 Implementasi dan Pengujian Perangkat Lunak

Bab ini akan membahas mengenai implementasi kode program dan pengujian perangkat lunak.

6. Bab 6 Kesimpulan dan Saran

Bab ini akan membahas mengenai kesimpulan dari penelitian yang telah dilakukan dan saran-saran untuk pengembangan lebih lanjut dari penelitian ini.



## BAB 2

### LANDASAN TEORI

Bab ini akan membahas teori-teori yang akan menjadi dasar dari penelitian ini. Teori yang dibahas yaitu mengenai *Javadoc*, *Doclet* dan L<sup>A</sup>T<sub>E</sub>X.

#### 2.1 Javadoc

*Javadoc* adalah sebuah *tools* yang dimiliki oleh *Java* yang berguna untuk mengambil informasi dari sekumpulan *source file Java* menjadi sebuah dokumentasi. Umumnya *Javadoc* menghasilkan sekumpulan *file HTML* yang mendeskripsikan sebuah kelas, *interface*, *method* dan *custom tag*. *Javadoc* dapat mengekstraksi informasi tersebut dari sebuah *package java*, sebuah *file java* atau keduanya [1].

##### 2.1.1 Processing of source files

*Javadoc* akan memproses *file* yang memiliki akhiran ".java" dan keseluruhan *file* yang terdapat di dalam folder yang sama. *Javadoc* dapat mengambil informasi dari 1 atau lebih *file java* dan sebuah *package*.

*Javadoc* dapat memproses sebuah *link* secara otomatis yang mengarah kepada sebuah *package*, kelas dan sebuah nama yang akan didokumentasikan pada saat *Javadoc* memprosesnya. *Link-link* tersebut berada pada beberapa posisi seperti:

1. *Declaration* (*return types*, *argument types*, *field types*).
2. Bagian "*See Also*" yang dihasilkan oleh tag *@see*.
3. *In-line text* yang dihasilkan oleh tag *@link*.
4. *Exeption* yang dihasilkan oleh tag *@throws*.
5. *Link "Specified by"* untuk *member* dari sebuah *interface*.
6. *Link "Override"* untuk *member* dari sebuah kelas.
7. Ringkasan daftar tabel *package*, kelas dan seluruh anggota dari kelas.
8. Turunan dari setiap *package* dan kelas.
9. Indeks

Dalam mengekstrak informasi yang terdapat dalam sebuah *package java* atau beberapa *file java* umumnya menghasilkan sebuah dokumentasi standar yang berbentuk *file HTML* dan format penulisan yang mengikuti standar *Javadoc*, akan tetapi untuk menghasilkan sebuah format dokumentasi yang diinginkan, dapat menggunakan sebuah *doclet* yang disediakan oleh *Javadoc*.

### 2.1.2 Terminologi

Terdapat beberapa istilah yang memiliki arti spesifik dalam konteks *Javadoc* sebagai berikut:

- *Generated Document*  
Dokumen yang dihasilkan oleh *Javadoc tools* adalah sebuah *file* HTML dan dibuat oleh *standard doclet*
- *Name*  
Nama dari sebuah perangkat lunak dituliskan dalam bahasa *Java*. Nama-nama tersebut yaitu nama *package*, kelas, *interface*, *field*, *constructor* atau *method*. Nama tersebut dapat berupa informasi lengkapnya seperti *java.lang.String.equals(java.lang.Object)* atau informasi pendeknya seperti *equals(Object)*
- *Documented Classes*  
Detail dari sebuah kelas dan *interface* akan didokumentasikan pada saat *Javadoc* berjalan. Untuk dapat didokumentasikan, *source file* harus tersedia, kemudian nama dari *source file* atau nama dari *package* tersebut harus diletakkan pada *Javadoc command-line*
- *Included Classes*  
kelas dan *Interface* akan didokumentasikan pada saat *Javadoc* berjalan, hal ini sama seperti *Documented Classes*
- *Excluded Classes*  
kelas dan *Interface* tidak akan didokumenasikan pada saat *Javadoc* berjalan.
- *Referenced Classes*  
kelas dan *Interface* yang secara eksplisit disebut oleh kelas dan *interface* lainnya, seperti *return type*, *parameter type*, *cast type*, *extended class*, *implemented interface*, *imported class*, kelas yang digunakan pada *method body*, *@see*, *@link*, *@linkplain* dan *@inheritDoc tag*
- *External Referenced Classes*  
kelas yang tidak dihasilkan saat *Javadoc* berjalan. Dengan kata lain, kelas tersebut tidak diletakkan pada *Javadoc command-line*. *Links* akan dihasilkan jika sebuah kelas mengatakan memiliki *external references* atau *external link*.

### 2.1.3 Source Files

*Javadoc* akan menghasilkan *output* yang berasal dari beberapa tipe *file*, yaitu sebagai berikut:

- *Class Source Code Files*  
Setiap kelas atau *interface* dapat memiliki dokumentasinya masing-masing yang terdapat pada *file java*
- *Package Comment Files*  
Setiap *package* dapat memiliki dokumentasinya masing-masing yang terdapat pada *root folder* kemudian *Javadoc* akan menggabungkan *file-file* yang terdapat pada *root* menjadi sebuah ringkasan. Untuk membuat dokumentasi tersebut, terdapat 2 pilihan yaitu sebuah *file package.html* 2.1 atau sebuah *file package-info.java* 2.2.

```

1  <html>
2  <body>
3  Provides the classes necessary to create an applet and the classes
4  an applet uses to communicate with its applet context.
5
6  @since 1.0
7  @see java.awt
8  </body>
```

9 </html>

Listing 2.1: *File* package.html

```

1  /**
2   * Provides the classes necessary to create an applet
3   * and the classes an applet uses to communicate
4   * with its applet context.
5   *
6   * @since 1.0
7   * @see java.awt
8   */
9  package java.lang.applet;
```

Listing 2.2: *File* package-info.java

Ketika *Javadoc* memproses *package* tersebut, *Javadoc* akan melakukan beberapa langkah yaitu sebagai berikut:

1. Menyalin informasi untuk diproses. Jika *file* berupa HTML maka pada bagian <body> hingga </body> akan disalin.
2. Memproses semua *tag* pada *package* yang ada.
3. Memasukan teks yang sudah diproses tersebut pada bagian bawah halaman dokumentasi yang dihasilkan.
4. Salin kalimat pertama pada *package* tersebut pada bagian atas halaman dokumentasi

- *Overview Comment Files*

Setiap aplikasi atau sekumpulan *package* yang akan didokumentasikan akan memiliki dokumentasi *overview*. Dokumentasi tersebut dapat dibuat lebih dari 1, jika pada saat pembuatan perangkat lunak menggunakan sekumpulan *package* yang berbeda. Untuk membuat sebuah dokumentasi ini, perlu membuat sebuah *file* HTML yang umumnya bernama *overview.html*. Kemudian *Javadoc* akan memproses seperti pada *Package Comment Files*

- *Miscellaneous Unprocessed Files*

*File* tersebut dapat berubah sebuah *graphic files*, *file java* dan sebuah *file* HTML.

#### 2.1.4 Generated Files

Secara *default*, *Javadoc* akan menggunakan *standard doclet* yang akan menghasilkan sebuah dokumentasi berformat HTML. Doclet tersebut akan menghasilkan *file* HTML secara terpisah. Terdapat 3 grup yang masing-masing grup memiliki kriterianya sendiri, 3 grup tersebut adalah sebagai berikut:

- *Basic Content Pages*

- sebuah halaman kelas atau *interface* (*classname.html*) untuk masing-masing kelas atau *interface* yang akan didokumentasikan
- sebuah halaman *package* (*package-summary.html*) untuk masing-masing *package* yang akan didokumentasikan
- sebuah halaman *overview* (*overview-summary.html*) untuk keseluruhan sekumpulan *package*. Halaman ini adalah halaman utama yang dihasilkan.

- *Cross-Reference Pages*

- sebuah halaman hirarki dari kelas untuk sekumpulan dari semua *package* (*overview-tree.html*)
- sehalaman hirarki dari kelas untuk setiap *package* (*package-tree.html*)

- sehalaman *"use"* (*package-use.html*) yang berisikan *package*, *classes*, *methods*, *constructors* atau *interface*. Jika diberikan sebuah kelas bernama A, maka halaman tersebut akan berisikan *subclasses* dari A, *methods* yang memiliki *return* A dan *methods* atau *constructors* dengan parameter bertipe A.
- sebuah halaman *deprecated API* (*deprecated-list.html*). Halaman ini adalah halaman dari sekumpulan nama yang tidak direkomendasikan untuk digunakan.
- sebuah halaman sekumpulan nilai *constant* (*constant-values.html*) untuk sekumpulan nilai *static*.
- sebuah halaman *serialized form* (*serialized-form.html*)
- sebuah halaman *index* (*index-\*.html*).

- *Support Files*

- sebuah halaman bantuan (*help-doc.html*).
- sebuah halaman *index* (*index.html*) yang membuat sebuah HTML *frames*.
- beberapa *frame file* (*\*-frame.html*) yang berisi sekumpulan *packages*, kelas dan *interface* dan digunakan pada saat HTML *frames* ditampilkan
- sebuah *file* teks *package list* (*package-list*).
- sebuah *style sheet file* (*stylesheet.css*) untuk mengontrol warna, jenis *font*, ukuran *font* dan posisi dari halaman yang dihasilkan
- sebuah *doc-files* yang berisikan gambar dan beberapa contoh *file java*

*Javadoc* akan menghasilkan 2 atau 3 HTML *frame*. *Javadoc* akan membuat minimum *frame* yang dibutuhkan. Jika hanya terdapat 1 *package*, maka *Javadoc* akan membuat 1 *frame* yang berisi dari sekumpulan kelas pada *package* tersebut. Jika terdapat lebih dari 2 *package*, maka *Javadoc* akan membuat 3 *frame* dari sekumpulan *package*. Jika kelas yang digunakan adalah *java.applet.Applet* dan semua dokumentasi yang dihasilkan akan berada pada folder yang bernama *apidocs*, struktur *file* yang dihasilkan adalah sebagai berikut:

1	apidocs	Top directory
2	index.html	Initial page that sets up HTML frames
3	* overview-summary.html	Lists all packages with first sentences summaries
4	overview-tree.html	Lists class hierarchy for all packages
5	deprecated-list.html	Lists deprecated API for all packages
6	constant-values.html	Lists values of static fields for all packages
7	serialized-form.html	Lists serialized form for all packages
8	* overview-frame.html	Lists all packages, used in upper-left frame
9	allclasses-frame.html	Lists all classes for all packages, used in lower-left frame
10		
11	help-doc.html	Lists user help for how these pages are organized
12	index-all.html	Default index created without -splitindex option
13	index-files	Directory created with -splitindex option
14	index-<number>.html	Index files created with -splitindex option
15	package-list	Lists package names, used only for resolving external refs
16		
17	stylesheet.css	HTML style sheet for defining fonts, colors and positions
18		
19	java	Package directory
20	applet	Subpackage directory
21	Applet.html	Page for Applet class
22	AppletContext.html	Page for AppletContext interface
23	AppletStub.html	Page for AppletStub interface
24	AudioClip.html	Page for AudioClip interface
25	* package-summary.html	Lists classes with first sentence summaries for this package
26		
27	* package-frame.html	Lists classes in this package, used in



28		lower left-hand frame
29	* package-tree.html	Lists class hierarchy for this package
30	package-use	Lists where this package is used
31	doc-files	Directory holding image and example files
32	class-use	Directory holding pages API is used
33	Applet.html	Page for uses of Applet class
34	AppletContext.html	Page for uses of AppletContext interface
35	AppletStub.html	Page for uses of AppletStub interface
36	AudioClip.html	Page for uses of AudioClip interface
37	src-html	Source code directory
38	java	Package directory
39	applet	Subpackage directory
40	Applet.html	Page for Applet source code
41	AppletContext.html	Page for AppletContext source code
42	AppletStub.html	Page for AppletStub source code
43	AudioClip.html	Page for AudioClip source code

Listing 2.3: Struktur *file* yang dihasilkan

## 2.2 Doclet

*Doclet* yang terdapat pada *Javadoc* dapat digunakan untuk menghasilkan sebuah *output Javadoc* yang dapat disesuaikan. Standar *doclet* yang dihasilkan oleh *Javadoc* adalah dokumentasi dengan format HTML. Selain menghasilkan *output* yang dapat disesuaikan, *Doclet* juga dapat mengekstrak informasi secara spesifik [2].

### 2.2.1 Interface-interface pada Doclet

Berikut adalah beberapa *interface* yang terdapat pada *Doclet*:

- **RootDoc** sebuah *interface* yang menyatakan sebuah *root* dari perangkat lunak yang dibuat. Dari *root* tersebut semua informasi dapat diekstrak. *Method-method* yang digunakan adalah sebagai berikut
  - `classes()`  
*Method* ini akan mengembalikan sejumlah kelas dan *interface* pada *package*
- **ClassDoc** sebuah *interface* yang menyatakan informasi dari sebuah kelas. Informasi tersebut dapat berupa nama kelas, nama *method* dan *tag*. *Method-method* yang digunakan adalah sebagai berikut
  - `name()`  
*Method* ini akan mengembalikan sebuah nama kelas atau *interface* pada *package*
  - `commentText()`  
*Method* ini akan mengembalikan sebuah informasi dari deskripsi kelas
  - `methods()`  
*Method* ini akan mengembalikan sebuah *array of methods*
- **MethodDoc** sebuah *interface* yang menyatakan informasi dari sebuah *method*. *Method-method* yang digunakan adalah sebagai berikut
  - `name()`  
*Method* ini akan mengembalikan sebuah nama *method*
  - `modifiers()`  
*Method* ini akan mengembalikan sebuah *access modifier* dari sebuah *method*

- `returnType()`  
*Method* ini akan mengembalikan sebuah *return type* dari sebuah *method*
- `flatSignature()`  
*Method* ini akan mengembalikan *signature* dari sebuah *method*. Jika terdapat *Method* dengan parameter (String x, int y), maka akan mengembalikan (String, int)
- **ParamTag** sebuah *interface* yang menyatakan informasi dari sebuah *Tag* parameter. *Method-method* yang digunakan adalah sebagai berikut
  - `name()`  
*Method* ini akan mengembalikan sebuah *tag @param*
  - `parameterName()`  
*Method* ini akan mengembalikan sebuah nama parameter dari sebuah *method*
  - `parameterComment()`  
*Method* ini akan mengembalikan sebuah deskripsi dari parameter yang terdapat pada *method*

### 2.2.2 Penggunaan Doclet

Doclet dapat menghasilkan sebuah *output Javadoc* yang dapat disesuaikan. Penggunaan *Doclet* API dapat mengekstrak bermacam-macam informasi seperti nama kelas, nama *method*, deskripsi singkat untuk sebuah parameter dari sebuah *method* hingga *return type* dari *method*.

Berikut adalah langkah-langkah untuk menggunakan *doclet*:

1. Membuat sebuah kelas pada *java* sebagai *doclet*. *class java* tersebut harus meng-*import* `com.sun.javadoc.*` untuk menggunakan *doclet* API.
2. *Doclet* tersebut diawali dengan sebuah *method* `public static boolean start` yang memiliki parameter `RootDoc`.
3. *Compile doclet* tersebut dengan menggunakan *compiler* Java 2 SDK yaitu *javac* pada *command prompt* (Windows)/*terminal* (Linux).
4. Jalankan *Javadoc* menggunakan `-doclet startingclass` *option* untuk menghasilkan *output* yang telah disesuaikan, dimana *startingclass* adalah sebuah kelas yang sudah dibuat pada langkah 1.

*File doclet* API terdapat pada direktori *folder jdk* yang ter-*install* pada komputer pada *subfolder* `lib\tools.jar.doclet` yang sudah dibuat harus di-*compile* menggunakan *file tools.jar* dan menambahkan *option -classpath* setelah *command javac*. Jika tidak menggunakan *option -doclet*, *Javadoc* akan menghasilkan *output* standar yaitu berupa *file HTML*.

*Package* `com.sun.javadoc` terdiri *interface* yang mendefinisikan *doclet* API dan sedangkan *file tools.jar* berisikan *interface-interface* tersebut dan juga berisikan *private package* dengan *class-class* yang mengimplementasi *interface* tersebut serta *file tools.jar* berisikan pula *class-class* yang mengimplementasi sebuah standar *doclet*.

```

1  import com.sun.javadoc.*;
2
3  public class ListClass {
4      public static boolean start(RootDoc doc) {
5          ClassDoc[] classes = doc.classes();
6          for(int i=0, i < classes.length; i++) {
7              System.out.println(classes[i]);
8          }
9          return true;
10     }

```

11 }

Listing 2.4: kelas ListClass.java

Potongan *program* ini 2.4 adalah sebuah *doclet* sederhana untuk menampilkan nama-nama kelas pada *file java*. Hal pertama yang harus dilakukan adalah meng-*import package* `com.sun.javadoc.*`, kemudian membuat sebuah *method* `public static boolean start` dengan parameter sebuah `RootDoc doc` yang akan menampung sekumpulan *file java* yang akan diproses. `ClassDoc` pada *method* tersebut akan menampung nama-nama kelas yang terdapat pada variabel `doc` dengan menggunakan *method* `classes()`.

## 2.3 L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X adalah sebuah bahasa *markup* untuk sistem penulisan dokumen yang dikembangkan oleh Leslie B. Lamport dan dirilis pada tahun 1985 [3]. L<sup>A</sup>T<sub>E</sub>X Memiliki filosofi WYMIWYG (*What you Mean Is What You Get*) yang berarti sesuatu yang ditulis akan berdasarkan arti dari hal tersebut. Oleh karena itu, untuk menambahkan suatu perintah pada dokumen yang sedang ditulis perlu menambahkan suatu *command*. *Command* adalah kata spesial yang menentukan suatu sifat pada L<sup>A</sup>T<sub>E</sub>X. Hampir semua *command* pada L<sup>A</sup>T<sub>E</sub>X selalu diawali dengan tanda '\' dan beberapa *command* memiliki *parameter*. *Parameter* diawali dengan tanda kurung kurawal buka dan diakhiri dengan kurung kurawal tutup (`{...}`). File L<sup>A</sup>T<sub>E</sub>X memiliki ekstensi `.tex`. Pada saat membuat sebuah *project* L<sup>A</sup>T<sub>E</sub>X hanya perlu menuliskan *command* `\documentclass[option]{class}` 1 kali.

Untuk menulis dokumen pada L<sup>A</sup>T<sub>E</sub>X dibutuhkan beberapa *command* yang wajib ada dalam sebuah dokumen, yaitu:

1. `\documentclass[option]{class}`  
Digunakan untuk menentukan jenis dokumen yang *layout* dokumen. Bagian *option* dapat dikosongkan atau dapat digunakan untuk menyimpan pilihan pengaturan *layouting*. Pada Bagian kelas digunakan untuk menentukan tipe dokumen yang akan dibuat. *Command* ini hanya perlu ditulis 1 kali dalam sebuah dokumen.
2. `\maketitle`  
Digunakan untuk menampilkan halaman judul. Biasanya halaman judul akan memuat judul dokumen, nama pengarang dan tanggal pembuatan dokumen. Judul dokumen, nama pengarang dan tanggal pembuatan dapat ditampilkan dengan menambahkan perintah `\title{judul}`, `\author{nama}` dan `\date{tanggal}`.
3. `\begin{document}...\end{document}`  
Digunakan untuk mengawali dan mengakhiri sebuah dokumen.
4. `\section{section}`  
Digunakan untuk menampilkan subbab sebuah dokumen.
5. `\texttt{text}`  
Digunakan untuk menampilkan tulisan *monospaced*.
6. `\begin{enumerate}...\end{enumerate}`  
Digunakan untuk menampilkan *ordered list*. *List* ini akan menampilkan angka yang terurut. Di dalam *list* ini terdapat *command* `\item` untuk menambahkan isi dari *list* tersebut.
7. `\begin{itemize}...\end{itemize}`  
Digunakan untuk menampilkan *unordered list*. *List* ini akan menampilkan simbol spesial. Di dalam *list* ini terdapat *command* `\item` untuk menambahkan isi dari *list* tersebut.



## BAB 3

### ANALISIS

Bab ini membahas mengenai analisis kebutuhan perangkat lunak dan analisis program sejenis.

#### 3.1 Analisis Kebutuhan Perangkat Lunak

Struktur L<sup>A</sup>T<sub>E</sub>X yang digunakan memiliki format sebagai berikut.

```
1 \begin{enumerate}
2 \item \texttt{namaKelas}\\
3 {penjelasan kelas}
4
5 Atribut yang dimiliki kelas ini adalah sebagai berikut.
6 \begin{itemize}
7 \item \texttt{atribut} –
8 {penjelasan tentang atribut}.
9 \end{itemize}
10
11 \textit{Method} yang terdapat pada kelas Pertambahan adalah sebagai berikut.
12 \begin{itemize}
13 \item \texttt{method}\\
14 {penjelasan method}
15
16 \textbf{Parameter:}
17 \begin{itemize}
18 \item \texttt{parameter} –
19 {penjelasan dari parameter}.
20 \end{itemize}
21
22 \textbf{Return Value:} {penjelasan return-type method}\\
23 \textbf{Exception:} {penjelasan exception jika terdapat exception}
24 \textbf{See Also:} {penjelasan tag @see jika terdapat tag tersebut}
25 \textbf{Override:} {penjelasan apabila jika terdapat {\it override method} }
26 \end{itemize}
```

Listing 3.1: Potongan kode L<sup>A</sup>T<sub>E</sub>X

Potongan kode yang terdapat pada listing 3.1 adalah struktur lengkap L<sup>A</sup>T<sub>E</sub>X yang digunakan, akan dijelaskan sebagai berikut.

1. *List level* pertama

Pada *list level* pertama ini menampilkan sebuah nama kelas dan penjelasan terkait dengan kelas tersebut. *List* yang dibuat menggunakan *ordered list* dengan *command* `\begin{enumerate}...` `\end{enumerate}` dan *command* `\texttt{namaKelas}` akan digunakan untuk menampilkan nama kelas.

2. *List level* kedua

Pada *list level* kedua ini terdapat dua *list* yang masing-masing menampilkan atribut dan *method* yang dimiliki oleh kelas tersebut. *List* pertama yang dibuat menggunakan *unordered list* dengan

*command* `\begin{itemize}...\end{itemize}` untuk mengisi atribut-atribut yang terdapat pada kelas ini jika kelas ini tidak memiliki atribut maka menampilkan tulisan tidak memiliki atribut. *Command* `\texttt{atribut}` digunakan untuk menampilkan atribut. Atribut ini menampilkan tipe atribut dan nama atribut.

*List* kedua menggunakan *unordered list* dengan *command* `\begin{itemize}...\end{itemize}` untuk mengisi *method-method* yang terdapat pada kelas ini dan penjelasan terkait dengan *method* tersebut. *Command* `\texttt{method}` digunakan untuk menampilkan *method*. *Method* ini menampilkan *access modifier* dari *method*, tipe kembalian *method*, nama *method* dan daftar nama parameter.

### 3. *List level* ketiga

Pada *list level* ketiga ini menampilkan parameter yang digunakan pada *method* dan penjelasan terkait dengan parameter tersebut. *List* yang dibuat menggunakan *unordered list* dengan *command* `\begin{itemize}...\end{itemize}` jika *method* tidak memiliki parameter maka menampilkan tulisan tidak memiliki parameter dan *command* `\texttt{parameter}` akan digunakan untuk menampilkan parameter. Parameter ini menampilkan tipe parameter dan nama parameter.

### 4. *Return Value & Exception*

*Return value* yang terdapat dalam *method* tersebut akan ditampilkan setelah *list level* ketiga jika tipe *return value* adalah *void* maka akan menampilkan tulisan tidak memiliki *return value*. *Exception* maka ditampilkan setelah *Return value* jika *method* tidak terdapat *exception* maka akan menampilkan tulisan tidak memiliki *exception*.

### 5. *Optional Tags*

*Optional tags* akan menampilkan informasi dari sebuah *tag @see* ataupun *tag {@link}*. Jika tidak ada informasi dari *tag - tag* tersebut akan menampilkan tulisan tidak ada.

### 6. *Override*

*Override* akan menampilkan informasi apakah *method* dari sebuah *superclass* ditulis kembali di sebuah *subclass*. jika tidak ada informasi tersebut maka bagian penjelasan akan dihilangkan.

Perangkat lunak yang dibuat akan menerima sebuah masukan berupa sekumpulan *file java* yang berada di dalam sebuah *package*. Struktur kode *java* yang digunakan dapat dilihat pada lampiran A. Struktur kode akan dijelaskan sebagai berikut.

1. Setiap *file java* harus terletak di dalam sebuah *package* yang sama.
2. Setiap deklarasi kelas harus diawali dengan huruf kapital serta memiliki javadoc untuk penjelasan tentang kelas tersebut dan secara opsional dapat menambahkan *tag - tag* javadoc seperti *tag @see* sebagai penunjuk ke sebuah referensi dan *tag {@link}* sebagai penunjuk ke dokumentasi sebuah *package*, *class* ataupun *method* yang dimiliki oleh kelas lain.
3. Setiap deklarasi atribut harus memiliki *access modifier*, tipe atribut dan nama atribut serta memiliki javadoc untuk penjelasan tentang atribut tersebut.
4. Setiap deklarasi *method* harus memiliki *access modifier*, tipe kembalian, nama *method*, tipe dan variabel parameter serta memiliki javadoc untuk penjelasan *method*, parameter yang digunakan dan hasil kembalian sebuah *method*.

Hasil dari sebuah perangkat lunak yang dibuat adalah sebuah *file* berformat *L<sup>A</sup>T<sub>E</sub>X*. Perangkat lunak akan membaca satu persatu *file java* dan informasi yang terdapat pada setiap *file java* tersebut dimasukkan ke dalam *file L<sup>A</sup>T<sub>E</sub>X*.

```

1 \begin{enumerate}
2   \item \texttt{Pertambahan}\\
3     Kelas ini merupakan Kelas Pertambahan.
4
5     Atribut yang dimiliki kelas ini adalah sebagai berikut.
6   \begin{itemize}
7     \item \texttt{int a} –
8       Atribut A.
9     \item \texttt{int b} –
10      Atribut B.
11   \end{itemize}
12
13   \textit{Method} yang terdapat pada kelas Pertambahan adalah sebagai berikut.
14   \begin{itemize}
15     \item \texttt{public int pertambahan(int a, int b)}\\
16       Method Pertambahan.
17
18     \textbf{Parameter:}
19     \begin{itemize}
20       \item \texttt{int a} –
21         Bilangan Pertama.
22       \item \texttt{int b} –
23         Bilangan Kedua.
24     \end{itemize}
25
26     \textbf{Return Value:} hasil penjumlahan 2 buah bilangan.\\
27     \textbf{Exception:} tidak memiliki \textit{exception}.
28     \textbf{Override:} \texttt{pertambahan} dari kelas \texttt{operasiMatematikaInterface}
29   \end{itemize}
30 \end{enumerate}

```

Listing 3.2: Contoh hasil konversi *Javadoc* ke  $\text{\LaTeX}$ 

Hasil konversi 3.2 akan menampilkan nama kelas serta penjelasan kelas tersebut, atribut yang digunakan serta penjelasan untuk setiap atributnya, *method* yang digunakan serta penjelasan *method*, parameter yang digunakan serta penjelasan setiap parameternya, *return value* dan *exception*.

## 3.2 Analisis Program Sejenis TeXDoclet

TeXDoclet merupakan sebuah program yang mengimplementasi *Doclet* yang dimiliki oleh *Java*. Program ini akan mengkonversi sekumpulan *file java* yang terletak di dalam satu *package* yang sama. TeXDoclet dapat menghasilkan dokumen berupa *file  $\text{\LaTeX}$*  atau *file PDF*. Untuk dapat menghasilkan *file PDF*, TeXDoclet mengintegrasikan Lua $\text{\LaTeX}$  untuk menghasilkan dokumen PDF dari sebuah *file  $\text{\LaTeX}$* .

TeXDoclet memiliki beberapa *option* yang dapat digunakan, akan dijelaskan sebagai berikut.

1. **-sectionlevel <level>**

Untuk menentukan *level* teratas dari *section* sebuah dokumen. *Section* tersebut bisa berupa *chapter*, *section* atau *subsection*

2. **-createPdf**

Untuk menghasilkan *file PDF* dari sebuah hasil *file  $\text{\LaTeX}$*  dengan menggunakan Lua $\text{\LaTeX}$ .

3. **-twosided**

Untuk menghasilkan dokumen 2 sisi. Jika dokumen tersebut menggunakan *option* ini maka dokumen tersebut pada saat dicetak akan memiliki 2 sisi yaitu depan dan belakang.

4. **-texinit <file>**

Untuk menambahkan *command-command* yang lain sebelum *command* `\begin{document}`.

5. `-docclass <class>`

Untuk menentukan tipe dokumen yang akan dibuat. *Default* untuk *option* adalah tipe dokumen *report*.



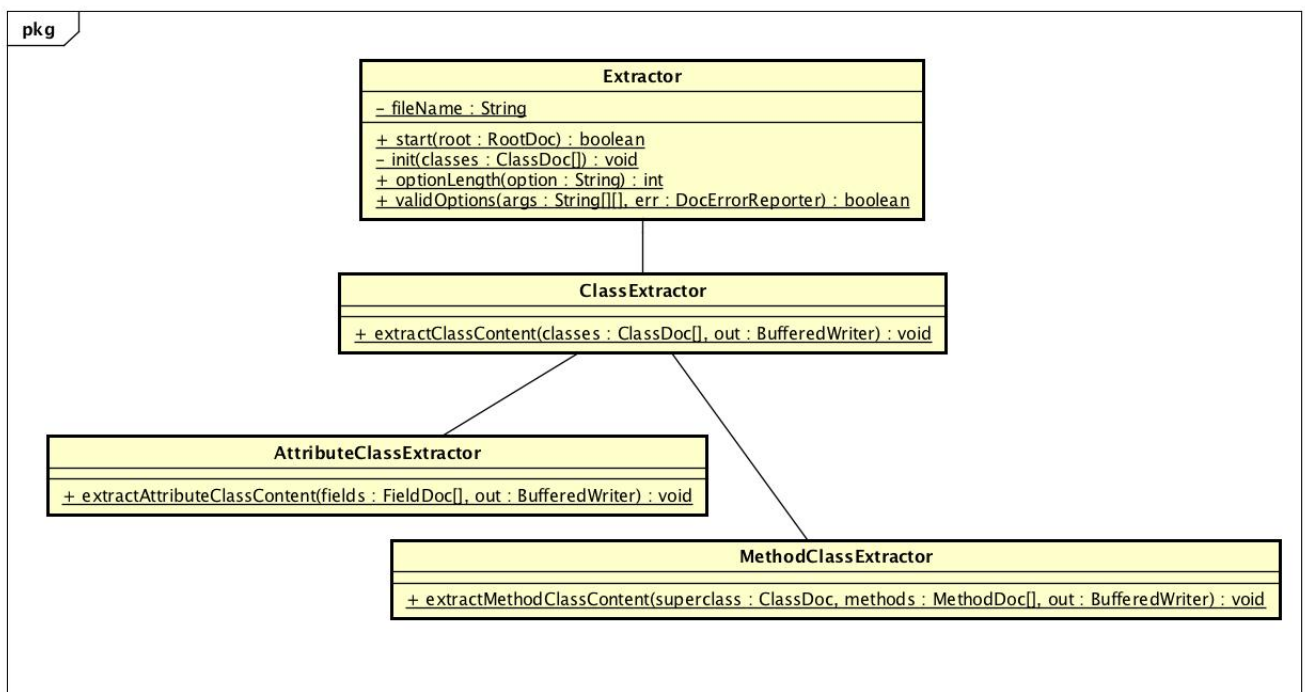
## BAB 4

### PERANCANGAN

Bab ini membahas mengenai perancangan aplikasi yang akan dibangun meliputi diagram kelas rinci beserta deskripsi dan fungsinya.

#### 4.1 Rancangan Kelas Lengkap

Rancangan kelas dibawah ini akan menampilkan keseluruhan kelas yang akan digunakan. Deskripsi kelas beserta fungsi dari diagram kelas tersebut adalah sebagai berikut:



Gambar 4.1: Kelas Diagram

##### 1. AttributeClassExtractor

Kelas ini merupakan kelas untuk mengambil informasi sebuah atribut yang terdapat pada kelas

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public static void extractAttributeClassContent(com.sun.javadoc.FieldDoc[] fields, java.io.BufferedWriter out)`

*Method* ini akan menampilkan atribut-atribut yang dimiliki oleh sebuah kelas

**Parameter:**

- `FieldDoc fields` - sebuah array berisikan sejumlah atribut dari kelas

- `BufferedWriter out` - turunan dari kelas `Writer` yang digunakan untuk menulis file text

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

## 2. ClassExtractor

Kelas ini merupakan kelas untuk mengambil informasi dari sebuah kelas

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public static void extractClassContent(com.sun.javadoc.ClassDoc[] classes, java.io.BufferedWriter out)`

*Method* ini akan menampilkan nama kelas berserta penjelasan dari sebuah kelas

**Parameter:**

- `ClassDoc classes` - sebuah array berisikan sejumlah kelas
- `BufferedWriter out` - turunan dari kelas `Writer` yang digunakan untuk menulis file text

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

## 3. Extractor

Kelas ini merupakan kelas untuk menjalankan *custom doclet*

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- `String fileName` - atribut untuk nama *file*

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public static boolean start(com.sun.javadoc.RootDoc root)`

*Method* ini berperan sebagai *method* untuk menjalankan *custom doclet*

**Parameter:**

- `RootDoc root` - berperan sebagai mengambil seluruh informasi spesifik dari *option* yang terdapat pada *command-line* sebuah *terminal*. Selain itu berperan juga untuk mengambil informasi dari sekumpulan *file java* yang akan di proses.

**Return Value:** kondisi true

**Exception:** Tidak memiliki *exception*

- `private static void init(com.sun.javadoc.ClassDoc[] classes)`

*Method* ini berperan untuk menulis kedalam sebuah *file* saat *javadoc* berjalan.

**Parameter:**

- `ClassDoc classes` - sebuah array yang berisikan sekumpulan *file java* yang akan di proses.

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public static int optionLength(java.lang.String option)`

*Method* untuk menghitung banyak *option* yang digunakan pada *command-line*

**Parameter:**

- `String option` - sebuah *option*

**Return Value:** panjang setiap *option*

**Exception:** Tidak memiliki *exception*

- `public static boolean validOptions(java.lang.String[] [] args, com.sun.javadoc.DocErrorReporter err)`

Pengecekan option valid

**Parameter:**

- `String args` - String array 2 dimensi dari option
- `DocErrorReporter err` - sebuah error jika tidak terdapat option tersebut.

**Return Value:** bernilai `true` jika option tersebut dikenali, `false` jika option tersebut tidak dikenali

**Exception:** Tidak memiliki *exception*

#### 4. MethodClassExtractor

Kelas ini merupakan kelas untuk mengambil informasi sebuah *method* terdapat pada kelas

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public static void extractMethodClassContent(com.sun.javadoc.ClassDoc superclass, com.sun.javadoc.MethodDoc[] methods, java.io.BufferedWriter out)`

*Method* ini akan menampilkan *method-method* yang dimiliki oleh sebuah kelas

**Parameter:**

- `ClassDoc superclass` - sebuah objek `ClassDoc`
- `MethodDoc methods` - sebuah array berisikan sejumlah *method* dari kelas
- `BufferedWriter out` - turunan dari kelas `Writer` yang digunakan untuk menulis file text

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `private static void ParameterMethod(java.io.BufferedWriter out, com.sun.javadoc.ParameterDoc paramMethod, com.sun.javadoc.ParamTag[] paramTags)`

*Method* ini akan menampilkan parameter *method-method* yang dimiliki oleh sebuah kelas

**Parameter:**

- `BufferedWriter out` - turunan dari kelas `Writer` yang digunakan untuk menulis file text
- `Parameter paramMethod` - sebuah array berisikan sejumlah *method* dari kelas
- `ParamTag paramTags` - sebuah array berisikan sejumlah parameter *method* dari kelas

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `private static void ReturnTypeMethod(java.io.BufferedWriter out, com.sun.javadoc.TypeDoc type, com.sun.javadoc.Tag[] returnTags)`

*Method* ini akan menampilkan *return type* dari *method-method* yang dimiliki oleh sebuah kelas

**Parameter:**

- `BufferedWriter out` - turunan dari kelas `Writer` yang digunakan untuk menulis file text
- `Type type` - sebuah objek `Type`
- `Tag returnTags` - sebuah array berisikan sejumlah *return type* dari *method* dari kelas

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `private static void ExceptionMethod(java.io.BufferedWriter out, com.sun.javadoc.Tag[] throwTags)`

*Method* ini akan menampilkan *return type* dari *method-method* yang dimiliki oleh sebuah kelas

**Parameter:**

- `BufferedWriter out` - turunan dari kelas `Writer` yang digunakan untuk menulis file text
- `Tag throwTags` - sebuah array berisikan sejumlah *exception* dari *method* dari kelas

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

## 4.2 Rancangan Antarmuka

Rancangan antarmuka perangkat lunak yang dibuat adalah melalui sebuah *terminal* pada *Linux* dan *command prompt* pada *Windows*. Berikut adalah antarmuka jika menggunakan *terminal* pada *Linux*:

```
Last login: Sun Apr 29 17:06:20 on ttys001
abathz:~ abathz$ cd Documents/KULIAH/Skripsi/javadoc-to-latex/
```

Gambar 4.2: Mengarahkan kedalam folder dari perangkat lunak

Langkah pertama adalah berpindah dari direktori awal ke direktori perangkat lunak yang dibuat. Untuk berpindah direktori perlukan *command* `cd` atau kepanjangan dari *change directory* lalu diikuti dengan lokasi direktori yang diinginkan. Pada gambar 4.2 direktori perangkat lunak terdapat di dalam folder Document lalu folder KULIAH lalu folder Skripsi dan terakhir folder javadoc-to-latex kemudian tekan tombol *enter* lalu direktori akan langsung berpindah ke direktori yang dituju.

```
abathz:javadoc-to-latex abathz$ javadoc -filename bab4 -classpath dist/javadoc-to-latex.jar -doclet
extractor.Extractor -docletpath dist/javadoc-to-latex.jar ../javadoc/*
```

Gambar 4.3: Memasukkan *option* yang akan digunakan

Langkah kedua adalah menjalankan perangkat lunak yang dibuat. Diawali dengan *command* `javadoc` lalu diikuti 5 buah argumen. Argumen pertama(hijau) adalah *option* untuk menamai *file* sesuai dengan yang ditentukan. Sebagai contoh pada gambar 4.3, *file* akan bernama "bab4", jika argumen pertama tidak dimasukkan pada *command-line* maka nama dari *file* tersebut secara otomatis menjadi "doc". Argumen kedua(biru muda) berperan sebagai penunjuk kelas-kelas yang digunakan. Argumen kedua ini bersifat *optional*, jika kode program yang akan didokumentasikan menggunakan *external library* maka argumen ini digunakan. Argumen ketiga(jingga) adalah sebuah kelas untuk menjalankan *custom doclet* dari perangkat lunak yang dibuat. Argumen ketiga tersebut akan menjalankan kelas bernama `Extractor` yang terdapat didalam *package* `extractor`. Kemudian argumen keempat(kuning) adalah *custom doclet* yang berperan untuk mengambil informasi kelas, atribut, *method* dari sekumpulan *file java*. Argumen kelima(biru) adalah lokasi sekumpulan *file java* yang akan diproses. Pada gambar 4.3, lokasi *file-file* tersebut terdapat pada folder `javadoc`. Folder `javadoc` tersebut berada direktori folder `Skripsi`.

```
abathz:javadoc-to-latex abathz$ javadoc -filename bab4 -classpath dist/javadoc-to-latex.jar -doclet
extractor.Extractor -docletpath dist/javadoc-to-latex.jar ../javadoc/*
Loading source file ../javadoc/Pembagian.java...
Loading source file ../javadoc/Pengurangan.java...
Loading source file ../javadoc/Perkalian.java...
Loading source file ../javadoc/Pertambahan.java...
Loading source file ../javadoc/operasiMatematikaInterface.java...
Constructing Javadoc information...
abathz:javadoc-to-latex abathz$ █
```

Gambar 4.4: Hasil tampilan jika proses konversi selesai

Perangkat lunak yang dibuat akan membaca seluruh isi folder yang dituju, pada contoh gambar 4.4, terdapat 5 *file java* yang terdapat didalam folder javadoc. Lalu perangkat lunak akan melakukan ekstrasi informasi terhadap masing-masing *file* tersebut. Jika proses ekstraksi selesai maka proses berhenti.



## BAB 5

### IMPLEMENTASI & PENGUJIAN

Bab ini terdiri atas dua bagian, yaitu Implementasi Perangkat Lunak dan Pengujian Perangkat Lunak. Bagian implementasi berisi penjelasan bagaimana perangkat lunak dibuat dan langkah-langkah dalam penggunaan perangkat lunak. Sedangkan bagian pengujian berisi hasil pengujian fungsional terhadap perangkat lunak yang telah dibuat.

#### 5.1 Implementasi Perangkat Lunak

Perangkat lunak dibuat menggunakan bahasa *java* dan dimasukkan ke dalam sebuah *jar* sehingga dapat digunakan dengan cara mengeksekusi perintah. Penggunaan *file jar* tersebut dapat dilakukan melalui *Terminal* di Linux atau *Command Prompt* di Windows. Berikut perintah yang digunakan untuk menjalankan perangkat lunak.

```
javadoc [options] {sourcefiles|packagenames}
```

Pada potongan perintah diatas memiliki 2 parameter yaitu *option* dan *packagenames*. Parameter *packagename* adalah parameter untuk *package* yang akan sebagai masukan dari perangkat lunak. Parameter *option* adalah beberapa perintah pendukung. Berikut beberapa perintah *option* yang digunakan untuk mendukung berjalannya perangkat lunak.

- `-filename <file-name>` - Menghasilkan *output file* dengan nama `file-name.tex`. Jika *option* ini tidak digunakan maka nama *file* yang dihasilkan akan bernama `doc.tex`
- `-doclet <class>` - Kelas yang dibuat untuk menghasilkan *output*.
- `-docletpath <path>` - Letak *doclet* yang sudah di-*package* menjadi *file jar*
- `-sourcepath <pathlist>` - Letak *source file* sebagai masukan.
- `-subpackages <subpkglist>` - Letak *subpackage* yang akan dimuat secara rekursif.

Untuk penggunaan perintah diatas, Langkah pertama membuka aplikasi *Terminal* atau *Command Prompt*. Langkah kedua mengetik perintah `javadoc` lalu diikuti dengan perintah pendukungnya seperti yang sudah dijelaskan diatas. Berikut contoh perintah lengkap yang digunakan.

```
javadoc -filename <file-name> -doclet extractor.Extractor -docletpath  
GenerateJavadocToLatex.jar -sourcepath <path/to/directory> -subpackages <  
packagenames> <sourcefiles|packagenames>
```

Listing 5.1: Perintah Javadoc

#### 5.2 Pengujian Perangkat Lunak

Pada sub bab ini akan menjelaskan Lingkungan Pengujian dan Pengujian Fungsional. Pengujian Fungsional akan menguji perangkat lunak terhadap kode program sederhana serta menguji kode program perangkat lunak yang dibuat.

### 5.2.1 Lingkungan Pengujian

Dalam proses pengujian perangkat lunak ini digunakan spesifikasi perangkat sebagai berikut.

1. Processor: Intel Core i7 2.5-3.7GHz
2. RAM: 16.00 GB DDR3
3. Harddisk : 512MB SSD
4. VGA : Intel Iris Pro dan AMD Radeon R9 M370X
5. Sistem Operasi: macOS High Sierra
6. Versi Java: 1.8.0\_121
7. Code Editor: Netbeans 8.2

### 5.2.2 Pengujian Fungsional

Pada pengujian fungsional dilakukan pengujian terhadap kode program sederhana dan kode program perangkat lunak yang dibuat. Berikut pengujian yang sudah dilakukan.

```

1 \begin{enumerate}
2 \item \texttt{OperasiMatematikaInterface}\\
3 Kelas Abstract OperasiMatematika.
4
5 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini
   adalah sebagai berikut.
6 \begin{itemize}
7 \item \texttt{public int calculate(int a, int b)}\\
8 Method untuk menghasilkan perhitungan 2 buah bilangan
9
10 \textbf{Parameter:}
11 \begin{itemize}
12 \item \texttt{int a} –
13 Bilangan pertama
14 \item \texttt{int b} –
15 Bilangan kedua
16 \end{itemize}
17 \textbf{Kembalian:} hasil perhitungan 2 buah bilangan
18
19 \textbf{Exception:} Tidak memiliki \textit{exception}
20
21 \end{itemize}
22 \item \texttt{Pembagian}\\
23 Kelas ini merupakan Kelas Pembagian.
24
25 Atribut yang dimiliki kelas ini adalah sebagai berikut.
26 \begin{itemize}
27 \item \texttt{int a} – Atribut A
28 \item \texttt{int b} – Atribut B
29 \end{itemize}
30 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
31 \begin{itemize}
32 \item \texttt{public int calculate(int a, int b)}\\
33 Method untuk menghasilkan perhitungan 2 buah bilangan
34
35 \textbf{Parameter:}
36 \begin{itemize}
37 \item \texttt{int a} –
38 Bilangan pertama
39 \item \texttt{int b} –

```



---

```

40 Bilangan kedua
41 \end{itemize}
42 \textbf{Kembalian}: hasil perhitungan 2 buah bilangan
43
44 \textbf{Exception}: Tidak memiliki \textit{exception}
45
46 \end{itemize}
47 \item \texttt{Pengurangan}\\
48 Kelas ini merupakan Kelas Pengurangan.
49
50 Atribut yang dimiliki kelas ini adalah sebagai berikut.
51 \begin{itemize}
52 \item \texttt{int a} – Atribut A
53 \item \texttt{int b} – Atribut B
54 \end{itemize}
55 \textit{Method–method} yang dimiliki kelas ini adalah sebagai berikut.
56 \begin{itemize}
57 \item \texttt{public int calculate(int a, int b)}\\
58 Method untuk menghasilkan perhitungan 2 buah bilangan
59
60 \textbf{Parameter}:
61 \begin{itemize}
62 \item \texttt{int a} –
63 Bilangan pertama
64 \item \texttt{int b} –
65 Bilangan kedua
66 \end{itemize}
67 \textbf{Kembalian}: hasil perhitungan 2 buah bilangan
68
69 \textbf{Exception}: Tidak memiliki \textit{exception}
70
71 \end{itemize}
72 \item \texttt{Perkalian}\\
73 Kelas ini merupakan Kelas Perkalian.
74
75 Atribut yang dimiliki kelas ini adalah sebagai berikut.
76 \begin{itemize}
77 \item \texttt{int a} – Atribut A
78 \item \texttt{int b} – Atribut B
79 \end{itemize}
80 \textit{Method–method} yang dimiliki kelas ini adalah sebagai berikut.
81 \begin{itemize}
82 \item \texttt{public int calculate(int a, int b)}\\
83 Method untuk menghasilkan perhitungan 2 buah bilangan
84
85 \textbf{Parameter}:
86 \begin{itemize}
87 \item \texttt{int a} –
88 Bilangan pertama
89 \item \texttt{int b} –
90 Bilangan kedua
91 \end{itemize}
92 \textbf{Kembalian}: hasil perhitungan 2 buah bilangan
93
94 \textbf{Exception}: Tidak memiliki \textit{exception}
95
96 \end{itemize}
97 \item \texttt{Pertambahan}\\
98 Kelas ini merupakan Kelas Pertambahan.
99
100 Atribut yang dimiliki kelas ini adalah sebagai berikut.
101 \begin{itemize}
102 \item \texttt{int a} – Atribut A

```

```

103 \item \texttt{int b} – Atribut B
104 \end{itemize}
105 \textit{Method–method} yang dimiliki kelas ini adalah sebagai berikut.
106 \begin{itemize}
107 \item \texttt{public int calculate(int a, int b)}\\
108 Method untuk menghasilkan perhitungan 2 buah bilangan
109
110 \textbf{Parameter:}
111 \begin{itemize}
112 \item \texttt{int a} –
113 Bilangan pertama
114 \item \texttt{int b} –
115 Bilangan kedua
116 \end{itemize}
117 \textbf{Kembalian:} hasil perhitungan 2 buah bilangan
118
119 \textbf{Exception:} Tidak memiliki \textit{exception}
120
121 \end{itemize}
122 \end{enumerate}

```

Listing 5.2: Hasil pengujian kode program sederhana

```

1 \begin{enumerate}
2 \item \texttt{AttributeClassExtractor}\\
3 Kelas ini merupakan kelas untuk mengambil informasi sebuah atribut yang
4 terdapat pada kelas.
5
6 Kelas ini tidak memiliki atribut. \textit{Method–method} yang dimiliki kelas ini
7 adalah sebagai berikut.
8 \begin{itemize}
9 \item \texttt{public static void extractAttributeClassContent(com.sun.javadoc.
10 FieldDoc[] fields, BufferedWriter out)}\\
11 \textit{Method} ini akan menampilkan atribut–atribut yang dimiliki oleh
12 sebuah kelas
13
14 \textbf{Parameter:}
15 \begin{itemize}
16 \item \texttt{com.sun.javadoc.FieldDoc[] fields} –
17 sebuah array berisikan sejumlah atribut dari kelas
18 \item \texttt{BufferedWriter out} –
19 turunan dari kelas \texttt{Writer} yang digunakan untuk menulis
20 file text
21 \end{itemize}
22 \textbf{Kembalian:} Tidak memiliki \textit{return value}
23
24 \textbf{Exception:} Tidak memiliki \textit{exception}
25
26 \end{itemize}
27 \item \texttt{ClassExtractor}\\
28 Kelas ini merupakan kelas untuk mengambil informasi dari sebuah kelas.
29
30 Kelas ini tidak memiliki atribut. \textit{Method–method} yang dimiliki kelas ini
31 adalah sebagai berikut.
32 \begin{itemize}
33 \item \texttt{public static void extractClassContent(com.sun.javadoc.ClassDoc[]
34 classes, BufferedWriter out)}\\
35 \textit{Method} ini akan menampilkan nama kelas berserta penjelasan dari sebuah
36 kelas

```

---

```

37 \item \texttt{BufferedWriter out} –
38 turunan dari kelas \texttt{Writer} yang digunakan untuk menulis file text
39 \end{itemize}
40 \textbf{Kembalian}: Tidak memiliki \textit{return value}
41
42 \textbf{Exception}: Tidak memiliki \textit{exception}
43
44 \end{itemize}
45 \item \texttt{Extractor}\\
46 Kelas ini merupakan kelas untuk menjalan \textit{custom doclet}.
47
48 Atribut yang dimiliki kelas ini adalah sebagai berikut.
49 \begin{itemize}
50 \item \texttt{String fileName} – atribut untuk nama \textit{file}
51 \end{itemize}
52 \textit{Method–method} yang dimiliki kelas ini adalah sebagai berikut.
53 \begin{itemize}
54 \item \texttt{public static boolean start(RootDoc root)}\\
55 \textit{Method} ini berperan sebagai \textit{method} untuk menjalankan
56 \textit{custom doclet}
57
58 \textbf{Parameter:}
59 \begin{itemize}
60 \item \texttt{RootDoc root} –
61 berperan sebagai mengambil seluruh informasi spesifik dari
62 \textit{option} yang terdapat pada \textit{command–line} sebuah
63 \textit{terminal}. Selain itu berperan juga untuk mengambil informasi dari
64 sekumpulan \textit{file java} yang akan di proses.
65 \end{itemize}
66 \textbf{Kembalian}: kondisi true
67
68 \textbf{Exception}: Tidak memiliki \textit{exception}
69
70 \item \texttt{private static void init(com.sun.javadoc.ClassDoc[] classes)}\\
71 \textit{Method} ini berperan untuk menulis kedalam sebuah \textit{file}
72 saat \textit{javadoc} berjalan.
73
74 \textbf{Parameter:}
75 \begin{itemize}
76 \item \texttt{com.sun.javadoc.ClassDoc[] classes} –
77 sebuah array yang berisikan sekumpulan \textit{file java}
78 yang akan di proses.
79 \end{itemize}
80 \textbf{Kembalian}: Tidak memiliki \textit{return value}
81
82 \textbf{Exception}: Tidak memiliki \textit{exception}
83
84 \item \texttt{public static int optionLength(String option)}\\
85 Method untuk menghitung banyak option yang digunakan pada
86 \textit{command–line}
87
88 \textbf{Parameter:}
89 \begin{itemize}
90 \item \texttt{String option} –
91 sebuah option
92 \end{itemize}
93 \textbf{Kembalian}: panjang setiap option
94
95 \textbf{Exception}: Tidak memiliki \textit{exception}
96
97 \item \texttt{public static boolean validOptions(java.lang.String[][] args,
98 DocErrorReporter err)}\\
99 Pengecekan option valid

```

```

99
100 \textbf{Parameter:}
101 \begin{itemize}
102 \item \texttt{java.lang.String [][] args} –
103 String array 2 dimensi dari option
104 \item \texttt{DocErrorReporter err} –
105 sebuah error jika tidak terdapat option tersebut.
106 \end{itemize}
107 \textbf{Kembalian}: bernilai true jika option tersebut dikenali, false jika option
108 tersebut tidak dikenali
109
110 \textbf{Exception}: Tidak memiliki \textit{exception}
111
112 \end{itemize}
113 \item \texttt{MethodClassExtractor}\\
114 Kelas ini merupakan kelas untuk mengambil informasi sebuah \textit{method}
115 terdapat pada kelas.
116
117 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini
    adalah sebagai berikut.
118 \begin{itemize}
119 \item \texttt{public static void extractMethodContent(ClassDoc superclass, com.sun.
    javadoc.MethodDoc[] methods, BufferedWriter out)}\\
120 \textit{Method} ini akan menampilkan \textit{method-method} yang dimiliki
121 oleh sebuah kelas
122
123 \textbf{Parameter:}
124 \begin{itemize}
125 \item \texttt{ClassDoc superclass} –
126 sebuah objek ClassDoc
127 \item \texttt{com.sun.javadoc.MethodDoc[] methods} –
128 sebuah array berisikan sejumlah \textit{method} dari kelas
129 \item \texttt{BufferedWriter out} –
130 turunan dari kelas \texttt{Writer} yang digunakan untuk menulis
131 file text
132 \end{itemize}
133 \textbf{Kembalian}: Tidak memiliki \textit{return value}
134
135 \textbf{Exception}: Tidak memiliki \textit{exception}
136
137 \end{itemize}
138 \end{enumerate}

```

Listing 5.3: Hasil Pengujian kode program perangkat punak

### 5.2.3 Pengujian Eksperimental

Pengujian eksperimental dilakukan terhadap kode program SIAModels. Hasil pengujian terdapat pada lampiran [B](#)

## BAB 6

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Dari hasil pembangunan perangkat lunak Konversi Javadoc ke  $\text{\LaTeX}$  , didapatkanlah kesimpulan-kesimpulan sebagai berikut:

1. Telah berhasil mengimplementasikan *library* Javadoc Doclet API untuk mengekstraksi informasi dari sekumpulan *file java*.
2. Telah berhasil mengambil informasi dari sekumpulan *file java* dan menghasilkan sebuah *file \LaTeX* yang berisi dokumentasi *javadoc*.

#### 6.2 Saran

Dari hasil penelitian termasuk kesimpulan yang didapat, berikut adalah beberapa saran untuk pengembangan.

1. Dalam pengembangan berikutnya, perlu diperhatikan agar hasil dokumentasi dalam terurut berdasarkan hirarki.
2. Pada *javadoc*, terdapat 2 tipe *override* yaitu *Override By* dan *Specified By*. Pada saat ini, perangkat lunak hanya dapat mengekstraksi sampai *Override By*. Jika memungkinkan, sebaiknya dapat mengekstraksi sampai *Specified By*.



## DAFTAR REFERENSI

- [1] Oracle (1993) javadoc - the java api documentation generator. <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html>. 27 September 2017.
- [2] Oracle (1993) Javadoc doclet api. <https://docs.oracle.com/javase/8/docs/jdk/api/javadoc/doclet/index.html>. 5 oktober 2017.
- [3] Lamport, L. (1994) *LaTeX: A Document Preparation System*, 2 edition.





# LAMPIRAN A

## KODE PROGRAM

Listing A.1: OperasiMatematikaInterface.java

```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package javadoc;
7
8  /**
9   * Kelas Abstract OperasiMatematika. Kelas ini memiliki method {@link #calculate(int, int)}
10   *
11   * @author abathz
12   */
13  public abstract class OperasiMatematikaInterface {
14
15      /**
16       * Method untuk menghasilkan perhitungan 2 buah bilangan
17       * @param a Bilangan pertama
18       * @param b Bilangan kedua
19       * @return hasil perhitungan 2 buah bilangan {@link {@link Integer#NaN}}
20       */
21      public int calculate(int a, int b){return 0;}
22  }
```

Listing A.2: Pembagian.java

```
1  package javadoc;
2
3  /**
4   * Kelas ini merupakan Kelas Pembagian
5   *
6   * @author Adli Fariz Bonaputra
7   * @see "Pembagian"
8   */
9  public class Pembagian extends OperasiMatematikaInterface {
10
11      /**
12       * Atribut A
13       */
14      private int a;
15      /**
16       * Atribut B
17       */
18      private int b;
19
20      @Override
21      public int calculate(int a, int b) {
22          int hasil = a / b;
23          return hasil;
24      }
25
26  }
```

Listing A.3: Pengurangan.java

```
1  package javadoc;
2
3  /**
4   * Kelas ini merupakan Kelas Pengurangan
5   *
6   * @author Adli Fariz Bonaputra
7   * @see "Pengurangan"
8   */
9  public class Pengurangan extends OperasiMatematikaInterface {
10
11      /**
12       * Atribut A
13       */
14      private int a;
15      /**
16       * Atribut B
17       */
18      private int b;
```

```

18 private int b;
19
20 @Override
21 public int calculate(int a, int b) {
22     int hasil = a - b;
23     return hasil;
24 }
25 }

```

Listing A.4: Perkalian.java

```

1 package javadoc;
2
3 /**
4  * Kelas ini merupakan Kelas Perkalian
5  *
6  * @author Adli Fariz Bonaputra
7  * @see "Perkalian"
8  */
9
10 public class Perkalian extends OperasiMatematikaInterface {
11
12     /**
13      * Atribut A
14      */
15     private int a;
16
17     /**
18      * Atribut B
19      */
20     private int b;
21
22     @Override
23     public int calculate(int a, int b) {
24         int hasil = a * b;
25         return hasil;
26     }
27 }

```

Listing A.5: Pertambahan.java

```

1 package javadoc;
2
3 /**
4  * Kelas ini merupakan Kelas Pertambahan
5  *
6  * @author Adli Fariz Bonaputra
7  * @see "Pertambahan"
8  */
9
10 public class Pertambahan extends OperasiMatematikaInterface {
11
12     /**
13      * Atribut A
14      */
15     private int a;
16
17     /**
18      * Atribut B
19      */
20     private int b;
21
22     @Override
23     public int calculate(int a, int b) {
24         int hasil = a + b;
25         return hasil;
26     }
27 }

```

Listing A.6: operasimatematika.tex

```

1 \documentclass{article}
2 \begin{document}
3 \begin{enumerate}
4 \item \texttt{OperasiMatematikaInterface}
5
6 Kelas Abstract OperasiMatematika. Kelas ini memiliki method \texttt{calculate(int, int)}
7
8 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
9 \begin{itemize}
10 \item \texttt{public int calculate(int a, int b)} Method untuk menghasilkan perhitungan 2 buah bilangan
11
12 \textbf{Parameter:}
13 \begin{itemize}
14 \item \texttt{int a} -
15 Bilangan pertama
16 \item \texttt{int b} -
17 Bilangan kedua
18 \end{itemize}
19 \textbf{Return Value:} hasil perhitungan 2 buah bilangan DoubleNaN
20
21 \textbf{Exception:} Tidak memiliki \textit{exception}
22
23 \end{itemize}
24 \item \texttt{Pembagian}
25

```

```

26 | Kelas ini merupakan Kelas Pembagian
27 |
28 | Atribut yang dimiliki kelas ini adalah sebagai berikut.
29 | \begin{itemize}
30 | \item \texttt{int a} - Atribut A
31 | \item \texttt{int b} - Atribut B
32 | \end{itemize}
33 | \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
34 | \begin{itemize}
35 | \item \texttt{public int calculate(int a, int b)}Method untuk menghasilkan perhitungan 2 buah bilangan
36 |
37 | \textbf{Parameter:}
38 | \begin{itemize}
39 | \item \texttt{int a} -
40 | Bilangan pertama
41 | \item \texttt{int b} -
42 | Bilangan kedua
43 | \end{itemize}
44 | \textbf{Return Value:} hasil perhitungan 2 buah bilangan DoubleNaN
45 |
46 | \textbf{Exception:} Tidak memiliki \textit{exception}
47 |
48 | \end{itemize}
49 | \item \texttt{Pengurangan}
50 |
51 | Kelas ini merupakan Kelas Pengurangan
52 |
53 | Atribut yang dimiliki kelas ini adalah sebagai berikut.
54 | \begin{itemize}
55 | \item \texttt{int a} - Atribut A
56 | \item \texttt{int b} - Atribut B
57 | \end{itemize}
58 | \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
59 | \begin{itemize}
60 | \item \texttt{public int calculate(int a, int b)}Method untuk menghasilkan perhitungan 2 buah bilangan
61 |
62 | \textbf{Parameter:}
63 | \begin{itemize}
64 | \item \texttt{int a} -
65 | Bilangan pertama
66 | \item \texttt{int b} -
67 | Bilangan kedua
68 | \end{itemize}
69 | \textbf{Return Value:} hasil perhitungan 2 buah bilangan DoubleNaN
70 |
71 | \textbf{Exception:} Tidak memiliki \textit{exception}
72 |
73 | \end{itemize}
74 | \item \texttt{Perkalian}
75 |
76 | Kelas ini merupakan Kelas Perkalian
77 |
78 | Atribut yang dimiliki kelas ini adalah sebagai berikut.
79 | \begin{itemize}
80 | \item \texttt{int a} - Atribut A
81 | \item \texttt{int b} - Atribut B
82 | \end{itemize}
83 | \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
84 | \begin{itemize}
85 | \item \texttt{public int calculate(int a, int b)}Method untuk menghasilkan perhitungan 2 buah bilangan
86 |
87 | \textbf{Parameter:}
88 | \begin{itemize}
89 | \item \texttt{int a} -
90 | Bilangan pertama
91 | \item \texttt{int b} -
92 | Bilangan kedua
93 | \end{itemize}
94 | \textbf{Return Value:} hasil perhitungan 2 buah bilangan DoubleNaN
95 |
96 | \textbf{Exception:} Tidak memiliki \textit{exception}
97 |
98 | \end{itemize}
99 | \item \texttt{Pertambahan}
100 |
101 | Kelas ini merupakan Kelas Pertambahan
102 |
103 | Atribut yang dimiliki kelas ini adalah sebagai berikut.
104 | \begin{itemize}
105 | \item \texttt{int a} - Atribut A
106 | \item \texttt{int b} - Atribut B
107 | \end{itemize}
108 | \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
109 | \begin{itemize}
110 | \item \texttt{public int calculate(int a, int b)}Method untuk menghasilkan perhitungan 2 buah bilangan
111 |
112 | \textbf{Parameter:}
113 | \begin{itemize}
114 | \item \texttt{int a} -
115 | Bilangan pertama
116 | \item \texttt{int b} -
117 | Bilangan kedua
118 | \end{itemize}
119 | \textbf{Return Value:} hasil perhitungan 2 buah bilangan DoubleNaN
120 |
121 | \textbf{Exception:} Tidak memiliki \textit{exception}
122 |
123 | \end{itemize}
124 | \end{enumerate}

```



### 1. OperasiMatematikaInterface

Kelas Abstract OperasiMatematika. Kelas ini memiliki method `calculate(int, int)`

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public int calculate(int a, int b)` Method untuk menghasilkan perhitungan 2 buah bilangan

**Parameter:**

- `int a` - Bilangan pertama
- `int b` - Bilangan kedua

**Return Value:** hasil perhitungan 2 buah bilangan DoubleNaN

**Exception:** Tidak memiliki *exception*

### 2. Pembagian

Kelas ini merupakan Kelas Pembagian

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- `int a` - Atribut A
- `int b` - Atribut B

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public int calculate(int a, int b)` Method untuk menghasilkan perhitungan 2 buah bilangan

**Parameter:**

- `int a` - Bilangan pertama
- `int b` - Bilangan kedua

**Return Value:** hasil perhitungan 2 buah bilangan DoubleNaN

**Exception:** Tidak memiliki *exception*

### 3. Pengurangan

Kelas ini merupakan Kelas Pengurangan

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- `int a` - Atribut A
- `int b` - Atribut B

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public int calculate(int a, int b)` Method untuk menghasilkan perhitungan 2 buah bilangan

**Parameter:**

- `int a` - Bilangan pertama

– `int b` - Bilangan kedua

**Return Value:** hasil perhitungan 2 buah bilangan DoubleNaN

**Exception:** Tidak memiliki *exception*

#### 4. Perkalian

Kelas ini merupakan Kelas Perkalian

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- `int a` - Atribut A
- `int b` - Atribut B

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public int calculate(int a, int b)` Method untuk menghasilkan perhitungan 2 buah bilangan

**Parameter:**

- `int a` - Bilangan pertama
- `int b` - Bilangan kedua

**Return Value:** hasil perhitungan 2 buah bilangan DoubleNaN

**Exception:** Tidak memiliki *exception*

#### 5. Pertambahan

Kelas ini merupakan Kelas Pertambahan

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- `int a` - Atribut A
- `int b` - Atribut B

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public int calculate(int a, int b)` Method untuk menghasilkan perhitungan 2 buah bilangan

**Parameter:**

- `int a` - Bilangan pertama
- `int b` - Bilangan kedua

**Return Value:** hasil perhitungan 2 buah bilangan DoubleNaN

**Exception:** Tidak memiliki *exception*

# LAMPIRAN B

## HASIL PENGUJIAN

Listing B.1: siamodels.tex

```
1 \documentclass{article}
2 \begin{document}
3 \begin{enumerate}
4 \item \texttt{InfoMataKuliah}
5
6
7
8 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
9 \begin{itemize}
10 \item \texttt{public int sks()}\Jumlah bobot sks dari mata kuliah ini
11
12 \textbf{Parameter:}
13 \begin{itemize}
14 \item Tidak memiliki parameter \textit{method}
15 \end{itemize}
16 \textbf{Return Value:} jumlah bobot sks
17
18 \textbf{Exception:} Tidak memiliki \textit{exception}
19
20 \item \texttt{public String nama()}\Nama mata kuliah ini
21
22 \textbf{Parameter:}
23 \begin{itemize}
24 \item Tidak memiliki parameter \textit{method}
25 \end{itemize}
26 \textbf{Return Value:} nama mata kuliah
27
28 \textbf{Exception:} Tidak memiliki \textit{exception}
29
30 \end{itemize}
31 \item \texttt{MataKuliahFactory}
32
33 Kelas yang bertugas membuat kelas mata kuliah, dan menyimpannya untuk bisa
34 digunakan kemudian (untuk hemat memori).
35
36 Atribut yang dimiliki kelas ini adalah sebagai berikut.
37 \begin{itemize}
38 \item \texttt{String DEFAULT\_MATAKULIAH\_PACKAGE} - Lokasi package untuk daftar mata kuliah
39 \item \texttt{MataKuliahFactory instance} - Singleton instance to factory.
40 \item \texttt{SortedMap mataKuliahCache} - Singleton instances untuk mata kuliah.
41 \end{itemize}
42 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
43 \begin{itemize}
44 \item \texttt{public static MataKuliahFactory getInstance()}\
45
46 \textbf{Parameter:}
47 \begin{itemize}
48 \item Tidak memiliki parameter \textit{method}
49 \end{itemize}
50 \textbf{Return Value:} Tidak memiliki \textit{return value}
51
52 \textbf{Exception:} Tidak memiliki \textit{exception}
53
54 \item \texttt{public MataKuliah createMataKuliah(java.lang.String kode, int sks, java.lang.String nama)}\Membuat baru atau
55 mendapatkan mata kuliah, jika memiliki informasi
56 nama dan jumlah SKS.
57
58 \textbf{Parameter:}
59 \begin{itemize}
60 \item \texttt{String kode} -
61 kode mata kuliah
62 \item \texttt{int sks} -
63 jumlah SKS
64 \item \texttt{String nama} -
65 nama mata kuliah
66 \end{itemize}
67 \textbf{Return Value:} objek mata kuliah
68
69 \textbf{Exception:} Tidak memiliki \textit{exception}
70
71 \item \texttt{public MataKuliah createMataKuliah(java.lang.String kode)}\Membuat baru atau mendapatkan mata kuliah, jika tidak
72 memiliki informasi
73 nama dan jumlah SKS.
74
75 \textbf{Parameter:}
76 \begin{itemize}
77 \item \texttt{String kode} -
78 kode mata kuliah
79 \end{itemize}
80 \textbf{Return Value:} objek mata kuliah
81
82 \textbf{Exception:} IllegalStateException
83 jika sks dan tidak sesuai dengan yang ada di kode
```

```

94 \item \texttt{Semester GENAP} -
95 \item \texttt{int order} -
96 \end{itemize}
97 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
98 \begin{itemize}
99 \item \texttt{public static Semester values()}
100
101 \textbf{Parameter:}
102 \begin{itemize}
103 \item Tidak memiliki parameter \textit{method}
104 \end{itemize}
105 \textbf{Return Value:} Tidak memiliki \textit{return value}
106
107 \textbf{Exception:} Tidak memiliki \textit{exception}
108
109 \item \texttt{public static Semester valueOf(java.lang.String name)}
110
111 \textbf{Parameter:}
112 \begin{itemize}
113 \item \texttt{String name} -
114 \end{itemize}
115 \textbf{Return Value:} Tidak memiliki \textit{return value}
116
117 \textbf{Exception:} Tidak memiliki \textit{exception}
118
119 \item \texttt{public static Semester fromString(java.lang.String text)}
120
121 \textbf{Parameter:}
122 \begin{itemize}
123 \item \texttt{String text} -
124 \end{itemize}
125 \textbf{Return Value:} Tidak memiliki \textit{return value}
126
127 \textbf{Exception:} Tidak memiliki \textit{exception}
128
129 \item \texttt{int getOrder()}
130
131 \textbf{Parameter:}
132 \begin{itemize}
133 \item Tidak memiliki parameter \textit{method}
134 \end{itemize}
135 \textbf{Return Value:} Tidak memiliki \textit{return value}
136
137 \textbf{Exception:} Tidak memiliki \textit{exception}
138
139 \end{itemize}
140 \item \texttt{TahunSemester}
141
142 Menyimpan konstanta untuk semester beserta tahunnya di UNPAR.
143
144 Atribut yang dimiliki kelas ini adalah sebagai berikut.
145 \begin{itemize}
146 \item \texttt{String kodeTahunSemester} - Kode semester 3 digit, sesuai DPS:
147 <ul>
148 <li>2 digit pertama berupa tahun, 2 digit terakhir</li>
149 <li>digit terakhir: 1 untuk ganjil, 2 untuk genap, 4 untuk pendek.
150 </ul>
151 \end{itemize}
152 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
153 \begin{itemize}
154 \item \texttt{public Semester getSemester()}
155
156 \textbf{Parameter:}
157 \begin{itemize}
158 \item Tidak memiliki parameter \textit{method}
159 \end{itemize}
160 \textbf{Return Value:} Tidak memiliki \textit{return value}
161
162 \textbf{Exception:} Tidak memiliki \textit{exception}
163
164 \item \texttt{public int getTahun()}
165
166 \textbf{Parameter:}
167 \begin{itemize}
168 \item Tidak memiliki parameter \textit{method}
169 \end{itemize}
170 \textbf{Return Value:} Tidak memiliki \textit{return value}
171
172 \textbf{Exception:} Tidak memiliki \textit{exception}
173
174 \item \texttt{private static void validateKodeSemester(java.lang.String kodeTahunSemester)}
175
176 \textbf{Parameter:}
177 \begin{itemize}
178 \item \texttt{String kodeTahunSemester} -
179 \end{itemize}
180 \textbf{Return Value:} Tidak memiliki \textit{return value}
181
182 \textbf{Exception:} Tidak memiliki \textit{exception}
183
184 \item \texttt{public String getKodeDPS()}\textit{Mendapatkan kode tahun/semester sesuai aturan di DPS.}
185
186 \textbf{Parameter:}
187 \begin{itemize}
188 \item Tidak memiliki parameter \textit{method}
189 \end{itemize}
190 \textbf{Return Value:} kode tahun/semester sesuai aturan di DPS.
191
192 \textbf{Exception:} Tidak memiliki \textit{exception}

```



```

193 | \item \texttt{public int compareTo(id.ac.unpar.siamodels.TahunSemester o)}
194 |
195 | \textbf{Parameter:}
196 | \begin{itemize}
197 | \item \texttt{TahunSemester o} -
198 | \end{itemize}
199 | \textbf{Return Value:} Tidak memiliki \textit{return value}
200 |
201 | \textbf{Exception:} Tidak memiliki \textit{exception}
202 |
203 | \textbf{Override:} \texttt{compareTo} dari kelas \texttt{Object}
204 |
205 | \item \texttt{public boolean equals(java.lang.Object arg0)}
206 |
207 | \textbf{Parameter:}
208 | \begin{itemize}
209 | \item \texttt{Object arg0} -
210 | \end{itemize}
211 | \textbf{Return Value:} Tidak memiliki \textit{return value}
212 |
213 | \textbf{Exception:} Tidak memiliki \textit{exception}
214 |
215 | \item \texttt{public String toString()}
216 |
217 | \textbf{Parameter:}
218 | \begin{itemize}
219 | \item Tidak memiliki parameter \textit{method}
220 | \end{itemize}
221 | \textbf{Return Value:} Tidak memiliki \textit{return value}
222 |
223 | \textbf{Exception:} Tidak memiliki \textit{exception}
224 |
225 | \end{itemize}
226 | \item \texttt{Mahasiswa}
227 |
228 |
229 |
230 |
231 | Atribut yang dimiliki kelas ini adalah sebagai berikut.
232 | \begin{itemize}
233 | \item \texttt{String npm} -
234 | \item \texttt{String nama} -
235 | \item \texttt{List riwayatNilai} -
236 | \item \texttt{URL photoURL} -
237 | \item \texttt{List jadwalKuliahList} -
238 | \item \texttt{SortedMap nilaiTOEFL} -
239 | \end{itemize}
240 | \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
241 | \begin{itemize}
242 | \item \texttt{public String getName()}
243 |
244 | \textbf{Parameter:}
245 | \begin{itemize}
246 | \item Tidak memiliki parameter \textit{method}
247 | \end{itemize}
248 | \textbf{Return Value:} Tidak memiliki \textit{return value}
249 |
250 | \textbf{Exception:} Tidak memiliki \textit{exception}
251 |
252 | \item \texttt{public void setName(java.lang.String nama)}
253 |
254 | \textbf{Parameter:}
255 | \begin{itemize}
256 | \item \texttt{String nama} -
257 | \end{itemize}
258 | \textbf{Return Value:} Tidak memiliki \textit{return value}
259 |
260 | \textbf{Exception:} Tidak memiliki \textit{exception}
261 |
262 | \item \texttt{public String getNpm()}
263 |
264 | \textbf{Parameter:}
265 | \begin{itemize}
266 | \item Tidak memiliki parameter \textit{method}
267 | \end{itemize}
268 | \textbf{Return Value:} Tidak memiliki \textit{return value}
269 |
270 | \textbf{Exception:} Tidak memiliki \textit{exception}
271 |
272 | \item \texttt{public URL getPhotoURL()}
273 |
274 | \textbf{Parameter:}
275 | \begin{itemize}
276 | \item Tidak memiliki parameter \textit{method}
277 | \end{itemize}
278 | \textbf{Return Value:} Tidak memiliki \textit{return value}
279 |
280 | \textbf{Exception:} Tidak memiliki \textit{exception}
281 |
282 | \item \texttt{public void setPhotoURL(java.net.URL photoURL)}
283 |
284 | \textbf{Parameter:}
285 | \begin{itemize}
286 | \item \texttt{URL photoURL} -
287 | \end{itemize}
288 | \textbf{Return Value:} Tidak memiliki \textit{return value}
289 |
290 | \textbf{Exception:} Tidak memiliki \textit{exception}
291 |

```

```

292 \item \texttt{public List getJadwalKuliahList()}
293
294 \textbf{Parameter:}
295 \begin{itemize}
296 \item Tidak memiliki parameter \textit{method}
297 \end{itemize}
298 \textbf{Return Value:} Tidak memiliki \textit{return value}
299
300 \textbf{Exception:} Tidak memiliki \textit{exception}
301
302 \item \texttt{public void setJadwalKuliahList(java.util.List jadwalKuliahList)}
303
304 \textbf{Parameter:}
305 \begin{itemize}
306 \item \texttt{java.util.List jadwalKuliahList} -
307 \end{itemize}
308 \textbf{Return Value:} Tidak memiliki \textit{return value}
309
310 \textbf{Exception:} Tidak memiliki \textit{exception}
311
312 \item \texttt{public String getEmailAddress()}
313
314 \textbf{Parameter:}
315 \begin{itemize}
316 \item Tidak memiliki parameter \textit{method}
317 \end{itemize}
318 \textbf{Return Value:} Tidak memiliki \textit{return value}
319
320 \textbf{Exception:} Tidak memiliki \textit{exception}
321
322 \item \texttt{public List getRiwayatNilai()}
323
324 \textbf{Parameter:}
325 \begin{itemize}
326 \item Tidak memiliki parameter \textit{method}
327 \end{itemize}
328 \textbf{Return Value:} Tidak memiliki \textit{return value}
329
330 \textbf{Exception:} Tidak memiliki \textit{exception}
331
332 \item \texttt{public SortedMap getNilaiTOEFL()}
333
334 \textbf{Parameter:}
335 \begin{itemize}
336 \item Tidak memiliki parameter \textit{method}
337 \end{itemize}
338 \textbf{Return Value:} Tidak memiliki \textit{return value}
339
340 \textbf{Exception:} Tidak memiliki \textit{exception}
341
342 \item \texttt{public void setNilaiTOEFL(java.util.SortedMap nilaiTOEFL)}
343
344 \textbf{Parameter:}
345 \begin{itemize}
346 \item \texttt{java.util.SortedMap nilaiTOEFL} -
347 \end{itemize}
348 \textbf{Return Value:} Tidak memiliki \textit{return value}
349
350 \textbf{Exception:} Tidak memiliki \textit{exception}
351
352 \item \texttt{public double calculateIPKLulus()}Menghitung IPK mahasiswa sampai saat ini, dengan aturan:
353 <ul>
354 <li>Kuliah yang tidak lulus tidak dihitung
355 <li>Jika pengambilan beberapa kali, diambil <em>nilai terbaik</em>.
356 </ul>
357 Sebelum memanggil method ini, \texttt{getRiwayatNilai()} harus sudah mengandung nilai per mata kuliah!
358
359 \textbf{Parameter:}
360 \begin{itemize}
361 \item Tidak memiliki parameter \textit{method}
362 \end{itemize}
363 \textbf{Return Value:} IPK Lulus, atau DoubleNaN jika belum mengambil satu kuliah pun.
364
365 \textbf{Exception:} Tidak memiliki \textit{exception}
366
367 \item \texttt{public double calculateIPLulus()}Menghitung IP mahasiswa sampai saat ini, dengan aturan:
368 <ul>
369 <li>Kuliah yang tidak lulus tidak dihitung
370 <li>Jika pengambilan beberapa kali, diambil <em>nilai terbaik</em>.
371 </ul>
372 Sebelum memanggil method ini, \texttt{getRiwayatNilai()} harus sudah mengandung nilai per mata kuliah!
373
374 \textbf{Parameter:}
375 \begin{itemize}
376 \item Tidak memiliki parameter \textit{method}
377 \end{itemize}
378 \textbf{Return Value:} IPK Lulus, atau DoubleNaN jika belum mengambil satu kuliah pun.
379
380 \textbf{Exception:} Tidak memiliki \textit{exception}
381
382 \item \texttt{public double calculateIPTempuh(boolean lulusSaja)}Menghitung IP mahasiswa sampai saat ini, dengan aturan:
383 <ul>
384 <li>Perhitungan kuliah yang tidak lulus ditentukan parameter
385 <li>Jika pengambilan beberapa kali, diambil <em>nilai terbaik</em>.
386 </ul>
387
388 \textbf{Parameter:}
389 \begin{itemize}
390 \item \texttt{boolean lulusSaja} -

```

```

391 set true jika ingin membuang mata kuliah tidak lulus, false jika ingin semua (sama dengan "IP N. Terbaik" di DPS)
392 Sebelum memanggil method ini, \texttt{getRiwayatNilai()} harus sudah mengandung nilai per mata kuliah!
393 \end{itemize}
394 \textbf{Return Value}: IPK Lulus, atau DoubleNaN jika belum mengambil satu kuliah pun.
395
396 \textbf{Exception}: Tidak memiliki \textit{exception}
397
398 \item \texttt{public double calculateIPKumulatif()}Menghitung IP Kumulatif mahasiswa sampai saat ini, dengan aturan:
399 <ul>
400 <li>Jika pengambilan beberapa kali, diambil semua.
401 </li>
402 </ul>
403 Sebelum memanggil method ini, \texttt{getRiwayatNilai()} harus sudah mengandung nilai per mata kuliah!
404 \textbf{Parameter}:
405 \begin{itemize}
406 \item Tidak memiliki parameter \textit{method}
407 \end{itemize}
408 \textbf{Return Value}: IPK Lulus, atau DoubleNaN jika belum mengambil satu kuliah pun.
409
410 \textbf{Exception}: Tidak memiliki \textit{exception}
411
412 \item \texttt{public double calculateIPKTempuh(boolean lulusSaja)}Menghitung IPK mahasiswa sampai saat ini, dengan aturan:
413 <ul>
414 <li>Perhitungan kuliah yang tidak lulus ditentukan parameter
415 <li>Jika pengambilan beberapa kali, diambil <em>nilai terbaik</em>.
416 </li>
417 </ul>
418 \textbf{Parameter}:
419 \begin{itemize}
420 \item \texttt{boolean lulusSaja} -
421 set true jika ingin membuang mata kuliah tidak lulus
422 Sebelum memanggil method ini, \texttt{getRiwayatNilai()} harus sudah mengandung nilai per mata kuliah!
423 \end{itemize}
424 \textbf{Return Value}: IPK Lulus, atau DoubleNaN jika belum mengambil satu kuliah pun.
425
426 \textbf{Exception}: Tidak memiliki \textit{exception}
427
428 \item \texttt{public double calculateIPS()}Menghitung IPS semester terakhir sampai saat ini, dengan aturan:
429 <ul>
430 <li>Kuliah yang tidak lulus <em>dihitung</em>.
431 </li>
432 </ul>
433 Sebelum memanggil method ini, \texttt{getRiwayatNilai()} harus sudah mengandung nilai per mata kuliah!
434 \textbf{Parameter}:
435 \begin{itemize}
436 \item Tidak memiliki parameter \textit{method}
437 \end{itemize}
438 \textbf{Return Value}: nilai IPS sampai saat ini
439
440 \textbf{Exception}: ArrayIndexOutOfBoundsException jika belum ada nilai satupun
441
442 \item \texttt{public int calculateSKSLulus()}Menghitung jumlah SKS lulus mahasiswa saat ini.
443 Sebelum memanggil method ini, \texttt{getRiwayatNilai()} harus sudah mengandung nilai per mata kuliah!
444
445 \textbf{Parameter}:
446 \begin{itemize}
447 \item Tidak memiliki parameter \textit{method}
448 \end{itemize}
449 \textbf{Return Value}: SKS Lulus
450
451 \textbf{Exception}: Tidak memiliki \textit{exception}
452
453 \item \texttt{public int calculateSKSTempuh(boolean lulusSaja)}Menghitung jumlah SKS tempuh mahasiswa saat ini.
454 Sebelum memanggil method ini, \texttt{getRiwayatNilai()} harus sudah mengandung nilai per mata kuliah!
455
456 \textbf{Parameter}:
457 \begin{itemize}
458 \item \texttt{boolean lulusSaja} -
459 set true jika ingin membuang SKS tidak lulus
460 \end{itemize}
461 \textbf{Return Value}: SKS tempuh
462
463 \textbf{Exception}: Tidak memiliki \textit{exception}
464
465 \item \texttt{public Set calculateTahunSemesterAktif()}Mendapatkan seluruh tahun semester di mana mahasiswa ini tercatat
466 sebagai mahasiswa aktif, dengan strategi memeriksa riwayat nilainya.
467 Jika ada satu nilai saja pada sebuah tahun semester, maka dianggap
468 aktif pada semester tersebut.
469
470 \textbf{Parameter}:
471 \begin{itemize}
472 \item Tidak memiliki parameter \textit{method}
473 \end{itemize}
474 \textbf{Return Value}: kumpulan tahun semester di mana mahasiswa ini aktif
475
476 \textbf{Exception}: Tidak memiliki \textit{exception}
477
478 \item \texttt{public boolean hasLulusKuliah(java.lang.String kodeMataKuliah)}Memeriksa apakah mahasiswa ini sudah lulus mata
479 kuliah tertentu. Kompleksitas O(n).
480 Sebelum memanggil method ini, \texttt{getRiwayatNilai()} harus sudah mengandung nilai per mata kuliah!
481 Note: jika yang dimiliki adalah MataKuliah, gunakanlah MataKuliah.getKode().
482
483 \textbf{Parameter}:
484 \begin{itemize}
485 \item \texttt{String kodeMataKuliah} -
486 kode mata kuliah yang ingin diperiksa kelulusannya.
487 \end{itemize}
488 \textbf{Return Value}: true jika sudah pernah mengambil dan lulus, false jika belum

```

```

489 \textbf{Exception}: Tidak memiliki \textit{exception}
490
491 \item \texttt{public boolean hasTempuhKuliah(java.lang.String kodeMataKuliah)}Memeriksa apakah mahasiswa ini sudah pernah menempuh
    mata kuliah tertentu. Kompleksitas  $O(n)$ .
492 Sebelum memanggil method ini, \texttt{getRiwayatNilai()} harus sudah ada isinya!
493 Note: jika yang dimiliki adalah MataKuliah, gunakanlah MataKuliah.getKode().
494
495 \textbf{Parameter:}
496 \begin{itemize}
497 \item \texttt{String kodeMataKuliah} -
498 kode mata kuliah yang ingin diperiksa.
499 \end{itemize}
500 \textbf{Return Value}: true jika sudah pernah mengambil, false jika belum
501
502 \textbf{Exception}: Tidak memiliki \textit{exception}
503
504 \item \texttt{public int getTahunAngkatan()}Mendapatkan tahun angkatan mahasiswa ini, berdasarkan NPM nya
505
506 \textbf{Parameter:}
507 \begin{itemize}
508 \item Tidak memiliki parameter \textit{method}
509 \end{itemize}
510 \textbf{Return Value}: tahun angkatan
511
512 \textbf{Exception}: Tidak memiliki \textit{exception}
513
514 \item \texttt{public String toString()}
515
516 \textbf{Parameter:}
517 \begin{itemize}
518 \item Tidak memiliki parameter \textit{method}
519 \end{itemize}
520 \textbf{Return Value}: Tidak memiliki \textit{return value}
521
522 \textbf{Exception}: Tidak memiliki \textit{exception}
523
524 \end{itemize}
525 \item \texttt{Mahasiswa.Nilai}
526
527 Merepresentasikan nilai yang ada di riwayat nilai mahasiswa
528
529 Atribut yang dimiliki kelas ini adalah sebagai berikut.
530 \begin{itemize}
531 \item \texttt{TahunSemester tahunSemester} - Tahun dan Semester kuliah ini diambil
532 \item \texttt{MataKuliah mataKuliah} - Mata kuliah yang diambil
533 \item \texttt{Character kelas} - Kelas kuliah
534 \item \texttt{Double nilaiART} - Nilai ART
535 \item \texttt{Double nilaiUTS} - Nilai UTS
536 \item \texttt{Double nilaiUAS} - Nilai UAS
537 \item \texttt{Character nilaiAkhir} - Nilai Akhir
538 \end{itemize}
539 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
540 \begin{itemize}
541 \item \texttt{public MataKuliah getMataKuliah()}
542
543 \textbf{Parameter:}
544 \begin{itemize}
545 \item Tidak memiliki parameter \textit{method}
546 \end{itemize}
547 \textbf{Return Value}: Tidak memiliki \textit{return value}
548
549 \textbf{Exception}: Tidak memiliki \textit{exception}
550
551 \item \texttt{public Character getKelas()}
552
553 \textbf{Parameter:}
554 \begin{itemize}
555 \item Tidak memiliki parameter \textit{method}
556 \end{itemize}
557 \textbf{Return Value}: Tidak memiliki \textit{return value}
558
559 \textbf{Exception}: Tidak memiliki \textit{exception}
560
561 \item \texttt{public Double getNilaiART()}
562
563 \textbf{Parameter:}
564 \begin{itemize}
565 \item Tidak memiliki parameter \textit{method}
566 \end{itemize}
567 \textbf{Return Value}: Tidak memiliki \textit{return value}
568
569 \textbf{Exception}: Tidak memiliki \textit{exception}
570
571 \item \texttt{public Double getNilaiUTS()}
572
573 \textbf{Parameter:}
574 \begin{itemize}
575 \item Tidak memiliki parameter \textit{method}
576 \end{itemize}
577 \textbf{Return Value}: Tidak memiliki \textit{return value}
578
579 \textbf{Exception}: Tidak memiliki \textit{exception}
580
581 \item \texttt{public Double getNilaiUAS()}
582
583 \textbf{Parameter:}
584 \begin{itemize}
585 \item Tidak memiliki parameter \textit{method}
586 \end{itemize}

```

---

```

587 \textbf{Return Value}: Tidak memiliki \textit{return value}
588
589 \textbf{Exception}: Tidak memiliki \textit{exception}
590
591 \item \texttt{public Character getNilaiAkhir()}\Mengembalikan nilai akhir dalam bentuk huruf (A, B, C, D, ..., atau K)
592
593 \textbf{Parameter:}
594 \begin{itemize}
595 \item Tidak memiliki parameter \textit{method}
596 \end{itemize}
597 \textbf{Return Value}: nilai akhir dalam huruf, atau null jika tidak ada.
598
599 \textbf{Exception}: Tidak memiliki \textit{exception}
600
601 \item \texttt{public Double getAngkaAkhir()}\Mendapatkan nilai akhir dalam bentuk angka
602
603 \textbf{Parameter:}
604 \begin{itemize}
605 \item Tidak memiliki parameter \textit{method}
606 \end{itemize}
607 \textbf{Return Value}: nilai akhir dalam angka, atau null jika getNilaiAkhir() mengembalikan 'K' atau null
608
609 \textbf{Exception}: Tidak memiliki \textit{exception}
610
611 \item \texttt{public TahunSemester getTahunSemester()}\
612
613 \textbf{Parameter:}
614 \begin{itemize}
615 \item Tidak memiliki parameter \textit{method}
616 \end{itemize}
617 \textbf{Return Value}: Tidak memiliki \textit{return value}
618
619 \textbf{Exception}: Tidak memiliki \textit{exception}
620
621 \item \texttt{public int getTahunAjaran()}\
622
623 \textbf{Parameter:}
624 \begin{itemize}
625 \item Tidak memiliki parameter \textit{method}
626 \end{itemize}
627 \textbf{Return Value}: Tidak memiliki \textit{return value}
628
629 \textbf{Exception}: Tidak memiliki \textit{exception}
630
631 \item \texttt{public Semester getSemester()}\
632
633 \textbf{Parameter:}
634 \begin{itemize}
635 \item Tidak memiliki parameter \textit{method}
636 \end{itemize}
637 \textbf{Return Value}: Tidak memiliki \textit{return value}
638
639 \textbf{Exception}: Tidak memiliki \textit{exception}
640
641 \item \texttt{public String toString()}\
642
643 \textbf{Parameter:}
644 \begin{itemize}
645 \item Tidak memiliki parameter \textit{method}
646 \end{itemize}
647 \textbf{Return Value}: Tidak memiliki \textit{return value}
648
649 \textbf{Exception}: Tidak memiliki \textit{exception}
650
651 \end{itemize}
652 \item \texttt{Mahasiswa.Nilai.ChronologicalComparator}
653
654 Pembanding antara satu nilai dengan nilai lainnya, secara
655 kronologis waktu pengambilan.
656
657 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
658 \begin{itemize}
659 \item \texttt{public int compare(id.ac.unpar.siamodels.Mahasiswa.Nilai o1, id.ac.unpar.siamodels.Mahasiswa.Nilai o2)}\
660
661 \textbf{Parameter:}
662 \begin{itemize}
663 \item \texttt{Mahasiswa.Nilai o1} -
664 \item \texttt{Mahasiswa.Nilai o2} -
665 \end{itemize}
666 \textbf{Return Value}: Tidak memiliki \textit{return value}
667
668 \textbf{Exception}: Tidak memiliki \textit{exception}
669
670 \textbf{Override}: \texttt{compare} dari kelas \texttt{Object}
671
672 \end{itemize}
673 \item \texttt{JadwalKuliah}
674
675
676
677 Atribut yang dimiliki kelas ini adalah sebagai berikut.
678 \begin{itemize}
679 \item \texttt{MataKuliah mataKuliah} -
680 \item \texttt{Character kelas} -
681 \item \texttt{DayOfWeek hari} -
682 \item \texttt{LocalTime waktuMulai} -
683 \item \texttt{LocalTime waktuSelesai} -
684 \item \texttt{String lokasi} -
685 \item \texttt{Dosen pengajar} -

```

```

686 \end{itemize}
687 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
688 \begin{itemize}
689 \item \texttt{public MataKuliah getMataKuliah()}
690
691 \textbf{Parameter:}
692 \begin{itemize}
693 \item Tidak memiliki parameter \textit{method}
694 \end{itemize}
695 \textbf{Return Value:} Tidak memiliki \textit{return value}
696
697 \textbf{Exception:} Tidak memiliki \textit{exception}
698
699 \item \texttt{public void setMataKuliah(id.ac.unpar.siamodels.MataKuliah mataKuliah)}
700
701 \textbf{Parameter:}
702 \begin{itemize}
703 \item \texttt{MataKuliah mataKuliah} -
704 \end{itemize}
705 \textbf{Return Value:} Tidak memiliki \textit{return value}
706
707 \textbf{Exception:} Tidak memiliki \textit{exception}
708
709 \item \texttt{public Character getKelas()}
710
711 \textbf{Parameter:}
712 \begin{itemize}
713 \item Tidak memiliki parameter \textit{method}
714 \end{itemize}
715 \textbf{Return Value:} Tidak memiliki \textit{return value}
716
717 \textbf{Exception:} Tidak memiliki \textit{exception}
718
719 \item \texttt{public void setKelas(java.lang.Character kelas)}
720
721 \textbf{Parameter:}
722 \begin{itemize}
723 \item \texttt{Character kelas} -
724 \end{itemize}
725 \textbf{Return Value:} Tidak memiliki \textit{return value}
726
727 \textbf{Exception:} Tidak memiliki \textit{exception}
728
729 \item \texttt{public DayOfWeek getHari()}
730
731 \textbf{Parameter:}
732 \begin{itemize}
733 \item Tidak memiliki parameter \textit{method}
734 \end{itemize}
735 \textbf{Return Value:} Tidak memiliki \textit{return value}
736
737 \textbf{Exception:} Tidak memiliki \textit{exception}
738
739 \item \texttt{public void setHari(java.time.DayOfWeek hari)}
740
741 \textbf{Parameter:}
742 \begin{itemize}
743 \item \texttt{DayOfWeek hari} -
744 \end{itemize}
745 \textbf{Return Value:} Tidak memiliki \textit{return value}
746
747 \textbf{Exception:} Tidak memiliki \textit{exception}
748
749 \item \texttt{public LocalTime getWaktuMulai()}
750
751 \textbf{Parameter:}
752 \begin{itemize}
753 \item Tidak memiliki parameter \textit{method}
754 \end{itemize}
755 \textbf{Return Value:} Tidak memiliki \textit{return value}
756
757 \textbf{Exception:} Tidak memiliki \textit{exception}
758
759 \item \texttt{public void setWaktuMulai(java.time.LocalTime waktuMulai)}
760
761 \textbf{Parameter:}
762 \begin{itemize}
763 \item \texttt{LocalTime waktuMulai} -
764 \end{itemize}
765 \textbf{Return Value:} Tidak memiliki \textit{return value}
766
767 \textbf{Exception:} Tidak memiliki \textit{exception}
768
769 \item \texttt{public LocalTime getWaktuSelesai()}
770
771 \textbf{Parameter:}
772 \begin{itemize}
773 \item Tidak memiliki parameter \textit{method}
774 \end{itemize}
775 \textbf{Return Value:} Tidak memiliki \textit{return value}
776
777 \textbf{Exception:} Tidak memiliki \textit{exception}
778
779 \item \texttt{public void setWaktuSelesai(java.time.LocalTime waktuSelesai)}
780
781 \textbf{Parameter:}
782 \begin{itemize}
783 \item \texttt{LocalTime waktuSelesai} -
784 \end{itemize}

```

---

```

785 \textbf{Return Value}: Tidak memiliki \textit{return value}
786
787 \textbf{Exception}: Tidak memiliki \textit{exception}
788
789 \item \texttt{public String getLocation()}
790
791 \textbf{Parameter:}
792 \begin{itemize}
793 \item Tidak memiliki parameter \textit{method}
794 \end{itemize}
795 \textbf{Return Value}: Tidak memiliki \textit{return value}
796
797 \textbf{Exception}: Tidak memiliki \textit{exception}
798
799 \item \texttt{public void setLocation(java.lang.String lokasi)}
800
801 \textbf{Parameter:}
802 \begin{itemize}
803 \item \texttt{String lokasi} -
804 \end{itemize}
805 \textbf{Return Value}: Tidak memiliki \textit{return value}
806
807 \textbf{Exception}: Tidak memiliki \textit{exception}
808
809 \item \texttt{public Dosen getPengajar()}
810
811 \textbf{Parameter:}
812 \begin{itemize}
813 \item Tidak memiliki parameter \textit{method}
814 \end{itemize}
815 \textbf{Return Value}: Tidak memiliki \textit{return value}
816
817 \textbf{Exception}: Tidak memiliki \textit{exception}
818
819 \item \texttt{public void setPengajar(id.ac.unpar.siamodels.Dosen pengajar)}
820
821 \textbf{Parameter:}
822 \begin{itemize}
823 \item \texttt{Dosen pengajar} -
824 \end{itemize}
825 \textbf{Return Value}: Tidak memiliki \textit{return value}
826
827 \textbf{Exception}: Tidak memiliki \textit{exception}
828
829 \item \texttt{public String getWaktuString()}
830
831 \textbf{Parameter:}
832 \begin{itemize}
833 \item Tidak memiliki parameter \textit{method}
834 \end{itemize}
835 \textbf{Return Value}: Tidak memiliki \textit{return value}
836
837 \textbf{Exception}: Tidak memiliki \textit{exception}
838
839 \item \texttt{public static DayOfWeek indonesianToDayOfWeek(java.lang.String indonesian)} Converts Indonesian day names to \texttt{DayOfWeek} enumeration.
840
841 \textbf{Parameter:}
842 \begin{itemize}
843 \item \texttt{String indonesian} -
844 the day name in Indonesian
845 \end{itemize}
846 \textbf{Return Value}: DayOfWeek object or null if not found.
847
848 \textbf{Exception}: Tidak memiliki \textit{exception}
849
850 \end{itemize}
851 \item \texttt{MataKuliah}
852
853
854
855 Atribut yang dimiliki kelas ini adalah sebagai berikut.
856 \begin{itemize}
857 \item \texttt{String kode} -
858 \item \texttt{String nama} -
859 \item \texttt{Integer sks} -
860 \end{itemize}
861 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
862 \begin{itemize}
863 \item \texttt{public String getCode()}
864
865 \textbf{Parameter:}
866 \begin{itemize}
867 \item Tidak memiliki parameter \textit{method}
868 \end{itemize}
869 \textbf{Return Value}: Tidak memiliki \textit{return value}
870
871 \textbf{Exception}: Tidak memiliki \textit{exception}
872
873 \item \texttt{public String getName()}
874
875 \textbf{Parameter:}
876 \begin{itemize}
877 \item Tidak memiliki parameter \textit{method}
878 \end{itemize}
879 \textbf{Return Value}: Tidak memiliki \textit{return value}
880
881 \textbf{Exception}: Tidak memiliki \textit{exception}
882

```

```

883 \item \texttt{public Integer getSks()}
884
885 \textbf{Parameter:}
886 \begin{itemize}
887 \item Tidak memiliki parameter \textit{method}
888 \end{itemize}
889 \textbf{Return Value:} Tidak memiliki \textit{return value}
890
891 \textbf{Exception:} Tidak memiliki \textit{exception}
892
893 \end{itemize}
894 \item \texttt{Dosen}
895
896
897
898 Atribut yang dimiliki kelas ini adalah sebagai berikut.
899 \begin{itemize}
900 \item \texttt{String nik} -
901 \item \texttt{String nama} -
902 \end{itemize}
903 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
904 \begin{itemize}
905 \item \texttt{public String getNik()}
906
907 \textbf{Parameter:}
908 \begin{itemize}
909 \item Tidak memiliki parameter \textit{method}
910 \end{itemize}
911 \textbf{Return Value:} Tidak memiliki \textit{return value}
912
913 \textbf{Exception:} Tidak memiliki \textit{exception}
914
915 \item \texttt{public void setNik(java.lang.String nik)}
916
917 \textbf{Parameter:}
918 \begin{itemize}
919 \item \texttt{String nik} -
920 \end{itemize}
921 \textbf{Return Value:} Tidak memiliki \textit{return value}
922
923 \textbf{Exception:} Tidak memiliki \textit{exception}
924
925 \item \texttt{public String getNama()}
926
927 \textbf{Parameter:}
928 \begin{itemize}
929 \item Tidak memiliki parameter \textit{method}
930 \end{itemize}
931 \textbf{Return Value:} Tidak memiliki \textit{return value}
932
933 \textbf{Exception:} Tidak memiliki \textit{exception}
934
935 \item \texttt{public void setNama(java.lang.String nama)}
936
937 \textbf{Parameter:}
938 \begin{itemize}
939 \item \texttt{String nama} -
940 \end{itemize}
941 \textbf{Return Value:} Tidak memiliki \textit{return value}
942
943 \textbf{Exception:} Tidak memiliki \textit{exception}
944
945 \item \texttt{public boolean equals(java.lang.Object arg0)}
946
947 \textbf{Parameter:}
948 \begin{itemize}
949 \item \texttt{Object arg0} -
950 \end{itemize}
951 \textbf{Return Value:} Tidak memiliki \textit{return value}
952
953 \textbf{Exception:} Tidak memiliki \textit{exception}
954
955 \end{itemize}
956 \item \texttt{MKU008}
957
958 Mendalami perilaku sehari-hari yang baik dalam bermasyarakat.
959
960 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{IIE210}
961
962
963
964 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF203}
965
966 Mata kuliah ini memperkenalkan kepada mahasiswa konsep struktur diskret yang
967 digunakan pada bidang informatika diantaranya graph, pohon dan finite state
968 machine
969
970 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
971 \begin{itemize}
972 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
973
974 \textbf{Parameter:}
975 \begin{itemize}
976 \item \texttt{Mahasiswa mahasiswa} -
977 \item \texttt{java.util.List reasonsContainer} -
978 \end{itemize}
979 \textbf{Return Value:} Tidak memiliki \textit{return value}
980
981 \textbf{Exception:} Tidak memiliki \textit{exception}

```



```

982 | \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
983 |
984 |
985 | \end{itemize}
986 | \item \texttt{AIF311}
987 |
988 | Kuliah Pemrograman Fungsional bertujuan untuk: 1. memperkenalkan paradigma
989 | pemrograman fungsional, yaitu sebuah pemrograman yang didasarkan pada konsep
990 | pemetaan dan fungsi matematika. Penyelesaian suatu masalah didasari atas
991 | aplikasi dari fungsi-fungsi tersebut. 2. memberikan dasar-dasar pemrograman
992 | fungsional dengan menggunakan bahasa fungsional Haskell.
993 |
994 | Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
995 | \begin{itemize}
996 | \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
997 |
998 | \textbf{Parameter}:
999 | \begin{itemize}
1000 | \item \texttt{Mahasiswa mahasiswa} -
1001 | \item \texttt{java.util.List reasonsContainer} -
1002 | \end{itemize}
1003 | \textbf{Return Value}: Tidak memiliki \textit{return value}
1004 |
1005 | \textbf{Exception}: Tidak memiliki \textit{exception}
1006 |
1007 | \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1008 |
1009 | \end{itemize}
1010 | \item \texttt{AIF192}
1011 |
1012 |
1013 |
1014 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF468}
1015 |
1016 |
1017 |
1018 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{IIE103}
1019 |
1020 |
1021 |
1022 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF385}
1023 |
1024 |
1025 |
1026 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF106}
1027 |
1028 | Mata kuliah ini memberikan pengetahuan tentang cara kerja komputer, dimulai
1029 | dari representasi data dan berbagai macam operasinya. Selanjutnya, juga
1030 | diperkenalkan bagaimana merepresentasikan suatu fungsi dalam rangkaian
1031 | gerbang logika, dan bagaimana menyederhanakannya. Berbagai rangkaian dasar
1032 | yang digunakan di dalam komputer juga diperkenalkan. Mahasiswa juga akan
1033 | mempelajari komponen komputer, misalnya register dan memori.
1034 |
1035 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF281}
1036 |
1037 |
1038 |
1039 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{EAA102}
1040 |
1041 |
1042 |
1043 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF405}
1044 |
1045 | Mata kuliah ini merupakan lanjutan dari Projek Sistem Informasi 1 dan
1046 | memberikan kesempatan bagi mahasiswa untuk melanjutkan/mengembangkan
1047 | perancangan sitem pada organisasi studi kasus, mengimplementasikan rancangan
1048 | dan melakukan pengujian perangkat lunak;
1049 |
1050 | Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1051 | \begin{itemize}
1052 | \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1053 |
1054 | \textbf{Parameter}:
1055 | \begin{itemize}
1056 | \item \texttt{Mahasiswa mahasiswa} -
1057 | \item \texttt{java.util.List reasonsContainer} -
1058 | \end{itemize}
1059 | \textbf{Return Value}: Tidak memiliki \textit{return value}
1060 |
1061 | \textbf{Exception}: Tidak memiliki \textit{exception}
1062 |
1063 | \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1064 |
1065 | \end{itemize}
1066 | \item \texttt{APS182}
1067 |
1068 |
1069 |
1070 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{MKU004}
1071 |
1072 | Fenomenologi Agama merupakan bagian yang tak terpisahkan dari kajian filosofis, kritis,
1073 | rasional, dan obyektif mengenai substansi ajaran agama. Fenomenologi merupakan sebuah
1074 | disiplin ilmu yang secara kritis-rasional mengkaji fenomena dan dinamika kehidupan manusia
1075 | beragama, dari upaya menjadikan Tuhan sebagai tujuan sesembahan sampai menempatkan Tuhan
1076 | sebagai instrumen legitimasi untuk melakukan tindakan yang justru bertolak belakang dengan
1077 | kehendak Tuhan yang disembah. Sehubungan dengan itu, kritik konstruktif terhadap perilaku
1078 | manusia beragama menjadi salah satu poin utama dalam mata kuliah ini. Kesiadaan untuk
1079 | melakukan otoritik terhadap agama sendiri erat terkait dengan upaya menemukan kembali nilai
1080 | sejati agama atau otentisitas hidup beragama.

```

1081 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{MKU012}

1082

1083 Perkuliahan logika ditujukan untuk memberikan dasar-dasar ketrampilan berpikir rasional dan

1084 sistematis. Isinya mencakup ketrampilan berpikir deduktif dan induktif, seperti silogisme,

1085 argumen analogikal dan generalisasi induktif. Pembahasan teoretis disertai pula dengan

1086 pelatihan praktis yang diarahkan pada proses berpikir. Untuk menajamkan kemampuan berpikir

1087 tersebut, mahasiswa dilatih pula mengidentifikasi kerancuan-kerancuan (fallacies) yang sering

1088 dijumpai baik dalam kehidupan sehari-hari maupun dalam konteks akademik.

1089

1090 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF389}

1091

1092

1093

1094 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AMS190}

1095

1096

1097

1098

1099 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AMS191}

1100

1101

1102 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AMS200}

1103

1104

1105

1106 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF330}

1107

1108

1109

1110

1111 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF388}

1112

1113

1114 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF465}

1115

1116

1117

1118

1119 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF453}

1120

1121 Mata kuliah ini memperkenalkan kebutuhan organisasi terhadap sistem business

1122 intelligent (BI) dan pemanfaatan BI untuk organisasi; memperkenalkan konsep

1123 sistem business intelligent dan komponennya; Mempelajari teknik-teknik

1124 analisis data bisnis dan visualisasi hasil analisis; Mempelajari konsep data

1125 warehouse dan perancangannya dan fungsi OLAP; Mempraktekkan teknik-teknik

1126 analisis data dan visualisasi hasil analisis.

1127

1128 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.

1129 \begin{itemize}

1130 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}

1131

1132 \textbf{Parameter:}

1133 \begin{itemize}

1134 \item \texttt{Mahasiswa mahasiswa} -

1135 \item \texttt{java.util.List reasonsContainer} -

1136 \end{itemize}

1137 \textbf{Return Value:} Tidak memiliki \textit{return value}

1138

1139 \textbf{Exception:} Tidak memiliki \textit{exception}

1140

1141 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}

1142

1143 \end{itemize}

1144 \item \texttt{AIF280}

1145

1146

1147

1148 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF445}

1149

1150

1151

1152 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.

1153 \begin{itemize}

1154 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}

1155

1156 \textbf{Parameter:}

1157 \begin{itemize}

1158 \item \texttt{Mahasiswa mahasiswa} -

1159 \item \texttt{java.util.List reasonsContainer} -

1160 \end{itemize}

1161 \textbf{Return Value:} Tidak memiliki \textit{return value}

1162

1163 \textbf{Exception:} Tidak memiliki \textit{exception}

1164

1165 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}

1166

1167 \end{itemize}

1168 \item \texttt{AIF469}

1169

1170 Mata kuliah ini mengajarkan kepada mahasiswa teknik-teknik untuk membuat

1171 layanan berbasis web. Mahasiswa diperkenalkan dengan standar-standar seperti

1172 HTTP, XML, JSON dan diajarkan untuk memanfaatkannya dalam membuat maupun

1173 menggunakan layanan pihak ketiga. Dalam kuliah ini, juga akan diperkenalkan

1174 minimal satu layanan pihak ketiga yang dapat dimanfaatkan mahasiswa, seperti

1175 Google Places Web Service.

1176

1177 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.

1178 \begin{itemize}

1179 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}

---

```

1180 |
1181 | \textbf{Parameter:}
1182 | \begin{itemize}
1183 | \item \texttt{Mahasiswa mahasiswa} -
1184 | \item \texttt{java.util.List reasonsContainer} -
1185 | \end{itemize}
1186 | \textbf{Return Value}: Tidak memiliki \textit{return value}
1187 |
1188 | \textbf{Exception}: Tidak memiliki \textit{exception}
1189 |
1190 | \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1191 |
1192 | \end{itemize}
1193 | \item \texttt{AIF486}
1194 |
1195 |
1196 |
1197 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{IEE207}
1198 |
1199 |
1200 |
1201 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF202}
1202 |
1203 | Mata kuliah ini memperkenalkan kepada mahasiswa beberapa algoritma dan
1204 | struktur data, alternatif cara implementasinya, dan analisis kompleksitas
1205 | waktunya. Mahasiswa diberikan beberapa masalah komputasi yang harus
1206 | diselesaikan dengan menggunakan algoritma atau struktur data yang sudah
1207 | diperkenalkan dan mengimplementasikannya dalam bahasa pemrograman Java.
1208 |
1209 | Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1210 | \begin{itemize}
1211 | \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1212 |
1213 | \textbf{Parameter:}
1214 | \begin{itemize}
1215 | \item \texttt{Mahasiswa mahasiswa} -
1216 | \item \texttt{java.util.List reasonsContainer} -
1217 | \end{itemize}
1218 | \textbf{Return Value}: Tidak memiliki \textit{return value}
1219 |
1220 | \textbf{Exception}: Tidak memiliki \textit{exception}
1221 |
1222 | \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1223 |
1224 | \end{itemize}
1225 | \item \texttt{AIF347}
1226 |
1227 |
1228 |
1229 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{MKU009}
1230 |
1231 | Tujuan dari mata kuliah ini adalah untuk mendalami keterampilan berbahasa Indonesia, agar
1232 | mampu mengkomunikasikan hasil pemikiran serta meningkatkan keterampilan dalam menyusun karya
1233 | ilmiah. Mata kuliah Bahasa Indonesia ini dimulai dengan mempelajari penulisan kata baku dan
1234 | non baku serta pengungkapan pikiran dengan puntuasi yang benar. Selanjutnya dipelajari
1235 | penyusunan kalimat yang baku serta menghubungkan kalimat-kalimat yang padu dalam menuangkan
1236 | gagasan dalam sebuah paragraf. Selain itu, dalam matakuliah ini dipelajari cara menyusun
1237 | surat dinas yang jelas dan komunikatif. Di akhir kuliah ini, mahasiswa diberi tugas
1238 | penyusunan makalah dengan benar.
1239 |
1240 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{ESM203}
1241 |
1242 |
1243 |
1244 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{APS402}
1245 |
1246 |
1247 |
1248 | Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1249 | \begin{itemize}
1250 | \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1251 |
1252 | \textbf{Parameter:}
1253 | \begin{itemize}
1254 | \item \texttt{Mahasiswa mahasiswa} -
1255 | \item \texttt{java.util.List reasonsContainer} -
1256 | \end{itemize}
1257 | \textbf{Return Value}: Tidak memiliki \textit{return value}
1258 |
1259 | \textbf{Exception}: Tidak memiliki \textit{exception}
1260 |
1261 | \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1262 |
1263 | \end{itemize}
1264 | \item \texttt{AIF306}
1265 |
1266 | Mata kuliah ini bertujuan untuk memberikan pengalaman bagi mahasiswa dalam
1267 | mengerjakan proyek dengan teknologi-teknologi terkini, secara berkelompok.
1268 | Teknologi-teknologi yang digunakan pada kuliah ini tidak spesifik dan dapat
1269 | berubah seiring perkembangan teknologi maupun disesuaikan dengan kompetensi
1270 | dosen pengajar. Beberapa teknologi yang dapat dimanfaatkan antara lain: DVCS
1271 | tool menggunakan Git + Github, Mobile native app (Android, iOS, dll), dan
1272 | responsive web design.
1273 |
1274 | Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1275 | \begin{itemize}
1276 | \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1277 |
1278 | \textbf{Parameter:}

```

```

1279 \begin{itemize}
1280 \item \texttt{Mahasiswa mahasiswa} -
1281 \item \texttt{java.util.List reasonsContainer} -
1282 \end{itemize}
1283 \textbf{Return Value}: Tidak memiliki \textit{return value}
1284
1285 \textbf{Exception}: Tidak memiliki \textit{exception}
1286
1287 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1288
1289 \end{itemize}
1290 \item \texttt{AKS122}
1291
1292
1293
1294 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{MKU002}
1295
1296 Pendidikan Kewarganegaraan menjelaskan pentingnya pemahaman tentang identitas nasional
1297 Indonesia, hak dan kewajiban warga negara Indonesia serta hubungannya dengan hak dan
1298 kewajiban asasi manusia. Materi kuliah mencakup juga wawasan nusantara, ketahanan nasional,
1299 politik dan strategi nasional, serta implementasinya dalam kehidupan bermasyarakat, berbangsa
1300 dan bernegara kesatuan Republik Indonesia.
1301
1302 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF462}
1303
1304
1305
1306 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF360}
1307
1308
1309
1310 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1311 \begin{itemize}
1312 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1313
1314 \textbf{Parameter}:
1315 \begin{itemize}
1316 \item \texttt{Mahasiswa mahasiswa} -
1317 \item \texttt{java.util.List reasonsContainer} -
1318 \end{itemize}
1319 \textbf{Return Value}: Tidak memiliki \textit{return value}
1320
1321 \textbf{Exception}: Tidak memiliki \textit{exception}
1322
1323 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1324
1325 \end{itemize}
1326 \item \texttt{AIF337}
1327
1328
1329
1330 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF458}
1331
1332
1333
1334 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1335 \begin{itemize}
1336 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1337
1338 \textbf{Parameter}:
1339 \begin{itemize}
1340 \item \texttt{Mahasiswa mahasiswa} -
1341 \item \texttt{java.util.List reasonsContainer} -
1342 \end{itemize}
1343 \textbf{Return Value}: Tidak memiliki \textit{return value}
1344
1345 \textbf{Exception}: Tidak memiliki \textit{exception}
1346
1347 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1348
1349 \end{itemize}
1350 \item \texttt{AIF301}
1351
1352 Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar tentang *sistem
1353 cerdas dan komponen-komponennya. " Terdapat 4 topik utama yang dibahas yaitu
1354 teknik pencarian untuk *penyelesaian masalah, representasi pengetahuan dalam
1355 sistem *cerdas, pemodelan ketidakpastian dalam masalah dan teknik
1356 pembelajaran mesin.
1357
1358 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1359 \begin{itemize}
1360 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1361
1362 \textbf{Parameter}:
1363 \begin{itemize}
1364 \item \texttt{Mahasiswa mahasiswa} -
1365 \item \texttt{java.util.List reasonsContainer} -
1366 \end{itemize}
1367 \textbf{Return Value}: Tidak memiliki \textit{return value}
1368
1369 \textbf{Exception}: Tidak memiliki \textit{exception}
1370
1371 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1372
1373 \end{itemize}
1374 \item \texttt{AIF182}
1375
1376
1377

```

---

```

1378 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{ESM204}
1379
1380
1381
1382 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF205}
1383
1384 Mata kuliah ini memperkenalkan kepada mahasiswa arsitektur komputer
1385 sederhana, modern, dan Advance. Perbedaan, kelebihan dan kekurangan untuk
1386 masing-masing arsitektur. Selain itu mahasiswa juga mempelajari cara kerja
1387 dari komponen-komponen komputer, terutama memory, cache, system BUS dan
1388 input/output.
1389
1390 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1391 \begin{itemize}
1392 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1393
1394 \textbf{Parameter:}
1395 \begin{itemize}
1396 \item \texttt{Mahasiswa mahasiswa} -
1397 \item \texttt{java.util.List reasonsContainer} -
1398 \end{itemize}
1399 \textbf{Return Value:} Tidak memiliki \textit{return value}
1400
1401 \textbf{Exception:} Tidak memiliki \textit{exception}
1402
1403 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1404
1405 \end{itemize}
1406 \item \texttt{AIF317}
1407
1408
1409
1410 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1411 \begin{itemize}
1412 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1413
1414 \textbf{Parameter:}
1415 \begin{itemize}
1416 \item \texttt{Mahasiswa mahasiswa} -
1417 \item \texttt{java.util.List reasonsContainer} -
1418 \end{itemize}
1419 \textbf{Return Value:} Tidak memiliki \textit{return value}
1420
1421 \textbf{Exception:} Tidak memiliki \textit{exception}
1422
1423 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1424
1425 \end{itemize}
1426 \item \texttt{AIF383}
1427
1428
1429
1430 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF442}
1431
1432
1433
1434 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1435 \begin{itemize}
1436 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1437
1438 \textbf{Parameter:}
1439 \begin{itemize}
1440 \item \texttt{Mahasiswa mahasiswa} -
1441 \item \texttt{java.util.List reasonsContainer} -
1442 \end{itemize}
1443 \textbf{Return Value:} Tidak memiliki \textit{return value}
1444
1445 \textbf{Exception:} Tidak memiliki \textit{exception}
1446
1447 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1448
1449 \end{itemize}
1450 \item \texttt{ESM101}
1451
1452
1453
1454 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF403}
1455
1456 1. Memberikan wawasan kepada mahasiswa tentang kemunculan dan pemanfaatan teknologi baru,
1457 khususnya yang berkaitan dengan komputer, dan dampaknya terhadap masyarakat luas.
1458 2. Memberikan kesadaran dan panduan bersikap kepada mahasiswa dalam menghadapi gejala yang
1459 disebabkan oleh munculnya teknologi baru, khususnya yang berkaitan dengan komputer.
1460
1461 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1462 \begin{itemize}
1463 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1464
1465 \textbf{Parameter:}
1466 \begin{itemize}
1467 \item \texttt{Mahasiswa mahasiswa} -
1468 \item \texttt{java.util.List reasonsContainer} -
1469 \end{itemize}
1470 \textbf{Return Value:} Tidak memiliki \textit{return value}
1471
1472 \textbf{Exception:} Tidak memiliki \textit{exception}
1473
1474 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1475
1476 \end{itemize}

```

```

1477 \item \texttt{AIF402}
1478
1479
1480
1481 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1482 \begin{itemize}
1483 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1484
1485 \textbf{Parameter:}
1486 \begin{itemize}
1487 \item \texttt{Mahasiswa mahasiswa} -
1488 \item \texttt{java.util.List reasonsContainer} -
1489 \end{itemize}
1490 \textbf{Return Value:} Tidak memiliki \textit{return value}
1491
1492 \textbf{Exception:} Tidak memiliki \textit{exception}
1493
1494 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1495
1496 \end{itemize}
1497 \item \texttt{AIF455}
1498
1499
1500
1501 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF443}
1502
1503
1504
1505 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF101}
1506
1507 Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar pemrograman
1508 seperti pengulangan dan percabangan, konsep dasar penyimpanan data kontigu
1509 menggunakan array, konsep dasar pemrograman berorientasi objek seperti kelas
1510 & objek, method, dll, termasuk di dalamnya 4 prinsip dasar pemrograman
1511 berorientasi objek : data abstraction, encapsulation, inheritance dan
1512 polymorphism. Selain itu diberikan masalah-masalah komputasi sederhana
1513 yang harus diselesaikan menggunakan konsep-konsep yang sudah diperkenalkan
1514 dan mengimplementasikannya menggunakan bahasa pemrograman Java
1515
1516 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF382}
1517
1518
1519
1520 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF480}
1521
1522
1523
1524 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF316}
1525
1526 Mata kuliah ini memperkenalkan konsep-konsep dasar komputasi paralel, dimana sebuah
1527 program yang berjalan secara paralel harus memiliki safety property dan liveness property.
1528 Mahasiswa dikenalkan dengan beberapa teknik pemrograman multi-thread
1529 seperti lock, monitor, barrier, thread pool, dan sebagainya, yang diimplementasikan
1530 dalam bahasa pemrograman Java. Mahasiswa juga dikenalkan dengan beberapa metode untuk
1531 menganalisis kebenaran program baik secara matematis maupun secara praktis dengan bantuan
1532 model checker.
1533
1534 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1535 \begin{itemize}
1536 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1537
1538 \textbf{Parameter:}
1539 \begin{itemize}
1540 \item \texttt{Mahasiswa mahasiswa} -
1541 \item \texttt{java.util.List reasonsContainer} -
1542 \end{itemize}
1543 \textbf{Return Value:} Tidak memiliki \textit{return value}
1544
1545 \textbf{Exception:} Tidak memiliki \textit{exception}
1546
1547 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1548
1549 \end{itemize}
1550 \item \texttt{AIF438}
1551
1552 Mata kuliah ini: Memperkenalkan karakteristik dan teknik visualisasi dari
1553 berbagai jenis data yang dapat dianalisis dengan teknik-teknik data mining;
1554 mempelajari teknik-teknik penyiapan data untuk berbagai jenis data dan teknik
1555 data mining; mempraktekkan teknik-teknik penyiapan data untuk menganalisis
1556 data nyata/simulasi dengan memanfaatkan perangkat lunak aplikasi.
1557
1558 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1559 \begin{itemize}
1560 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1561
1562 \textbf{Parameter:}
1563 \begin{itemize}
1564 \item \texttt{Mahasiswa mahasiswa} -
1565 \item \texttt{java.util.List reasonsContainer} -
1566 \end{itemize}
1567 \textbf{Return Value:} Tidak memiliki \textit{return value}
1568
1569 \textbf{Exception:} Tidak memiliki \textit{exception}
1570
1571 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1572
1573 \end{itemize}
1574 \item \texttt{AIF204}
1575

```

```

1576 Mata kuliah ini memperkenalkan konsep dan arsitektur DBMS, mengajarkan
1577 aljabar relasional dan SQL serta pemanfaatannya pada pemrograman kueri
1578 sederhana s/d relatif kompleks. Selain itu, mata kuliah ini juga mengajarkan
1579 dan mempraktekkan perancangan basisdata untuk masalah sederhana
1580 (lingkup kecil) termasuk pengembangan program aplikasinya;
1581
1582 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1583 \begin{itemize}
1584 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1585
1586 \textbf{Parameter:}
1587 \begin{itemize}
1588 \item \texttt{Mahasiswa mahasiswa} -
1589 \item \texttt{java.util.List reasonsContainer} -
1590 \end{itemize}
1591 \textbf{Return Value:} Tidak memiliki \textit{return value}
1592
1593 \textbf{Exception:} Tidak memiliki \textit{exception}
1594
1595 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1596
1597 \end{itemize}
1598 \item \texttt{AIF341}
1599
1600 Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar
1601 jaringan dan aplikasinya di kehidupan sehari-hari. Mahasiswa
1602 dikenalkan dengan teknologi-teknologi terbaru di bidang jaringan,
1603 sehingga mahasiswa memiliki pengetahuan yang dapat digunakan
1604 dalam kehidupan sehari-hari. Selain itu mahasiswa juga
1605 diperkenalkan dengan NetAcad, sebuah layanan dari Cisco yang
1606 dapat digunakan untuk memenuhi segala macam kebutuhan terkait
1607 dengan Cisco Academy.
1608
1609 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF183}
1610
1611
1612
1613 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AKS124}
1614
1615
1616
1617 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF459}
1618
1619
1620
1621 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF336}
1622
1623 Mata kuliah ini merupakan mata kuliah lanjutan dari mata kuliah Keamanan
1624 Informasi, dengan titik berat pada materi kriptografi. Mata kuliah ini
1625 memperkenalkan tambahan konsep kriptografi, misalnya tentang otentikasi
1626 yaitu otentikasi entitas, manajemen kunci, dan bentuk lain dari metode
1627 merahasiakan pesan, yaitu dengan menggunakan secret sharing. Selanjutnya,
1628 diperkenalkan juga penggunaan kriptografi pada protokol-protokol yang
1629 sebenarnya banyak digunakan sehari-hari, misalnya pada e-cash, auction,
1630 dan electronic voting.
1631
1632 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF463}
1633
1634
1635
1636 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF208}
1637
1638 Mata kuliah ini memperkenalkan kepada mahasiswa tahapan rekayasa perangkat
1639 lunak, terutama dengan paradigma berorientasi objek, dilengkapi dengan
1640 pengenalan tentang manajemen proyek perangkat lunak.
1641 Selain, itu diberikan deskripsi proyek berskala kecil yang harus dikerjakan
1642 oleh mahasiswa dalam kelompok dengan menerapkan teori yang telah
1643 dipelajarinya.
1644
1645 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1646 \begin{itemize}
1647 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1648
1649 \textbf{Parameter:}
1650 \begin{itemize}
1651 \item \texttt{Mahasiswa mahasiswa} -
1652 \item \texttt{java.util.List reasonsContainer} -
1653 \end{itemize}
1654 \textbf{Return Value:} Tidak memiliki \textit{return value}
1655
1656 \textbf{Exception:} Tidak memiliki \textit{exception}
1657
1658 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1659
1660 \end{itemize}
1661 \item \texttt{MKU003}
1662
1663 Mata kuliah ini membentuk karakteristik mahasiswa sebagai manusia yang memiliki religiusitas
1664 melalui pendalaman akan makna agama dan beragama, mendeteksi dinamika Wahyu Tuhan dan iman
1665 mereka, memahami relasi dengan Tuhan dan sesama, mengenal makna keselamatan dalam konteks
1666 Kerajaan Allah, dan mampu menyatakan ajaran Gereja dalam pelayanan terhadap orang miskin dan
1667 terlantar.
1668
1669 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{APS309}
1670
1671 APS302 atau APS309 ?
1672
1673 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF335}
1674

```

```

1675
1676
1677 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF362}
1678
1679
1680
1681 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1682 \begin{itemize}
1683 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1684
1685 \textbf{Parameter:}
1686 \begin{itemize}
1687 \item \texttt{Mahasiswa mahasiswa} -
1688 \item \texttt{java.util.List reasonsContainer} -
1689 \end{itemize}
1690 \textbf{Return Value:} Tidak memiliki \textit{return value}
1691
1692 \textbf{Exception:} Tidak memiliki \textit{exception}
1693
1694 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1695
1696 \end{itemize}
1697 \item \texttt{AIF460}
1698
1699
1700
1701 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF358}
1702
1703
1704
1705 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AMS100}
1706
1707 Sistem Bilangan, Fungsi, Limit dan Kekontinuan Fungsi, Turunan, Integral,
1708 Penggunaan Integral, Sistem Persamaan Linear, Determinan, Vektor, Nilai dan
1709 Vektor Eigen.
1710
1711 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF401}
1712
1713
1714
1715 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1716 \begin{itemize}
1717 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1718
1719 \textbf{Parameter:}
1720 \begin{itemize}
1721 \item \texttt{Mahasiswa mahasiswa} -
1722 \item \texttt{java.util.List reasonsContainer} -
1723 \end{itemize}
1724 \textbf{Return Value:} Tidak memiliki \textit{return value}
1725
1726 \textbf{Exception:} Tidak memiliki \textit{exception}
1727
1728 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1729
1730 \end{itemize}
1731 \item \texttt{AIF456}
1732
1733
1734
1735 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{SIR104}
1736
1737
1738
1739 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF339}
1740
1741
1742
1743 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1744 \begin{itemize}
1745 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1746
1747 \textbf{Parameter:}
1748 \begin{itemize}
1749 \item \texttt{Mahasiswa mahasiswa} -
1750 \item \texttt{java.util.List reasonsContainer} -
1751 \end{itemize}
1752 \textbf{Return Value:} Tidak memiliki \textit{return value}
1753
1754 \textbf{Exception:} Tidak memiliki \textit{exception}
1755
1756 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1757
1758 \end{itemize}
1759 \item \texttt{AIF102}
1760
1761 Mata kuliah ini memperkenalkan berbagai algoritma dan teknik-teknik
1762 penyelesaian masalah komputasi seperti rekursif, sorting, teknik divide dan
1763 conquer, serta exhaustive search. Selain itu, pada kuliah ini juga
1764 dikenalkan berbagai struktur data yang dapat digunakan untuk mendukung
1765 penyelesaian masalah komputasi seperti ADT List, Stack dan Queue. Baik
1766 algoritma maupun struktur data yang dikenalkan harus dapat diimplementasikan
1767 dan digunakan oleh mahasiswa untuk menyelesaikan masalah dengan menggunakan
1768 suatu bahasa pemrograman berorientasi objek.
1769
1770 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1771 \begin{itemize}
1772 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1773

```



```

1774 \textbf{Parameter:}
1775 \begin{itemize}
1776 \item \texttt{Mahasiswa mahasiswa} -
1777 \item \texttt{java.util.List reasonsContainer} -
1778 \end{itemize}
1779 \textbf{Return Value}: Tidak memiliki \textit{return value}
1780
1781 \textbf{Exception}: Tidak memiliki \textit{exception}
1782
1783 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1784
1785 \end{itemize}
1786 \item \texttt{AIF381}
1787
1788
1789
1790 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF483}
1791
1792
1793
1794 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF315}
1795
1796 Mata kuliah ini memperkenalkan konsep dan lingkungan pemrograman berbasis web,
1797 kemudian belajar membuat aplikasi berbasis web menggunakan HTML5, CSS, Java Script
1798 dan PHP. Untuk meningkatkan keterampilan pemrograman dilengkapi dengan praktikum.
1799 Sedangkan untuk mendapatkan pengalaman penerapan konsep diberikan tugas besar membuat
1800 program berbasis web dengan kasus yang ditentukan oleh mahasiswa.
1801
1802 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1803 \begin{itemize}
1804 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1805
1806 \textbf{Parameter:}
1807 \begin{itemize}
1808 \item \texttt{Mahasiswa mahasiswa} -
1809 \item \texttt{java.util.List reasonsContainer} -
1810 \end{itemize}
1811 \textbf{Return Value}: Tidak memiliki \textit{return value}
1812
1813 \textbf{Exception}: Tidak memiliki \textit{exception}
1814
1815 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1816
1817 \end{itemize}
1818 \item \texttt{AIF342}
1819
1820
1821
1822 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1823 \begin{itemize}
1824 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1825
1826 \textbf{Parameter:}
1827 \begin{itemize}
1828 \item \texttt{Mahasiswa mahasiswa} -
1829 \item \texttt{java.util.List reasonsContainer} -
1830 \end{itemize}
1831 \textbf{Return Value}: Tidak memiliki \textit{return value}
1832
1833 \textbf{Exception}: Tidak memiliki \textit{exception}
1834
1835 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1836
1837 \end{itemize}
1838 \item \texttt{IIE214}
1839
1840
1841
1842 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF303}
1843
1844 Mempelajari Konsep Data, Informasi, Pengetahuan, Sistem Informasi, proses dan
1845 pemodelan bisnis, jenis-jenis sistem informasi, untuk mendukung pengambilan
1846 keputusan. Mempelajari trend Teknologi Informasi, tahap-tahap pembangunan
1847 sistem informasi. Mempelajari pengantar : EIS, e-bisnis/e-commerce, Business
1848 Intelligence, Cloud Computing dan Mobile Applications
1849
1850 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1851 \begin{itemize}
1852 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1853
1854 \textbf{Parameter:}
1855 \begin{itemize}
1856 \item \texttt{Mahasiswa mahasiswa} -
1857 \item \texttt{java.util.List reasonsContainer} -
1858 \end{itemize}
1859 \textbf{Return Value}: Tidak memiliki \textit{return value}
1860
1861 \textbf{Exception}: Tidak memiliki \textit{exception}
1862
1863 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1864
1865 \end{itemize}
1866 \item \texttt{AIF302}
1867
1868 Mata kuliah ini melatih mahasiswa dalam menulis ilmiah serta memperkenalkan
1869 metodologi penelitian serta kakas untuk menulis ilmiah.
1870
1871 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1872 \begin{itemize}

```

```

1873 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1874
1875 \textbf{Parameter:}
1876 \begin{itemize}
1877 \item \texttt{Mahasiswa mahasiswa} -
1878 \item \texttt{java.util.List reasonsContainer} -
1879 \end{itemize}
1880 \textbf{Return Value:} Tidak memiliki \textit{return value}
1881
1882 \textbf{Exception:} Tidak memiliki \textit{exception}
1883
1884 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1885
1886 \end{itemize}
1887 \item \texttt{AIF181}
1888
1889
1890
1891 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF210}
1892
1893
1894
1895 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF343}
1896
1897
1898
1899 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF206}
1900
1901 Mata kuliah ini memperkenalkan kepada mahasiswa mengenai konsep sistem
1902 operasi, jenis-jenis sistem operasi yang digunakan dalam kehidupan
1903 sehari-hari dan beberapa perangkat keras yang dibutuhkan pada komputer.
1904 Selain itu juga mempelajari mengenai teknik dan algoritma yang digunakan
1905 dalam pengelolaan sistem operasi.
1906
1907 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1908 \begin{itemize}
1909 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1910
1911 \textbf{Parameter:}
1912 \begin{itemize}
1913 \item \texttt{Mahasiswa mahasiswa} -
1914 \item \texttt{java.util.List reasonsContainer} -
1915 \end{itemize}
1916 \textbf{Return Value:} Tidak memiliki \textit{return value}
1917
1918 \textbf{Exception:} Tidak memiliki \textit{exception}
1919
1920 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1921
1922 \end{itemize}
1923 \item \texttt{AIF314}
1924
1925 Kuliah ini merupakan kelanjutan dari kuliah Manajemen Informasi Basisdata.
1926 Pada perkuliahan ini, mahasiswa akan mempelajari teknik-teknik pengelolaan
1927 basis data dan membuat program dengan basis data yang optimal/efisien.
1928
1929 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1930 \begin{itemize}
1931 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1932
1933 \textbf{Parameter:}
1934 \begin{itemize}
1935 \item \texttt{Mahasiswa mahasiswa} -
1936 \item \texttt{java.util.List reasonsContainer} -
1937 \end{itemize}
1938 \textbf{Return Value:} Tidak memiliki \textit{return value}
1939
1940 \textbf{Exception:} Tidak memiliki \textit{exception}
1941
1942 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1943
1944 \end{itemize}
1945 \item \texttt{AIF380}
1946
1947
1948
1949 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF103}
1950
1951 Mata kuliah ini merupakan salah satu cara untuk mencapai kompetensi dasar
1952 tentang matematika diskrit yang prinsipnya banyak digunakan dalam bidang
1953 ilmu komputer. Selain itu, kuliah ini juga merupakan cara untuk membentuk
1954 pola pikir logis yang dibutuhkan untuk menempuh kuliah-kuliah di tingkat
1955 yang lebih tinggi.
1956
1957 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF292}
1958
1959
1960
1961 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF441}
1962
1963 Mata kuliah ini memperkenalkan kepada mahasiswa konsep jaringan lanjut
1964 terutama di layer data link dan layer network. Materi utama dari mata kuliah
1965 ini adalah pengembangan jaringan dan pengenalan fungsi-fungsi yang terdapat
1966 pada alat jaringan Cisco yang berkaitan dengan layer 2 dan layer 3.
1967
1968 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1969 \begin{itemize}
1970 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1971

```

```

1972 \textbf{Parameter:}
1973 \begin{itemize}
1974 \item \texttt{Mahasiswa mahasiswa} -
1975 \item \texttt{java.util.List reasonsContainer} -
1976 \end{itemize}
1977 \textbf{Return Value:} Tidak memiliki \textit{return value}
1978
1979 \textbf{Exception:} Tidak memiliki \textit{exception}
1980
1981 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1982
1983 \end{itemize}
1984 \item \texttt{AIF457}
1985
1986 Mata kuliah ini memperkenalkan konsep kewirausahaan dengan memanfaatkan teknologi, khususnya
1987 teknologi informasi, sebagai basis usaha dan inovasi produk/jasa; Mempelajari
1988 teknik mencari peluang dan merumuskan bidang usaha spesifik yang akan
1989 diterjuni; Mempelajari konsep manajemen pemasaran, keuangan dan SDM dalam
1990 kaitannya dengan berwira-usaha di bidang TI; Menyusun proposal bisnis untuk
1991 berwira-usaha di bidang TI dan mempresentasikannya.
1992
1993 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1994 \begin{itemize}
1995 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1996
1997 \textbf{Parameter:}
1998 \begin{itemize}
1999 \item \texttt{Mahasiswa mahasiswa} -
2000 \item \texttt{java.util.List reasonsContainer} -
2001 \end{itemize}
2002 \textbf{Return Value:} Tidak memiliki \textit{return value}
2003
2004 \textbf{Exception:} Tidak memiliki \textit{exception}
2005
2006 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
2007
2008 \end{itemize}
2009 \item \texttt{MKU001}
2010
2011 Mata Kuliah Pendidikan Pancasila berupaya menelaah/mengkaji berbagai fenomena kehidupan
2012 bangsa dan Negara Indonesia sebagai sebuah ruang publik dengan menggunakan pendekatan
2013 hermeneutika (filsafat) dan pendidikan nilai (pedagogik). Dengan bantuan hermeneutika
2014 mahasiswa diajak berpikir kritis terhadap segala bentuk ideologisme Pancasila dan melalui
2015 pendidikan nilai mahasiswa dilatih untuk memiliki nilai Pancasila. Nilai pengembangan diri
2016 intra-personal dan relasi inter-personal dapat tertanam melalui pendidikan Pancasila yang
2017 tujuannya adalah membangun kepribadian (character building) manusia Indonesia yang utuh,
2018 baik menyangkut aspek kognitif, afektif, maupun psikomotor. Dengan demikian, Pendidikan
2019 Pancasila mengajak mahasiswa menilai realitas ruang publik sehari-hari secara mandiri
2020 dengan panduan nilai-nilai etis Pancasila.
2021
2022 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{ESA101}
2023
2024
2025
2026 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AMS290}
2027
2028
2029
2030 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF461}
2031
2032
2033
2034 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF318}
2035
2036 Mata kuliah ini memperkenalkan konsep perangkat mobile dan pemrograman pada perangkat
2037 mobile. Pemrograman dikhususkan pada lingkungan J2ME dan Android.
2038 Untuk meningkatkan keterampilan pemrograman dilengkapi dengan praktikum.
2039 Sedangkan untuk mendapatkan pengalaman penerapan konsep diberikan tugas implementasi suatu
2040 kasus pada lingkungan mobile-cloud dengan kasus yang sudah ditentukan.
2041
2042 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
2043 \begin{itemize}
2044 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
2045
2046 \textbf{Parameter:}
2047 \begin{itemize}
2048 \item \texttt{Mahasiswa mahasiswa} -
2049 \item \texttt{java.util.List reasonsContainer} -
2050 \end{itemize}
2051 \textbf{Return Value:} Tidak memiliki \textit{return value}
2052
2053 \textbf{Exception:} Tidak memiliki \textit{exception}
2054
2055 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
2056
2057 \end{itemize}
2058 \item \texttt{AIF334}
2059
2060
2061
2062 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{ESM105}
2063
2064
2065
2066 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF450}
2067
2068
2069
2070 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF446}

```

2071  
2072  
2073  
2074 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF104}  
2075  
2076 Mata kuliah ini memberikan pengetahuan tentang logika yang digunakan di  
2077 dalam ilmu komputer. Dalam kuliah ini, mahasiswa belajar untuk bisa  
2078 memodelkan suatu kalimat dalam kehidupan sehari-hari, ke dalam kalimat  
2079 dengan sintaks tertentu, yang hanya memiliki satu arti. Lalu, diperkenalkan  
2080 juga, bagaimana mengartikan suatu kalimat (benar atau salah) dan bagaimana  
2081 menentukan sifat dari kalimat tersebut.  
2082  
2083 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF387}  
2084  
2085  
2086  
2087 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF313}  
2088  
2089 Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar pembuatan grafik  
2090 dengan menggunakan komputer seperti mengenal berbagai algoritma pembuatan  
2091 primitif 2 dimensi seperti titik, garis, lingkaran, elips, berbagai macam  
2092 bentuk kurva, fraktal, konsep warna (RGB), dasar-dasar grafika 3 dimensi  
2093 seperti pewarnaan, pencahayaan, pemberian tekstur pada objek, transformasi,  
2094 animasi, dan sebagainya. Selain, itu diberikan masalah-masalah komputasi  
2095 sederhana yang harus diselesaikan menggunakan konsep-konsep yang sudah  
2096 diperkenalkan dan mengimplementasikannya menggunakan bahasa pemrograman Java.  
2097  
2098 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF344}  
2099  
2100  
2101  
2102 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.  
2103 \begin{itemize}  
2104 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}  
2105  
2106 \textbf{Parameter:}  
2107 \begin{itemize}  
2108 \item \texttt{Mahasiswa mahasiswa} -  
2109 \item \texttt{java.util.List reasonsContainer} -  
2110 \end{itemize}  
2111 \textbf{Return Value:} Tidak memiliki \textit{return value}  
2112  
2113 \textbf{Exception:} Tidak memiliki \textit{exception}  
2114  
2115 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}  
2116  
2117 \end{itemize}  
2118 \item \texttt{AIF201}  
2119  
2120 Mata kuliah ini memperkenalkan prinsip-prinsip yang digunakan dalam  
2121 melakukan analisa serta desain program berorientasi objek. Di samping itu,  
2122 mahasiswa juga belajar menggunakan kaskas berupa diagram UML (Unified  
2123 Modelling Language) sehingga dapat mengkomunikasikan desain secara visual.  
2124 Mahasiswa juga akan mengenal beberapa software design pattern dari Gang of  
2125 Four. Terakhir, mahasiswa akan belajar mengenai konsep MVC  
2126 (Model-View-Controller) yang menjadi dasar dari banyak framework masa kini.  
2127 Bahasa yang digunakan adalah bahasa Java, namun diusahakan tetap umum  
2128 sehingga dapat diaplikasikan pada bahasa yang lain.  
2129  
2130 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.  
2131 \begin{itemize}  
2132 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}  
2133  
2134 \textbf{Parameter:}  
2135 \begin{itemize}  
2136 \item \texttt{Mahasiswa mahasiswa} -  
2137 \item \texttt{java.util.List reasonsContainer} -  
2138 \end{itemize}  
2139 \textbf{Return Value:} Tidak memiliki \textit{return value}  
2140  
2141 \textbf{Exception:} Tidak memiliki \textit{exception}  
2142  
2143 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}  
2144  
2145 \end{itemize}  
2146 \item \texttt{AIF352}  
2147  
2148  
2149  
2150 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF305}  
2151  
2152 Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar jaringan  
2153 komputer dengan menggunakan top-down. Selain itu mengajarkan juga kepada  
2154 mahasiswa mengenai aplikasi-aplikasi berbasis jaringan sehingga diharapkan  
2155 mahasiswa dapat membuat aplikasi berbasis jaringan dengan menggunakan socket.  
2156 Pada akhirnya, mahasiswa akan ditugaskan untuk membangun jaringan komputer  
2157 LAN, baik menggunakan kabel maupun nirkabel.  
2158  
2159 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.  
2160 \begin{itemize}  
2161 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}  
2162  
2163 \textbf{Parameter:}  
2164 \begin{itemize}  
2165 \item \texttt{Mahasiswa mahasiswa} -  
2166 \item \texttt{java.util.List reasonsContainer} -  
2167 \end{itemize}  
2168 \textbf{Return Value:} Tidak memiliki \textit{return value}  
2169

```

2170 \textbf{Exception}: Tidak memiliki \textit{exception}
2171
2172 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
2173
2174 \end{itemize}
2175 \item \texttt{MKU010}
2176
2177 Mata kuliah ini difokuskan pada pemahaman sumber referensi dalam Bahasa Inggris dan
2178 pengembangan kosakata Bahasa Inggris (vocabularies). Hampir keseluruhan waktu perkuliahan
2179 didedikasikan untuk menjelaskan metode mengekstraksi isi bacaan secara tepat dan melatih
2180 mahasiswa untuk menerapkan metode tersebut seraya menambah kosakata-kosakata baru.
2181 Mahasiswa juga dilatih untuk mempresentasikan hasil pemahamannya akan isi bahan bacaan.
2182
2183 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{MKU011}
2184
2185 Mata kuliah estetika memberi pemahaman konseptual filosofis "seni" dalam khasanah keilmuan,
2186 pembentukan kesadaran ekologis juga dalam proses pembudayaan dan peradaban. Mata kuliah ini
2187 akan menjadi fondasi bagi mahasiswa untuk memahami dan mempraktekkan seni dari sudut pandang
2188 filsafat, sejarah, kultural, dan global. Melalui mata kuliah ini, mahasiswa mempelajari
2189 mengenai dunia manusia (manusia dan pikirannya), pluralitas dan relativitas seni, serta
2190 aliran-aliran seni rupa Barat.
2191
2192 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF332}
2193
2194
2195
2196 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
2197 \begin{itemize}
2198 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
2199
2200 \textbf{Parameter}:
2201 \begin{itemize}
2202 \item \texttt{Mahasiswa mahasiswa} -
2203 \item \texttt{java.util.List reasonsContainer} -
2204 \end{itemize}
2205 \textbf{Return Value}: Tidak memiliki \textit{return value}
2206
2207 \textbf{Exception}: Tidak memiliki \textit{exception}
2208
2209 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
2210
2211 \end{itemize}
2212 \item \texttt{AIF304}
2213
2214 Mata kuliah ini memberikan kesempatan bagi mahasiswa untuk memperdalam konsep
2215 tentang pengembangan sistem informasi dan mempraktekkan analisis kebutuhan,
2216 analisis sistem dan perancangan sitem pada organisasi studi kasus;
2217
2218 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
2219 \begin{itemize}
2220 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
2221
2222 \textbf{Parameter}:
2223 \begin{itemize}
2224 \item \texttt{Mahasiswa mahasiswa} -
2225 \item \texttt{java.util.List reasonsContainer} -
2226 \end{itemize}
2227 \textbf{Return Value}: Tidak memiliki \textit{return value}
2228
2229 \textbf{Exception}: Tidak memiliki \textit{exception}
2230
2231 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
2232
2233 \end{itemize}
2234 \item \texttt{ESM201}
2235
2236
2237
2238 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF312}
2239
2240 Mata kuliah ini memberikan pengetahuan awal tentang keamanan informasi. Pada
2241 beberapa pertemuan awal, dibahas keamanan informasi secara matematis, yaitu
2242 di materi-materi seputar kriptografi dan serangannya. Lalu, dibahas pula
2243 konsep keamanan informasi pada jaringan komputer dan pada software.
2244
2245 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
2246 \begin{itemize}
2247 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
2248
2249 \textbf{Parameter}:
2250 \begin{itemize}
2251 \item \texttt{Mahasiswa mahasiswa} -
2252 \item \texttt{java.util.List reasonsContainer} -
2253 \end{itemize}
2254 \textbf{Return Value}: Tidak memiliki \textit{return value}
2255
2256 \textbf{Exception}: Tidak memiliki \textit{exception}
2257
2258 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
2259
2260 \end{itemize}
2261 \item \texttt{AIF484}
2262
2263
2264
2265 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF191}
2266
2267
2268

```

```

2269 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF386}
2270
2271
2272
2273 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF105}
2274
2275 Mata kuliah ini memperkenalkan kepada mahasiswa terminologi dan konsep dasar
2276 yang akan banyak dipakai sepanjang kuliah di Teknik Informatika. Selain itu
2277 mata kuliah ini juga mempersiapkan dan membiasakan mahasiswa dengan suasana
2278 akademik yang khas perguruan tinggi seperti kedisiplinan, kerja sama,
2279 kemampuan menggunakan teknologi informasi dalam pembuatan tugas, kemampuan
2280 komunikasi, dsb.
2281
2282 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF294}
2283
2284
2285
2286 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF282}
2287
2288
2289
2290 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF451}
2291
2292
2293
2294 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{APS302}
2295
2296
2297
2298 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{EAA101}
2299
2300
2301
2302 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{HasPrasyarat}
2303
2304 Mendefinisikan kelas-kelas yang memiliki prasyarat, terkustomisasi
2305 untuk seorang \texttt{Mahasiswa}. Jika ada tambahan, jangan lupa untuk
2306 mendaftarkannya di DEFAULT\_HASPRASYARAT\_CLASSES. Jika berubah package,
2307 jangan lupa mengupdate DEFAULT\_PRASYARAT\_PACKAGE.
2308
2309 Atribut yang dimiliki kelas ini adalah sebagai berikut.
2310 \begin{itemize}
2311 \item \texttt{String DEFAULT\_HASPRASYARAT\_CLASSES} - Daftar dari nama kelas default seluruh turunan interface ini. Perlu
    didaftarkan
2312 manual, karena Java reflection tidak dapat mendeteksi otomatis.
2313 \item \texttt{String DEFAULT\_PRASYARAT\_PACKAGE} - Package tempat menyimpan seluruh turunan standar interface ini. Perlu
    didefinisikan
2314 manual, karena Java reflection tidak dapat mendeteksi otomatis.
2315 \end{itemize}
2316 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
2317 \begin{itemize}
2318 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}Memeriksa
    prasyarat-prasyarat dari kuliah, spesifik untuk mahasiswa
2319 yang dituju. Jika ada pesan-pesan khusus, akan ditambahkan pada parameter
2320 reasonsContainer.
2321
2322 \textbf{Parameter:}
2323 \begin{itemize}
2324 \item \texttt{Mahasiswa mahasiswa} -
2325 prasyarat kuliah akan diperiksa spesifik pada mahasiswa ini
2326 \item \texttt{List reasonsContainer} -
2327 pesan-pesan terkait prasyarat akan ditambahkan di sini, jika ada.
2328 \end{itemize}
2329 \textbf{Return Value:} true jika seluruh prasyarat dipenuhi, false jika tidak.
2330
2331 \textbf{Exception:} Tidak memiliki \textit{exception}
2332
2333 \end{itemize}
2334 \item \texttt{HasPraktikum}
2335
2336
2337
2338 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{HasResponsi}
2339
2340
2341
2342 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{Kelulusan}
2343
2344
2345
2346 Atribut yang dimiliki kelas ini adalah sebagai berikut.
2347 \begin{itemize}
2348 \item \texttt{String PILIHAN\_WAJIB} -
2349 \item \texttt{String WAJIB} -
2350 \item \texttt{String AGAMA} -
2351 \item \texttt{int MIN\_SKS\_LULUS} -
2352 \item \texttt{int MIN\_PILIHAN\_WAJIB} -
2353 \end{itemize}
2354 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
2355 \begin{itemize}
2356 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
2357
2358 \textbf{Parameter:}
2359 \begin{itemize}
2360 \item \texttt{Mahasiswa mahasiswa} -
2361 \item \texttt{java.util.List reasonsContainer} -
2362 \end{itemize}
2363 \textbf{Return Value:} Tidak memiliki \textit{return value}
2364

```

```
2365 | \textbf{Exception}: Tidak memiliki \textit{exception}
2366 |
2367 | \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{Object}
2368 |
2369 | \end{itemize}
2370 | \end{enumerate}
2371 | \end{document}
```

## 1. InfoMataKuliah

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- **public int sks()**  
Jumlah bobot sks dari mata kuliah ini  
**Parameter:**
  - Tidak memiliki parameter *method***Return Value:** jumlah bobot sks  
**Exception:** Tidak memiliki *exception*
- **public String nama()**  
Nama mata kuliah ini  
**Parameter:**
  - Tidak memiliki parameter *method***Return Value:** nama mata kuliah  
**Exception:** Tidak memiliki *exception*

## 2. MataKuliahFactory

Kelas yang bertugas membuat kelas mata kuliah, dan menyimpannya untuk bisa digunakan kemudian (untuk hemat memori).

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- **String DEFAULT\_MATAKULIAH\_PACKAGE** - Lokasi package untuk daftar mata kuliah
- **MataKuliahFactory instance** - Singleton instance to factory.
- **SortedMap mataKuliahCache** - Singleton instances untuk mata kuliah.

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- **public static MataKuliahFactory getInstance()**  
**Parameter:**
  - Tidak memiliki parameter *method***Return Value:** Tidak memiliki *return value*  
**Exception:** Tidak memiliki *exception*
- **public MataKuliah createMataKuliah(java.lang.String kode, int sks, java.lang.String nama)**  
Membuat baru atau mendapatkan mata kuliah, jika memiliki informasi nama dan jumlah SKS.  
**Parameter:**



- `String kode` - kode mata kuliah
- `int sks` - jumlah SKS
- `String nama` - nama mata kuliah

**Return Value:** objek mata kuliah

**Exception:** Tidak memiliki *exception*

- `public MataKuliah createMataKuliah(java.lang.String kode)`  
Membuat baru atau mendapatkan mata kuliah, jika tidak memiliki informasi nama dan jumlah SKS.

**Parameter:**

- `String kode` - kode mata kuliah

**Return Value:** objek mata kuliah

**Exception:** `IllegalStateException` jika sks dan tidak sesuai dengan yang ada di kode

### 3. Semester

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- `Semester UNKNOWN5` -
- `Semester TRANSFER` -
- `Semester PENDEK` -
- `Semester GANJIL` -
- `Semester GENAP` -
- `int order` -

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public static Semester values()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public static Semester valueOf(java.lang.String name)`

**Parameter:**

- `String name` -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public static Semester fromString(java.lang.String text)`

**Parameter:**

- `String text` -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `int getOrder()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

#### 4. TahunSemester

Menyimpan konstanta untuk semester beserta tahunnya di UNPAR.

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- `String kodeTahunSemester` - Kode semester 3 digit, sesuai DPS: `jul`, `li`, 2 digit pertama berupa tahun, 2 digit terakhir: `li`, 1 digit terakhir: 1 untuk ganjil, 2 untuk genap, 4 untuk pendek. `i`/`ul`

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public Semester getSemester()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public int getTahun()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `private static void validateKodeSemester(java.lang.String kodeTahunSemester)`

**Parameter:**

- `String kodeTahunSemester` -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public String getKodeDPS()`  
Mendapatkan kode tahun/semester sesuai aturan di DPS.  
**Parameter:**  
– Tidak memiliki parameter *method*  
**Return Value:** kode tahun/semester sesuai aturan di DPS.  
**Exception:** Tidak memiliki *exception*
- `public int compareTo(id.ac.unpar.siamodels.TahunSemester o)`  
**Parameter:**  
– TahunSemester o -  
**Return Value:** Tidak memiliki *return value*  
**Exception:** Tidak memiliki *exception*  
**Override:** `compareTo` dari kelas `Object`
- `public boolean equals(java.lang.Object arg0)`  
**Parameter:**  
– `Object arg0` -  
**Return Value:** Tidak memiliki *return value*  
**Exception:** Tidak memiliki *exception*
- `public String toString()`  
**Parameter:**  
– Tidak memiliki parameter *method*  
**Return Value:** Tidak memiliki *return value*  
**Exception:** Tidak memiliki *exception*

## 5. Mahasiswa

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- `String npm` -
- `String nama` -
- `List riwayatNilai` -
- `URL photoURL` -
- `List jadwalKuliahList` -
- `SortedMap nilaiTOEFL` -

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public String getNama()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public void setNama(java.lang.String nama)`

**Parameter:**

- String nama -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public String getNpm()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public URL getPhotoURL()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public void setPhotoURL(java.net.URL photoURL)`

**Parameter:**

- URL photoURL -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public List getJadwalKuliahList()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public void setJadwalKuliahList(java.util.List jadwalKuliahList)`

**Parameter:**

- `java.util.List jadwalKuliahList` -
- Return Value:** Tidak memiliki *return value*
- Exception:** Tidak memiliki *exception*
- `public String getEmailAddress()`
- Parameter:**
  - Tidak memiliki parameter *method*
- Return Value:** Tidak memiliki *return value*
- Exception:** Tidak memiliki *exception*
- `public List getRiwayatNilai()`
- Parameter:**
  - Tidak memiliki parameter *method*
- Return Value:** Tidak memiliki *return value*
- Exception:** Tidak memiliki *exception*
- `public SortedMap getNilaiTOEFL()`
- Parameter:**
  - Tidak memiliki parameter *method*
- Return Value:** Tidak memiliki *return value*
- Exception:** Tidak memiliki *exception*
- `public void setNilaiTOEFL(java.util.SortedMap nilaiTOEFL)`
- Parameter:**
  - `java.util.SortedMap nilaiTOEFL` -
- Return Value:** Tidak memiliki *return value*
- Exception:** Tidak memiliki *exception*
- `public double calculateIPKLulus()`  
Menghitung IPK mahasiswa sampai saat ini, dengan aturan:  $\frac{\sum \text{Nilai Kuliah yang tidak lulus}}{\text{Jumlah Kuliah yang tidak lulus}}$  tidak dihitung. Jika pengambilan beberapa kali, diambil nilai terbaik. Sebelum memanggil method ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah!
- Parameter:**
  - Tidak memiliki parameter *method*
- Return Value:** IPK Lulus, atau DoubleNaN jika belum mengambil satu kuliah pun.
- Exception:** Tidak memiliki *exception*

- **public double calculateIPLulus()**  
Menghitung IP mahasiswa sampai saat ini, dengan aturan:  $\sum_i \text{li}_i$  Kuliah yang tidak lulus tidak dihitung  $\text{li}_i$ . Jika pengambilan beberapa kali, diambil  $\max_i \text{nilai}_i$ .  $\sum_i$  Sebelum memanggil method ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah!  
**Parameter:**
  - Tidak memiliki parameter *method***Return Value:** IPK Lulus, atau DoubleNaN jika belum mengambil satu kuliah pun.  
**Exception:** Tidak memiliki *exception*
- **public double calculateIPTempuh(boolean lulusSaja)**  
Menghitung IP mahasiswa sampai saat ini, dengan aturan:  $\sum_i \text{li}_i$  Perhitungan kuliah yang tidak lulus ditentukan parameter  $\text{li}_i$ . Jika pengambilan beberapa kali, diambil  $\max_i \text{nilai}_i$ .  $\sum_i$   
**Parameter:**
  - **boolean lulusSaja** - set true jika ingin membuang mata kuliah tidak lulus, false jika ingin semua (sama dengan "IP N. Terbaik" di DPS) Sebelum memanggil method ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah!**Return Value:** IPK Lulus, atau DoubleNaN jika belum mengambil satu kuliah pun.  
**Exception:** Tidak memiliki *exception*
- **public double calculateIPKumulatif()**  
Menghitung IP Kumulatif mahasiswa sampai saat ini, dengan aturan:  $\sum_i \text{li}_i$ . Jika pengambilan beberapa kali, diambil semua.  $\sum_i$  Sebelum memanggil method ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah!  
**Parameter:**
  - Tidak memiliki parameter *method***Return Value:** IPK Lulus, atau DoubleNaN jika belum mengambil satu kuliah pun.  
**Exception:** Tidak memiliki *exception*
- **public double calculateIPKTempuh(boolean lulusSaja)**  
Menghitung IPK mahasiswa sampai saat ini, dengan aturan:  $\sum_i \text{li}_i$  Perhitungan kuliah yang tidak lulus ditentukan parameter  $\text{li}_i$ . Jika pengambilan beberapa kali, diambil  $\max_i \text{nilai}_i$ .  $\sum_i$   
**Parameter:**
  - **boolean lulusSaja** - set true jika ingin membuang mata kuliah tidak lulus Sebelum memanggil method ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah!

**Return Value:** IPK Lulus, atau DoubleNaN jika belum mengambil satu kuliah pun.

**Exception:** Tidak memiliki *exception*

- `public double calculateIPS()`  
Menghitung IPS semester terakhir sampai saat ini, dengan aturan:  $\sum_{i=1}^n \text{Kuliah yang tidak lulus} \cdot \text{dihitung} / \text{em} \cdot \sum_{i=1}^n \text{Sebelum}$  memanggil method ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah!

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** nilai IPS sampai saat ini

**Exception:** `ArrayIndexOutOfBoundsException` jika belum ada nilai satupun

- `public int calculateSKSLulus()`  
Menghitung jumlah SKS lulus mahasiswa saat ini. Sebelum memanggil method ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah!

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** SKS Lulus

**Exception:** Tidak memiliki *exception*

- `public int calculateSKSTempuh(boolean lulusSaja)`  
Menghitung jumlah SKS tempuh mahasiswa saat ini. Sebelum memanggil method ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah!

**Parameter:**

- `boolean lulusSaja` - set true jika ingin membuang SKS tidak lulus

**Return Value:** SKS tempuh

**Exception:** Tidak memiliki *exception*

- `public Set calculateTahunSemesterAktif()`  
Mendapatkan seluruh tahun semester di mana mahasiswa ini tercatat sebagai mahasiswa aktif, dengan strategi memeriksa riwayat nilainya. Jika ada satu nilai saja pada sebuah tahun semester, maka dianggap aktif pada semester tersebut.

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** kumpulan tahun semester di mana mahasiswa ini aktif

**Exception:** Tidak memiliki *exception*

- `public boolean hasLulusKuliah(java.lang.String kodeMataKuliah)`  
Memeriksa apakah mahasiswa ini sudah lulus mata kuliah tertentu. Kompleksitas  $O(n)$ . Sebelum memanggil method ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah! Note: jika yang dimiliki adalah `MataKuliah`, gunakanlah `MataKuliahgetKode()`.

**Parameter:**

- `String kodeMataKuliah` - kode mata kuliah yang ingin diperiksa kelulusannya.

**Return Value:** true jika sudah pernah mengambil dan lulus, false jika belum

**Exception:** Tidak memiliki *exception*

- `public boolean hasTempuhKuliah(java.lang.String kodeMataKuliah)`  
Memeriksa apakah mahasiswa ini sudah pernah menempuh mata kuliah tertentu. Kompleksitas  $O(n)$ . Sebelum memanggil method ini, `getRiwayatNilai()` harus sudah ada isinya! Note: jika yang dimiliki adalah `MataKuliah`, gunakanlah `MataKuliahgetKode()`.

**Parameter:**

- `String kodeMataKuliah` - kode mata kuliah yang ingin diperiksa.

**Return Value:** true jika sudah pernah mengambil, false jika belum

**Exception:** Tidak memiliki *exception*

- `public int getTahunAngkatan()`  
Mendapatkan tahun angkatan mahasiswa ini, berdasarkan NPM nya

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** tahun angkatan

**Exception:** Tidak memiliki *exception*

- `public String toString()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

## 6. Mahasiswa.Nilai

Merepresentasikan nilai yang ada di riwayat nilai mahasiswa

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- `TahunSemester tahunSemester` - Tahun dan Semester kuliah ini diambil
- `MataKuliah mataKuliah` - Mata kuliah yang diambil
- `Character kelas` - Kelas kuliah



- Double nilaiART - Nilai ART
- Double nilaiUTS - Nilai UTS
- Double nilaiUAS - Nilai UAS
- Character nilaiAkhir - Nilai Akhir

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- public MataKuliah getMataKuliah()

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- public Character getKelas()

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- public Double getNilaiART()

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- public Double getNilaiUTS()

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- public Double getNilaiUAS()

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- public Character getNilaiAkhir()  
Mengembalikan nilai akhir dalam bentuk huruf (A, B, C, D, ..., atau K)

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** nilai akhir dalam huruf, atau null jika tidak ada.

**Exception:** Tidak memiliki *exception*

- `public Double getAngkaAkhir()`  
Mendapatkan nilai akhir dalam bentuk angka

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** nilai akhir dalam angka, atau null jika `getNilaiAkhir()` mengembalikan 'K' atau null

**Exception:** Tidak memiliki *exception*

- `public TahunSemester getTahunSemester()`

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public int getTahunAjaran()`

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public Semester getSemester()`

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public String toString()`

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

#### 7. `Mahasiswa.Nilai.ChronologicalComparator`

Pembandingan antara satu nilai dengan nilai lainnya, secara kronologis waktu pengambilan.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public int compare(id.ac.unpar.siamodels.Mahasiswa.Nilai o1, id.ac.unpar.siamodels.Mahasiswa.Nilai o2)`

**Parameter:**

- Mahasiswa.Nilai o1 -
- Mahasiswa.Nilai o2 -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `compare` dari kelas `Object`

## 8. JadwalKuliah

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- `MataKuliah mataKuliah` -
- `Character kelas` -
- `DayOfWeek hari` -
- `LocalTime waktuMulai` -
- `LocalTime waktuSelesai` -
- `String lokasi` -
- `Dosen pengajar` -

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public MataKuliah getMataKuliah()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public void setMataKuliah(id.ac.unpar.siamodels.MataKuliah mataKuliah)`

**Parameter:**

- MataKuliah mataKuliah -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public Character getKelas()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public void setKelas(java.lang.Character kelas)`

**Parameter:**

- Character kelas -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public DayOfWeek getHari()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public void setHari(java.time.DayOfWeek hari)`

**Parameter:**

- DayOfWeek hari -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public LocalTime getWaktuMulai()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public void setWaktuMulai(java.time.LocalTime waktuMulai)`

**Parameter:**

- LocalTime waktuMulai -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public LocalTime getWaktuSelesai()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public void setWaktuSelesai(java.time.LocalDateTime waktuSelesai)`

**Parameter:**

– `LocalTime waktuSelesai` -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public String getLokasi()`

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public void setLokasi(java.lang.String lokasi)`

**Parameter:**

– `String lokasi` -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public Dosen getPengajar()`

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public void setPengajar(id.ac.unpar.siamodels.Dosen pengajar)`

**Parameter:**

– `Dosen pengajar` -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public String getWaktuString()`

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public static DayOfWeek indonesianToDayOfWeek(java.lang.String indonesian)`

Converts Indonesian day names to `DayOfWeek` enumeration.

**Parameter:**

– `String indonesian` - the day name in Indonesian

**Return Value:** `DayOfWeek` object or null if not found.

**Exception:** Tidak memiliki *exception*

## 9. MataKuliah

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- `String kode` -
- `String nama` -
- `Integer sks` -

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public String getKode()`

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public String getNama()`

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public Integer getSks()`

**Parameter:**

– Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

## 10. Dosen

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- `String nik` -
- `String nama` -

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public String getNik()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public void setNik(java.lang.String nik)`

**Parameter:**

- String `nik` -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public String getNama()`

**Parameter:**

- Tidak memiliki parameter *method*

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public void setNama(java.lang.String nama)`

**Parameter:**

- String `nama` -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

- `public boolean equals(java.lang.Object arg0)`

**Parameter:**

- Object `arg0` -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

#### 11. MKU008

Mendalami perilaku sehari-hari yang baik dalam bermasyarakat.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

#### 12. IIE210

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

13. AIF203

Mata kuliah ini memperkenalkan kepada mahasiswa konsep struktur diskret yang digunakan pada bidang informatika diantaranya graph, pohon dan finite state machine

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas MataKuliah

14. AIF311

Kuliah Pemrograman Fungsional bertujuan untuk: 1. memperkenalkan paradigma pemrograman fungsional, yaitu sebuah pemrograman yang didasarkan pada konsep pemetaan dan fungsi matematika. Penyelesaian suatu masalah didasari atas aplikasi dari fungsi-fungsi tersebut. 2. memberikan dasar-dasar pemrograman fungsional dengan menggunakan bahasa fungsional Haskell.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas MataKuliah

15. AIF192

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

16. AIF468

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.



17. IIE103

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

18. AIF385

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

19. AIF106

Mata kuliah ini memberikan pengetahuan tentang cara kerja komputer, dimulai dari representasi data dan berbagai macam operasinya. Selanjutnya, juga diperkenalkan bagaimana merepresentasikan suatu fungsi dalam rangkaian gerbang logika, dan bagaimana menyederhanakannya. Berbagai rangkaian dasar yang digunakan di dalam komputer juga diperkenalkan. Mahasiswa juga akan mempelajari komponen komputer, misalnya register dan memori.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

20. AIF281

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

21. EAA102

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

22. AIF405

Mata kuliah ini merupakan lanjutan dari Projek Sistem Informasi 1 dan memberikan kesempatan bagi mahasiswa untuk melanjutkan/mengembangkan perancangan sistem pada organisasi studi kasus, mengimplementasikan rancangan dan melakukan pengujian perangkat lunak;

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas `MataKuliah`

23. APS182

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

24. MKU004

Fenomenologi Agama merupakan bagian yang tak terpisahkan dari kajian filosofis, kritis, rasional, dan obyektif mengenai substansi ajaran agama. Fenomenologi merupakan sebuah disiplin ilmu yang secara kritis-rasional mengkaji fenomena dan dinamika kehidupan manusia beragama, dari upaya menjadikan Tuhan sebagai tujuan sesembahan sampai menempatkan Tuhan sebagai instrumen legitimasi untuk melakukan tindakan yang justru bertolak belakang dengan kehendak Tuhan yang disembah. Sehubungan dengan itu, kritik konstruktif terhadap perilaku manusia beragama menjadi salah satu poin utama dalam mata kuliah ini. Kesiapan untuk melakukan otoritik terhadap agama sendiri erat terkait dengan upaya menemukan kembali nilai sejati agama atau otentisitas hidup beragama.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

25. MKU012

Perkuliahan logika ditujukan untuk memberikan dasar-dasar ketrampilan berpikir rasional dan sistematis. Isinya mencakup ketrampilan berpikir deduktif dan induktif, seperti silogisme, argumen analogikal dan generalisasi induktif. Pembahasan teoretis disertai pula dengan pelatihan praktis yang diarahkan pada proses berpikir. Untuk menajamkan kemampuan berpikir tersebut, mahasiswa dilatih pula mengidentifikasi kerancuan-kerancuan (fallacies) yang sering dijumpai baik dalam kehidupan sehari-hari maupun dalam konteks akademik.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

26. AIF389

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

27. AMS190

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

28. AMS191

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

29. AMS200

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

30. AIF330

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

31. AIF388

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

32. AIF465

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

33. AIF453

Mata kuliah ini memperkenalkan kebutuhan organisasi terhadap sistem business intelligent (BI) dan pemanfaatan BI untuk organisasi; memperkenalkan konsep sistem business intelligent dan komponennya; Mempelajari teknik-teknik analisis data bisnis dan visualisasi hasil analisis; Mempelajari konsep data warehouse dan perancangannya dan fungsi OLAP; Mempraktekan teknik-teknik analisis data dan visualisasi hasil analisis.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas `MataKuliah`

34. AIF280

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

35. AIF445

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas MataKuliah

#### 36. AIF469

Mata kuliah ini mengajarkan kepada mahasiswa teknik-teknik untuk membuat layanan berbasis web. Mahasiswa diperkenalkan dengan standar-standar seperti HTTP, XML, JSON dan diajarkan untuk memanfaatkannya dalam membuat maupun menggunakan layanan pihak ketiga. Dalam kuliah ini, juga akan diperkenalkan minimal satu layanan pihak ketiga yang dapat dimanfaatkan mahasiswa, seperti Google Places Web Service.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas MataKuliah

#### 37. AIF486

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

#### 38. IIE207

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

#### 39. AIF202

Mata kuliah ini memperkenalkan kepada mahasiswa beberapa algoritma dan struktur data, alternatif cara implementasinya, dan analisis kompleksitas waktunya. Mahasiswa diberikan beberapa masalah komputasi yang harus diselesaikan dengan menggunakan algoritma atau struktur data yang sudah diperkenalkan dan mengimplementasikannya dalam bahasa pemrograman Java.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas MataKuliah

40. AIF347

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

41. MKU009

Tujuan dari mata kuliah ini adalah untuk mendalami keterampilan berbahasa Indonesia, agar mampu mengkomunikasikan hasil pemikiran serta meningkatkan keterampilan dalam menyusun karya ilmiah. Mata kuliah Bahasa Indonesia ini dimulai dengan mempelajari penulisan kata baku dan non baku serta pengungkapan pikiran dengan punctuation yang benar. Selanjutnya dipelajari penyusunan kalimat yang baku serta menghubungkan kalimat-kalimat yang padu dalam menuangkan gagasan dalam sebuah paragraf. Selain itu, dalam matakuliah ini dipelajari cara menyusun surat dinas yang jelas dan komunikatif. Di akhir kuliah ini, mahasiswa diberi tugas penyusunan makalah dengan benar.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

42. ESM203

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

43. APS402

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas MataKuliah

44. AIF306

Mata kuliah ini bertujuan untuk memberikan pengalaman bagi mahasiswa dalam mengerjakan proyek dengan teknologi-teknologi terkini, secara berkelompok. Teknologi-teknologi yang digunakan pada kuliah ini tidak spesifik dan dapat berubah seiring perkembangan teknologi maupun disesuaikan dengan kompetensi dosen pengajar. Beberapa teknologi yang dapat dimanfaatkan antara lain: DVCS tool menggunakan Git + Github, Mobile native app (Android, iOS, dll), dan responsive web design.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas MataKuliah

45. AKS122

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

46. MKU002

Pendidikan Kewarganegaraan menjelaskan pentingnya pemahaman tentang identitas nasional Indonesia, hak dan kewajiban warga negara Indonesia serta hubungannya dengan hak dan kewajiban asasi manusia. Materi kuliah mencakup juga wawasan nusantara, ketahanan nasional, politik dan strategi nasional, serta implementasinya dalam kehidupan bermasyarakat, berbangsa dan bernegara kesatuan Republik Indonesia.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

47. AIF462

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

48. AIF360

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas MataKuliah

49. AIF337

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

50. AIF458

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas MataKuliah

51. AIF301

Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar tentang \*sistem cerdas dan komponen-komponennya. " "Terdapat 4 topik utama yang dibahas yaitu teknik pencarian untuk \*penyelesaian masalah, representasi pengetahuan dalam sistem \*cerdas, pemodelan ketidakpastian dalam masalah dan teknik pembelajaran mesin.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -

– java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas MataKuliah

52. AIF182

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

53. ESM204

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

54. AIF205

Mata kuliah ini memperkenalkan kepada mahasiswa arsitektur komputer sederhana, modern, dan Advance. Perbedaan, kelebihan dan kekurangan untuk masing-masing arsitektur. Selain itu mahasiswa juga mempelajari cara kerja dari komponen-komponen komputer, terutama memory, cache, system BUS dan input/output.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas MataKuliah

55. AIF317

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -



**Return Value:** Tidak memiliki *return value*  
**Exception:** Tidak memiliki *exception*  
**Override:** `checkPrasyarat` dari kelas `MataKuliah`

56. AIF383

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

57. AIF442

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas `MataKuliah`

58. ESM101

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

59. AIF403

1. Memberikan wawasan kepada mahasiswa tentang kemunculan dan pemanfaatan teknologi baru, khususnya yang berkaitan dengan komputer, dan dampaknya terhadap masyarakat luas. 2. Memberikan kesadaran dan panduan bersikap kepada mahasiswa dalam menghadapi gejala yang disebabkan oleh munculnya teknologi baru, khususnya yang berkaitan dengan komputer.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*  
**Exception:** Tidak memiliki *exception*  
**Override:** `checkPrasyarat` dari kelas `MataKuliah`

60. AIF402

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*  
**Exception:** Tidak memiliki *exception*  
**Override:** `checkPrasyarat` dari kelas `MataKuliah`

61. AIF455

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

62. AIF443

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

63. AIF101

Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar pemrograman seperti pengulangan dan percabangan, konsep dasar penyimpanan data kontigu menggunakan array, konsep dasar pemrograman berorientasi objek seperti kelas & objek, method, dll, termasuk di dalamnya 4 prinsip dasar pemrograman berorientasi objek : data abstraction, encapsulation, inheritance dan polymorphism. Selain, itu diberikan masalah-masalah komputasi sederhana yang harus diselesaikan menggunakan konsep-konsep yang sudah diperkenalkan dan mengimplementasikannya menggunakan bahasa pemrograman Java

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

64. AIF382

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

65. AIF480

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

66. AIF316

Mata kuliah ini memperkenalkan konsep-konsep dasar komputasi paralel, dimana sebuah program yang berjalan secara paralel harus memiliki safety property dan liveness property. Mahasiswa dikenalkan dengan beberapa teknik pemrograman multi-thread seperti lock, monitor, barrier, thread pool, dan sebagainya, yang diimplementasikan dalam bahasa pemrograman Java. Mahasiswa juga dikenalkan dengan beberapa metode untuk menganalisis kebenaran program baik secara matematis maupun secara praktis dengan bantuan model checker.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas MataKuliah

67. AIF438

Mata kuliah ini: Memperkenalkan karakteristik dan teknik visualisasi dari berbagai jenis data yang dapat dianalisis dengan teknik-teknik data mining; mempelajari teknik-teknik penyiapan data untuk berbagai jenis data dan teknik data mining; mempraktekkan teknik-teknik penyiapan data untuk menganalisis data nyata/simulasi dengan memanfaatkan perangkat lunak aplikasi.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas MataKuliah

68. AIF204

Mata kuliah ini memperkenalkan konsep dan arsitektur DBMS, mengajarkan aljabar relasional dan SQL serta pemanfaatannya pada pemrograman kueri sederhana s/d relatif kompleks. Selain itu, mata kuliah ini juga mengajarkan dan mempraktekkan perancangan basisdata untuk masalah sederhana (lingkup kecil) termasuk pengembangan program aplikasinya;

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas MataKuliah

69. AIF341

Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar jaringan dan aplikasinya di kehidupan sehari-hari. Mahasiswa dikenalkan dengan teknologi-teknologi terbaru di bidang jaringan, sehingga mahasiswa memiliki pengetahuan yang dapat digunakan dalam kehidupan sehari-hari. Selain itu mahasiswa juga diperkenalkan dengan NetAcad, sebuah layanan dari Cisco yang dapat digunakan untuk memenuhi segala macam kebutuhan terkait dengan Cisco Academy.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

70. AIF183

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

71. AKS124

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

72. AIF459

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

73. AIF336

Mata kuliah ini merupakan mata kuliah lanjutan dari mata kuliah Keamanan Informasi, dengan titik berat pada materi kriptografi. Mata kuliah

ini memperkenalkan tambahan konsep kriptografi, misalnya tentang otentikasi yaitu otentikasi entitas, manajemen kunci, dan bentuk lain dari metode merahasiakan pesan, yaitu dengan menggunakan secret sharing. Selanjutnya, diperkenalkan juga penggunaan kriptografi pada protokol-protokol yang sebenarnya banyak digunakan sehari-hari, misalnya pada e-cash, auction, dan electronic voting.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

#### 74. AIF463

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

#### 75. AIF208

Mata kuliah ini memperkenalkan kepada mahasiswa tahapan rekayasa perangkat lunak, terutama dengan paradigma berorientasi objek, dilengkapi dengan pengenalan tentang manajemen proyek perangkat lunak. Selain, itu diberikan deskripsi proyek berskala kecil yang harus dikerjakan oleh mahasiswa dalam kelompok dengan menerapkan teori yang telah dipelajarinya.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

##### **Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas `MataKuliah`

#### 76. MKU003

Mata kuliah ini membentuk karakteristik mahasiswa sebagai manusia yang memiliki religiusitas melalui pendalaman akan makna agama dan beragama, mendeteksi dinamika Wahyu Tuhan dan iman mereka, memahami relasi dengan Tuhan dan sesama, mengenal makna keselamatan dalam konteks Kerajaan Allah, dan mampu menyatakan ajaran Gereja dalam pelayanan terhadap orang miskin dan terlantar.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

#### 77. APS309

APS302 atau APS309 ?

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

78. AIF335

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

79. AIF362

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas MataKuliah

80. AIF460

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

81. AIF358

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

82. AMS100

Sistem Bilangan, Fungsi, Limit dan Kekontinuan Fungsi, Turunan, Integral, Penggunaan Integral, Sistem Persamaan Linear, Determinan, Vektor, Nilai dan Vektor Eigen.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

83. AIF401

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*  
**Exception:** Tidak memiliki *exception*  
**Override:** `checkPrasyarat` dari kelas `MataKuliah`

84. AIF456

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

85. SIR104

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

86. AIF339

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- `java.util.List reasonsContainer` -

**Return Value:** Tidak memiliki *return value*  
**Exception:** Tidak memiliki *exception*  
**Override:** `checkPrasyarat` dari kelas `MataKuliah`

87. AIF102

Mata kuliah ini memperkenalkan berbagai algoritma dan teknik-teknik penyelesaian masalah komputasi seperti rekursif, sorting, teknik divide dan conquer, serta exhaustive search. Selain itu, pada kuliah ini juga dikenalkan berbagai struktur data yang dapat digunakan untuk mendukung penyelesaian masalah komputasi seperti ADT List, Stack dan Queue. Baik algoritma maupun struktur data yang dikenalkan harus dapat diimplementasikan dan digunakan oleh mahasiswa untuk menyelesaikan masalah dengan menggunakan suatu bahasa pemrograman berorientasi objek.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -

– java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas MataKuliah

88. AIF381

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

89. AIF483

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

90. AIF315

Mata kuliah ini memperkenalkan konsep dan lingkungan pemrograman berbasis web, kemudian belajar membuat aplikasi berbasis web menggunakan HTML5, CSS, Java Script dan PHP. Untuk meningkatkan keterampilan pemrograman dilengkapi dengan praktikum. Sedangkan untuk mendapatkan pengalaman penerapan konsep diberikan tugas besar membuat program berbasis web dengan kasus yang ditentukan oleh mahasiswa.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas MataKuliah

91. AIF342

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -



**Return Value:** Tidak memiliki *return value*  
**Exception:** Tidak memiliki *exception*  
**Override:** `checkPrasyarat` dari kelas `MataKuliah`

92. IIE214

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

93. AIF303

Mempelajari Konsep Data, Informasi, Pengetahuan, Sistem Informasi, proses dan pemodelan bisnis, jenis-jenis sistem informasi, untuk mendukung pengambilan keputusan. Mempelajari trend Teknologi Informasi, tahap-tahap pembangunan sistem informasi. Mempelajari pengantar : EIS, e-bisnis/e-commerce, Business Intelligence, Cloud Computing dan Mobile Applications

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*  
**Exception:** Tidak memiliki *exception*  
**Override:** `checkPrasyarat` dari kelas `MataKuliah`

94. AIF302

Mata kuliah ini melatih mahasiswa dalam menulis ilmiah serta memperkenalkan metodologi penelitian serta kakas untuk menulis ilmiah.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*  
**Exception:** Tidak memiliki *exception*  
**Override:** `checkPrasyarat` dari kelas `MataKuliah`

95. AIF181

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

96. AIF210

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

97. AIF343

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

98. AIF206

Mata kuliah ini memperkenalkan kepada mahasiswa mengenai konsep sistem operasi, jenis-jenis sistem operasi yang digunakan dalam kehidupan sehari-hari dan beberapa perangkat keras yang dibutuhkan pada komputer. Selain itu juga mempelajari mengenai teknik dan algoritma yang digunakan dalam pengelolaan sistem operasi.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas `MataKuliah`

99. AIF314

Kuliah ini merupakan kelanjutan dari kuliah Manajemen Informasi Basis-data. Pada perkuliahan ini, mahasiswa akan mempelajari teknik-teknik pengelolaan basis data dan membuat program dengan basis data yang optimal/efisien.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -

– `java.util.List reasonsContainer` -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas `MataKuliah`

100. AIF380

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

101. AIF103

Mata kuliah ini merupakan salah satu cara untuk mencapai kompetensi dasar tentang matematika diskrit yang prinsipnya banyak digunakan dalam bidang ilmu komputer. Selain itu, kuliah ini juga merupakan cara untuk membentuk pola pikir logis yang dibutuhkan untuk menempuh kuliah-kuliah di tingkat yang lebih tinggi.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

102. AIF292

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

103. AIF441

Mata kuliah ini memperkenalkan kepada mahasiswa konsep jaringan lanjut terutama di layer data link dan layer network. Materi utama dari mata kuliah ini adalah pengembangan jaringan dan pengenalan fungsi-fungsi yang terdapat pada alat jaringan Cisco yang berkaitan dengan layer 2 dan layer 3.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa `mahasiswa` -
- `java.util.List reasonsContainer` -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas `MataKuliah`

104. AIF457

Mata kuliah ini memperkenalkan konsep kewirausahaan dengan memanfaatkan teknologi, khususnya teknologi informasi, sebagai basis usaha dan inovasi produk/jasa; Mempelajari teknik mencari peluang dan merumuskan

bidang usaha spesifik yang akan diterjuni; Mempelajari konsep manajemen pemasaran, keuangan dan SDM dalam kaitannya dengan berwira-usaha di bidang TI; Menyusun proposal bisnis untuk berwira-usaha di bidang TI dan mempresentasikannya.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas MataKuliah

105. MKU001

Mata Kuliah Pendidikan Pancasila berupaya menelaah/mengkaji berbagai fenomena kehidupan bangsa dan Negara Indonesia sebagai sebuah ruang publik dengan menggunakan pendekatan hermeneutika (filsafat) dan pendidikan nilai (pedagogik). Dengan bantuan hermeneutika mahasiswa diajak berpikir kritis terhadap segala bentuk ideologisme Pancasila dan melalui pendidikan nilai mahasiswa dilatih untuk memiliki nilai Pancasila. Nilai pengembangan diri intra-personal dan relasi interpersonal dapat tertanam melalui pendidikan Pancasila yang tujuannya adalah membangun kepribadian (character building) manusia Indonesia yang utuh, baik menyangkut aspek kognitif, afektif, maupun psikomotor. Dengan demikian, Pendidikan Pancasila mengajak mahasiswa menilai realitas ruang publik sehari-hari secara mandiri dengan panduan nilai-nilai etis Pancasila.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

106. ESA101

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

107. AMS290

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

108. AIF461

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

109. AIF318

Mata kuliah ini memperkenalkan konsep perangkat mobile dan pemrograman pada perangkat mobile. Pemrograman dikhususkan pada lingkungan J2ME dan Android. Untuk meningkatkan keterampilan pemrograman dilengkapi dengan praktikum. Sedangkan untuk mendapatkan pengalaman penerapan konsep diberikan tugas implementasi suatu kasus pada lingkungan mobile-cloud dengan kasus yang sudah ditentukan.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas MataKuliah

110. AIF334

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

111. ESM105

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

112. AIF450

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

113. AIF446

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

114. AIF104

Mata kuliah ini memberikan pengetahuan tentang logika yang digunakan di dalam ilmu komputer. Dalam kuliah ini, mahasiswa belajar untuk bisa memodelkan suatu kalimat dalam kehidupan sehari-hari, ke dalam kalimat dengan sintaks tertentu, yang hanya memiliki satu arti. Lalu, diperkenalkan juga, bagaimana mengartikan suatu kalimat (benar atau salah) dan bagaimana menentukan sifat dari kalimat tersebut.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

115. AIF387

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

116. AIF313

Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar pembuatan grafik dengan menggunakan komputer seperti mengenal berbagai algoritma pembuatan primitif 2 dimensi seperti titik, garis, lingkaran, elips, berbagai macam bentuk kurva, fraktal, konsep warna (RGB), dasar-dasar grafika 3 dimensi seperti pewarnaan, pencahayaan, pemberian tekstur pada objek, transformasi, animasi, dan sebagainya. Selain itu diberikan masalah-masalah komputasi sederhana yang harus diselesaikan menggunakan konsep-konsep yang sudah diperkenalkan dan mengimplementasikannya menggunakan bahasa pemrograman Java.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

117. AIF344

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas `MataKuliah`

118. AIF201

Mata kuliah ini memperkenalkan prinsip-prinsip yang digunakan dalam melakukan analisa serta desain program berorientasi objek. Di samping itu, mahasiswa juga belajar menggunakan kakas berupa diagram UML (Unified Modelling Language) sehingga dapat mengkomunikasikan desain secara visual. Mahasiswa juga akan mengenal beberapa software design pattern dari Gang of Four. Terakhir, mahasiswa akan belajar mengenai konsep MVC (Model-View-Controller) yang menjadi dasar dari banyak framework masa kini. Bahasa yang digunakan adalah bahasa Java, namun diusahakan tetap umum sehingga dapat diaplikasikan pada bahasa yang lain.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas MataKuliah

119. AIF352

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

120. AIF305

Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar jaringan komputer dengan menggunakan top-down. Selain itu mengajarkan juga kepada mahasiswa mengenai aplikasi-aplikasi berbasis jaringan sehingga diharapkan mahasiswa dapat membuat aplikasi berbasis jaringan dengan menggunakan socket. Pada akhirnya, mahasiswa akan ditugaskan untuk membangun jaringan komputer LAN, baik menggunakan kabel maupun nirkabel.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas MataKuliah

121. MKU010

Mata kuliah ini difokuskan pada pemahaman sumber referensi dalam Bahasa Inggris dan pengembangan kosakata Bahasa Inggris (vocabularies). Hampir keseluruhan waktu perkuliahan didedikasikan untuk menjelaskan metode mengekstraksi isi bacaan secara tepat dan melatih mahasiswa untuk menerapkan metode tersebut seraya menambah kosakata-kosakata baru. Mahasiswa juga dilatih untuk mempresentasikan hasil pemahamannya akan isi bahan bacaan.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

122. MKU011

Mata kuliah estetika memberi pemahaman konseptual filosofis "seni" dalam khasanah keilmuan, pembentukan kesadaran ekologis juga dalam proses pembudayaan dan peradaban. Mata kuliah ini akan menjadi fondasi bagi mahasiswa untuk memahami dan mempraktekkan seni dari sudut pandang filsafat, sejarah, kultural, dan global. Melalui mata kuliah ini, mahasiswa mempelajari mengenai dunia manusia (manusia dan pikirannya), pluralitas dan relativitas seni, serta aliran-aliran seni rupa Barat.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

123. AIF332

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas MataKuliah

124. AIF304

Mata kuliah ini memberikan kesempatan bagi mahasiswa untuk memperdalam konsep tentang pengembangan sistem informasi dan mempraktekkan analisis kebutuhan, analisis sistem dan perancangan sistem pada organisasi studi kasus;

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas MataKuliah



125. ESM201

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

126. AIF312

Mata kuliah ini memberikan pengetahuan awal tentang keamanan informasi. Pada beberapa pertemuan awal, dibahas keamanan informasi secara matematis, yaitu di materi-materi seputar kriptografi dan serangannya. Lalu, dibahas pula konsep keamanan informasi pada jaringan komputer dan pada software.

Kelas ini tidak memiliki atribut. *Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- `public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** `checkPrasyarat` dari kelas `MataKuliah`

127. AIF484

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

128. AIF191

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

129. AIF386

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

130. AIF105

Mata kuliah ini memperkenalkan kepada mahasiswa terminologi dan konsep dasar yang akan banyak dipakai sepanjang kuliah di Teknik Informatika. Selain itu mata kuliah ini juga mempersiapkan dan membiasakan mahasiswa dengan suasana akademik yang khas perguruan tinggi seperti kedisiplinan, kerja sama, kemampuan menggunakan teknologi informasi dalam pembuatan tugas, kemampuan komunikasi, dsb.

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

131. AIF294

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

132. AIF282

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

133. AIF451

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

134. APS302

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

135. EAA101

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

136. **HasPrasyarat**

Mendefinisikan kelas-kelas yang memiliki prasyarat, terkustomisasi untuk seorang **Mahasiswa**. Jika ada tambahan, jangan lupa untuk mendaftarkannya di `DEFAULT_HASPRASYARAT_CLASSES`. Jika berubah package, jangan lupa mengupdate `DEFAULT_PRASYARAT_PACKAGE`.

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- **String** `DEFAULT_HASPRASYARAT_CLASSES` - Daftar dari nama kelas default seluruh turunan interface ini. Perlu didaftarkan manual, karena Java reflection tidak dapat mendeteksi otomatis.
- **String** `DEFAULT_PRASYARAT_PACKAGE` - Package tempat menyimpan seluruh turunan standar interface ini. Perlu didefinisikan manual, karena Java reflection tidak dapat mendeteksi otomatis.

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- **public boolean** `checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)`  
Memeriksa prasyarat-prasyarat dari kuliah, spesifik untuk mahasiswa yang dituju. Jika ada pesan-pesan khusus, akan ditambahkan pada parameter `reasonsContainer`.

**Parameter:**

- **Mahasiswa mahasiswa** - prasyarat kuliah akan diperiksa spesifik pada mahasiswa ini
- **List reasonsContainer** - pesan-pesan terkait prasyarat akan ditambahkan di sini, jika ada.

**Return Value:** true jika seluruh prasyarat dipenuhi, false jika tidak.

**Exception:** Tidak memiliki *exception*

137. HasPraktikum

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

138. HasResponsi

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method.

139. Kelulusan

Atribut yang dimiliki kelas ini adalah sebagai berikut.

- String PILIHAN\_WAJIB -
- String WAJIB -
- String AGAMA -
- int MIN\_SKS\_LULUS -
- int MIN\_PILIHAN\_WAJIB -

*Method-method* yang dimiliki kelas ini adalah sebagai berikut.

- public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)

**Parameter:**

- Mahasiswa mahasiswa -
- java.util.List reasonsContainer -

**Return Value:** Tidak memiliki *return value*

**Exception:** Tidak memiliki *exception*

**Override:** checkPrasyarat dari kelas Object