

# 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.

### 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  $\text{\LaTeX}$ .

#### 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 \LaTeX* atau *file PDF*. Untuk dapat menghasilkan *file PDF*, TeXDoclet mengintegrasikan Lua $\text{\LaTeX}$  untuk menghasilkan dokumen PDF dari sebuah *file \LaTeX*. Hasil PDF yang dihasilkan oleh TeXDoclet dapat dilihat pada lampiran D

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 \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*.



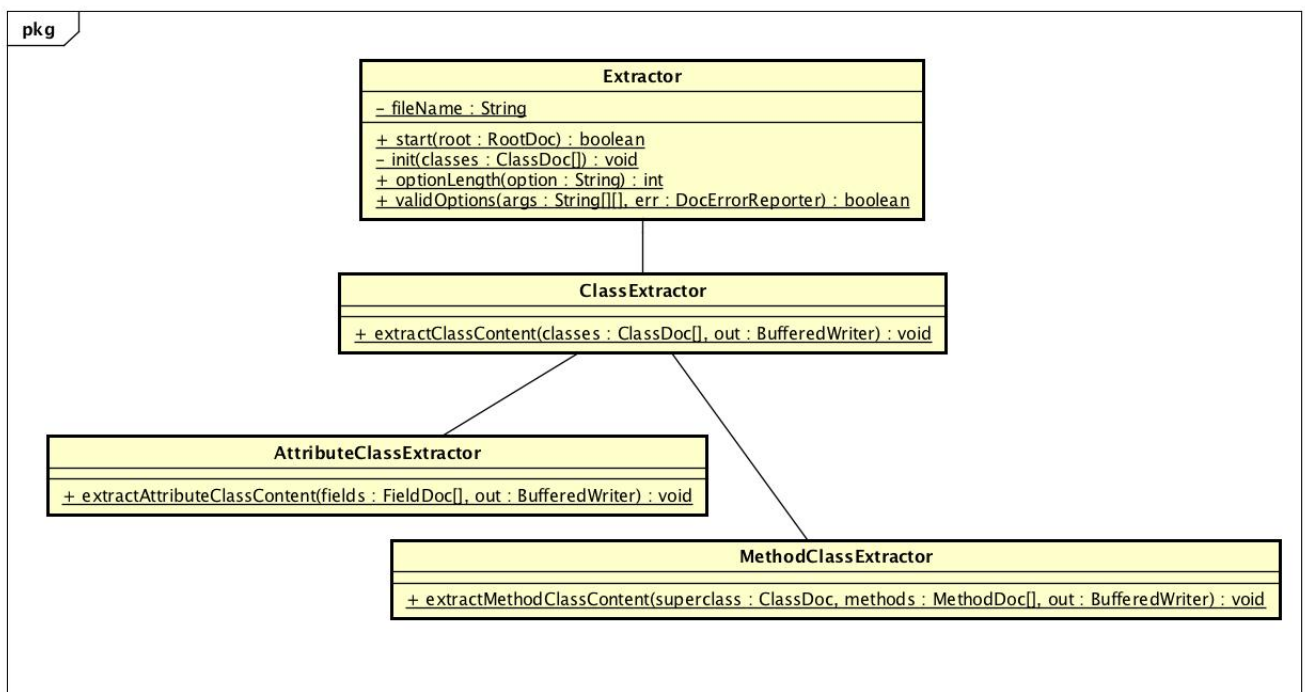
## BAB 4

### 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 DAN 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/Mac 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>
```

## 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
4 Kelas Abstract OperasiMatematika. Kelas ini memiliki method \texttt{calculate(int,
   int)}
5
6 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini
   adalah sebagai berikut.
7 \begin{itemize}
8 \item \texttt{public int calculate(int a, int b)}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{Return Value:} hasil perhitungan 2 buah bilangan DoubleNaN
18
19 \textbf{Exception:} Tidak memiliki \textit{exception}
20
21 \end{itemize}
22 \item \texttt{Pembagian}
23
24 Kelas ini merupakan Kelas Pembagian
25
26 Atribut yang dimiliki kelas ini adalah sebagai berikut.
27 \begin{itemize}
28 \item \texttt{int a} – Atribut A
29 \item \texttt{int b} – Atribut B

```



---

```

30 \end{itemize}
31 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
32 \begin{itemize}
33 \item \texttt{public int calculate(int a, int b)}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{Return Value:} hasil perhitungan 2 buah bilangan DoubleNaN
43
44 \textbf{Exception:} Tidak memiliki \textit{exception}
45
46 \end{itemize}
47 \item \texttt{Pengurangan}
48
49 Kelas ini merupakan Kelas Pengurangan
50
51 Atribut yang dimiliki kelas ini adalah sebagai berikut.
52 \begin{itemize}
53 \item \texttt{int a} – Atribut A
54 \item \texttt{int b} – Atribut B
55 \end{itemize}
56 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
57 \begin{itemize}
58 \item \texttt{public int calculate(int a, int b)}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{Return Value:} hasil perhitungan 2 buah bilangan DoubleNaN
68
69 \textbf{Exception:} Tidak memiliki \textit{exception}
70
71 \end{itemize}
72 \item \texttt{Perkalian}
73
74 Kelas ini merupakan Kelas Perkalian
75
76 Atribut yang dimiliki kelas ini adalah sebagai berikut.
77 \begin{itemize}
78 \item \texttt{int a} – Atribut A
79 \item \texttt{int b} – Atribut B
80 \end{itemize}
81 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
82 \begin{itemize}
83 \item \texttt{public int calculate(int a, int b)}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{Return Value}: hasil perhitungan 2 buah bilangan DoubleNaN
93
94 \textbf{Exception}: Tidak memiliki \textit{exception}
95
96 \end{itemize}
97 \item \texttt{Pertambahan}
98
99 Kelas ini merupakan Kelas Pertambahan
100
101 Atribut yang dimiliki kelas ini adalah sebagai berikut.
102 \begin{itemize}
103 \item \texttt{int a} – Atribut A
104 \item \texttt{int b} – Atribut B
105 \end{itemize}
106 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
107 \begin{itemize}
108 \item \texttt{public int calculate(int a, int b)}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{Return Value}: hasil perhitungan 2 buah bilangan DoubleNaN
118
119 \textbf{Exception}: Tidak memiliki \textit{exception}
120
121 \end{itemize}
122 \end{enumerate}

```

Listing 5.1: Hasil pengujian kode program sederhana

```

1 \begin{enumerate}
2 \item \texttt{AttributeClassExtractor}
3
4 Kelas ini merupakan kelas untuk mengambil informasi sebuah atribut yang
5 terdapat pada kelas
6
7 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini
    adalah sebagai berikut.
8 \begin{itemize}
9 \item \texttt{public static void extractAttributeClassContent(FieldDoc[] fields,
    java.io.BufferedWriter out)}\textit{Method} ini akan menampilkan atribut-atribut
    yang dimiliki oleh
10 sebuah kelas
11
12 \textbf{Parameter:}
13 \begin{itemize}
14 \item \texttt{FieldDoc fields} –
15 sebuah array berisikan sejumlah atribut dari kelas
16 \item \texttt{BufferedWriter out} –
17 turunan dari kelas \texttt{Writer} yang digunakan untuk menulis
18 file text
19 \end{itemize}
20 \textbf{Return Value}: Tidak memiliki \textit{return value}
21
22 \textbf{Exception}: Tidak memiliki \textit{exception}
23
24 \end{itemize}

```

```
25 \item \texttt{ClassExtractor}
26
27 Kelas ini merupakan kelas untuk mengambil informasi dari sebuah kelas
28
29 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini
    adalah sebagai berikut.
30 \begin{itemize}
31 \item \texttt{public static void extractClassContent(ClassDoc[] classes, java.io.
    BufferedWriter out)}\textit{Method} ini akan menampilkan nama kelas beserta
    penjelasan dari
32 sebuah kelas
33
34 \textbf{Parameter:}
35 \begin{itemize}
36 \item \texttt{ClassDoc classes} –
37 sebuah array berisikan sejumlah kelas
38 \item \texttt{BufferedWriter out} –
39 turunan dari kelas \texttt{Writer} yang digunakan untuk menulis
40 file text
41 \end{itemize}
42 \textbf{Return Value:} Tidak memiliki \textit{return value}
43
44 \textbf{Exception:} Tidak memiliki \textit{exception}
45
46 \end{itemize}
47 \item \texttt{Extractor}
48
49 Kelas ini merupakan kelas untuk menjalankan \textit{custom doclet}
50
51 Atribut yang dimiliki kelas ini adalah sebagai berikut.
52 \begin{itemize}
53 \item \texttt{String fileName} – atribut untuk nama \textit{file}
54 \end{itemize}
55 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
56 \begin{itemize}
57 \item \texttt{public static boolean start(RootDoc root)}\textit{Method} ini berperan
    sebagai \textit{method} untuk menjalankan
58 \textit{custom doclet}
59
60 \textbf{Parameter:}
61 \begin{itemize}
62 \item \texttt{RootDoc root} –
63 berperan sebagai mengambil seluruh informasi spesifik dari
64 \textit{option} yang terdapat pada \textit{command-line} sebuah
65 \textit{terminal}. Selain itu berperan juga untuk mengambil informasi
    dari
66 sekumpulan \textit{file java} yang akan di proses.
67 \end{itemize}
68 \textbf{Return Value:} kondisi true
69
70 \textbf{Exception:} Tidak memiliki \textit{exception}
71
72 \item \texttt{private static void init(ClassDoc[] classes)}\textit{Method} ini
    berperan untuk menulis kedalam sebuah \textit{file}
73 saat \textit{javadoc} berjalan.
74
75 \textbf{Parameter:}
76 \begin{itemize}
77 \item \texttt{ClassDoc classes} –
78 sebuah array yang berisikan sekumpulan \textit{file java}
79 yang akan di proses.
80 \end{itemize}
81 \textbf{Return Value:} Tidak memiliki \textit{return value}
```

---

```

82
83 \textbf{Exception}: Tidak memiliki \textit{exception}
84
85 \item \texttt{public static int optionLength(java.lang.String option)}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{Return Value}: panjang setiap option
94
95 \textbf{Exception}: Tidak memiliki \textit{exception}
96
97 \item \texttt{public static boolean validOptions(java.lang.String [][] args,
    DocErrorReporter err)}Pengecekan option valid
98
99 \textbf{Parameter:}
100 \begin{itemize}
101 \item \texttt{String args} –
102 String array 2 dimensi dari option
103 \item \texttt{DocErrorReporter err} –
104 sebuah error jika tidak terdapat option tersebut.
105 \end{itemize}
106 \textbf{Return Value}: bernilai true jika option tersebut dikenali, false jika
    option
107 tersebut tidak dikenali
108
109 \textbf{Exception}: Tidak memiliki \textit{exception}
110
111 \end{itemize}
112 \item \texttt{MethodClassExtractor}
113
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 extractMethodClassContent(ClassDoc superclass,
    MethodDoc[] methods, java.io.BufferedWriter out)}\textit{Method} ini akan
    menampilkan \textit{method-method} yang dimiliki
120 oleh sebuah kelas
121
122 \textbf{Parameter:}
123 \begin{itemize}
124 \item \texttt{ClassDoc superclass} –
125 sebuah objek ClassDoc
126 \item \texttt{MethodDoc methods} –
127 sebuah array berisikan sejumlah \textit{method} dari kelas
128 \item \texttt{BufferedWriter out} –
129 turunan dari kelas \texttt{Writer} yang digunakan untuk menulis
130 file text
131 \end{itemize}
132 \textbf{Return Value}: Tidak memiliki \textit{return value}
133
134 \textbf{Exception}: Tidak memiliki \textit{exception}
135
136 \item \texttt{private static void ParameterMethod(java.io.BufferedWriter out,
    Parameter[] paramMethod, ParamTag[] paramTags)}\textit{Method} ini akan
    menampilkan parameter \textit{method-method} yang dimiliki

```

```

137  oleh sebuah kelas
138
139 \textbf{Parameter:}
140 \begin{itemize}
141 \item \texttt{BufferedWriter out} –
142 turunan dari kelas \texttt{Writer} yang digunakan untuk menulis file text
143 \item \texttt{Parameter paramMethod} –
144 sebuah array berisikan sejumlah \textit{method} dari kelas
145 \item \texttt{ParamTag paramTags} –
146 sebuah array berisikan sejumlah parameter \textit{method} dari kelas
147 \end{itemize}
148 \textbf{Return Value:} Tidak memiliki \textit{return value}
149
150 \textbf{Exception:} Tidak memiliki \textit{exception}
151
152 \item \texttt{private static void ReturnRequestMethod(java.io.BufferedWriter out, Type
      type, Tag[] returnTags)} \textit{Method} ini akan menampilkan \textit{return type}
      } dari \textit{method-method} yang dimiliki
153 oleh sebuah kelas
154
155 \textbf{Parameter:}
156 \begin{itemize}
157 \item \texttt{BufferedWriter out} –
158 turunan dari kelas \texttt{Writer} yang digunakan untuk menulis file text
159 \item \texttt{Type type} –
160 sebuah objek Type
161 \item \texttt{Tag returnTags} –
162 sebuah array berisikan sejumlah \textit{return type} dari \textit{method} dari kelas
163 \end{itemize}
164 \textbf{Return Value:} Tidak memiliki \textit{return value}
165
166 \textbf{Exception:} Tidak memiliki \textit{exception}
167
168 \item \texttt{private static void ExceptionMethod(java.io.BufferedWriter out, Tag[]
      throwTags)} \textit{Method} ini akan menampilkan \textit{return type} dari \
      \textit{method-method} yang dimiliki
169 oleh sebuah kelas
170
171 \textbf{Parameter:}
172 \begin{itemize}
173 \item \texttt{BufferedWriter out} –
174 turunan dari kelas \texttt{Writer} yang digunakan untuk menulis file text
175 \item \texttt{Tag throwTags} –
176 sebuah array berisikan sejumlah \textit{exception} dari \textit{method} dari kelas
177 \end{itemize}
178 \textbf{Return Value:} Tidak memiliki \textit{return value}
179
180 \textbf{Exception:} Tidak memiliki \textit{exception}
181
182 \end{itemize}
183 \end{enumerate}

```

Listing 5.2: Hasil Pengujian kode program perangkat lunak

Hasil pengujian lengkap terdapat pada lampiran [A](#) untuk pengujian kode program sederhana dan lampiran [B](#) untuk pengujian kode program perangkat lunak.

### 5.2.3 Pengujian Eksperimental

Pengujian eksperimental dilakukan terhadap kode program SIAModels. SIAModels adalah sekumpulan kelas *java* yang mewakili objek-objek Sistem Informasi Akademik Universitas Katolik Parahyangan. SIAModels ini berisikan objek matakuliah pada Fakultas Teknologi Informasi dan Sains. Hasil pengujian terdapat pada lampiran [C](#)



## 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*.

#### 6.2 Saran

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

1. Pada saat ini, perangkat lunak tidak mengatasi *tag-tag html*. Pada pengujian menggunakan SIAModels terdapat beberapa kelas yang memiliki *tag-tag html* pada *javadoc* seperti pada kelas `Mahasiswa` pada fungsi `calculateIPKLulus()`. Terdapat kode aneh yaitu tanda "!" yang merepresentasikan tanda lebih kecil "<" dan tanda "?" terbalik yang merepresentasikan tanda lebih besar ">". Sebaiknya perangkat lunak dapat mengatasi *tag-tag html* jika javadoc memiliki *tag-tag* tersebut.





## 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

## PENGUJIAN TERHADAP KODE PROGRAM SEDERHANA

### A.1 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  }
```

```

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 }

```

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     * Atribut B
18     */
19     private int b;
20
21     @Override
22     public int calculate(int a, int b) {
23         int hasil = a * b;
24         return hasil;
25     }
26 }

```

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     * Atribut B
18     */
19     private int b;
20
21     @Override
22     public int calculate(int a, int b) {
23         int hasil = a + b;
24         return hasil;
25     }
26 }

```

## A.2 Hasil Latex

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| Bilagan 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}
125| \end{document}
```

### A.3 Hasil PDF

Hasil PDF pada listing [A.6](#) dapat dilihat pada halaman selanjutnya.

### 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

## PENGUJIAN TERHADAP KODE PROGRAM PERANGKAT LUNAK

### B.1 Kode Program

Listing B.1: Extractor.java

```
1 package extractor;
2
3 import com.sun.javadoc.*;
4 import java.io.*;
5
6 /**
7  * Kelas ini merupakan kelas untuk menjalankan \textit{custom doclet}
8  *
9  * @author Adli Fariz Bonaputra
10 */
11 public class Extractor {
12
13     /**
14      * atribut untuk nama \textit{file}
15      */
16     private static String fileName;
17
18     /**
19      * \textit{Method} ini berperan sebagai \textit{method} untuk menjalankan
20      * \textit{custom doclet}
21      *
22      * @param root berperan sebagai mengambil seluruh informasi spesifik dari
23      * \textit{option} yang terdapat pada \textit{command-line} sebuah
24      * \textit{terminal}. Selain itu berperan juga untuk mengambil informasi dari
25      * sekumpulan \textit{file java} yang akan di proses.
26      * @return kondisi true
27      */
28     public static boolean start(RootDoc root) {
29         init(root.classes());
30         return true;
31     }
32
33     /**
34      * \textit{Method} ini berperan untuk menulis kedalam sebuah \textit{file}
35      * saat \textit{javadoc} berjalan.
36      *
37      * @param classes sebuah array yang berisikan sekumpulan \textit{file java}
38      * yang akan di proses.
39      */
40     private static void init(ClassDoc[] classes) {
41         FileWriter file;
42         new File("output").mkdirs();
43         try {
44             if (fileName == null) {
45                 file = new FileWriter("output/doc.tex");
46             } else {
47                 file = new FileWriter("output/" + fileName + ".tex");
48             }
49             BufferedWriter out = new BufferedWriter(file);
50             out.write("\\documentclass{article}\n");
51             out.write("\\begin{document}\n");
52             out.write("\\begin{enumerate}\n");
53
54             ClassExtractor.extractClassContent(classes, out);
55
56             out.write("\\end{enumerate}\n");
57             out.write("\\end{document}\n");
58             out.close();
59         } catch (IOException e) { }
60     }
61
62     /**
63      * Method untuk menghitung banyak option yang digunakan pada
64      * \textit{command-line}
65      *
66      * @param option sebuah option
67      * @return panjang setiap option
68      */
69 }
```

```

69 public static int optionLength(String option) {
70     if (option.equals("-filename")) {
71         return 2;
72     }
73     return Doclet.optionLength(option);
74 }
75 }
76
77 /**
78  * Pengecekan option valid
79  *
80  * @param args String array 2 dimensi dari option
81  * @param err sebuah error jika tidak terdapat option tersebut.
82  * @return bernilai true jika option tersebut dikenali, false jika option
83  * tersebut tidak dikenali
84  */
85 public static boolean validOptions(String[][] args, DocErrorReporter err) {
86     for (int i = 0; i < args.length; ++i) {
87         if (args[i][0].equals("-filename")) {
88             fileName = args[i][1];
89         }
90     }
91     return Doclet.validOptions(args, err);
92 }
93 }

```

Listing B.2: ClassExtractor.java

```

1 package extractor;
2
3 import com.sun.javadoc.*;
4 import java.io.*;
5
6 /**
7  * Kelas ini merupakan kelas untuk mengambil informasi dari sebuah kelas
8  *
9  * @author Adli Fariz Bonaputra
10  */
11 public class ClassExtractor {
12
13     /**
14      * \textit{Method} ini akan menampilkan nama kelas berserta penjelasan dari
15      * sebuah kelas
16      *
17      * @param classes sebuah array berisikan sejumlah kelas
18      * @param out turunan dari kelas \texttt{Writer} yang digunakan untuk menulis
19      * file text
20      */
21     public static void extractClassContent(ClassDoc[] classes, BufferedWriter out) {
22         try {
23             for (ClassDoc classDoc : classes) {
24                 out.write("\\item_\\texttt{" + classDoc.name() + "}\\n\\n");
25
26                 Tag[] inlineTags = classDoc.inlineTags();
27                 for (int i = 0; i < inlineTags.length; i++) {
28                     if (i == 1) {
29                         out.write("\\texttt{" + inlineTags[i].text().replace("#", "") + "}");
30                     } else {
31                         out.write(inlineTags[i].text().replace("#", "").replace("_", "\\_").replace("&", "\\&"));
32                     }
33                 }
34                 out.write("\\n\\n");
35
36                 FieldDoc[] fields = classDoc.fields(false);
37                 AttributeClassExtractor.extractAttributeClassContent(fields, out);
38
39                 MethodDoc[] methods = classDoc.methods(false);
40                 MethodClassExtractor.extractMethodClassContent(classDoc, methods, out);
41             }
42         } catch (IOException e) {}
43     }
44 }

```

Listing B.3: AttributeClassExtractor.java

```

1 package extractor;
2
3 import com.sun.javadoc.*;
4 import java.io.*;
5
6 /**
7  * Kelas ini merupakan kelas untuk mengambil informasi sebuah atribut yang
8  * terdapat pada kelas
9  *
10  * @author Adli Fariz Bonaputra
11  */
12 public class AttributeClassExtractor {
13
14     /**
15      * \textit{Method} ini akan menampilkan atribut-atribut yang dimiliki oleh
16      * sebuah kelas
17      *
18      * @param fields sebuah array berisikan sejumlah atribut dari kelas
19      * @param out turunan dari kelas \texttt{Writer} yang digunakan untuk menulis
20      * file text
21      */

```



```

77     }
78     }
79     out.write(")");
80
81     Tag[] inlineTags = method.inlineTags();
82     for (int i = 0; i < inlineTags.length; i++) {
83         if (i == 1) {
84             out.write("\\texttt{" + inlineTags[i].text().replace("#", "") + "}");
85         } else {
86             out.write(inlineTags[i].text().replace("#", ""));
87         }
88     }
89
90     out.write("\n\n");
91
92     ParamTag[] paramTags = method.paramTags();
93     ParameterMethod(out, paramMethod, paramTags);
94
95     Tag[] returnTags = method.tags("return");
96     Tag[] throwTags = method.tags("throws");
97     AnnotationDesc[] override = method.annotations();
98     ReturnTypeInfo(out, type, returnTags);
99     ExceptionMethod(out, throwTags);
100     if (override.length != 0) {
101         out.write("\\textbf{Override}:_\\texttt{" + method.name() + "_dari_kelas_\\texttt{" + superclass.superclass().name
102             () + "}");
103         out.write("\n\n");
104     }
105 }
106 out.write("\\end{itemize}\n");
107 }
108 } catch (IOException e) { }
109 }
110
111 /**
112  * \\textit{Method} ini akan menampilkan parameter \\textit{method-method} yang dimiliki
113  * oleh sebuah kelas
114  *
115  * @param out turunan dari kelas \\texttt{Writer} yang digunakan untuk menulis file text
116  * @param paramMethod sebuah array berisikan sejumlah \\textit{method} dari kelas
117  * @param paramTags sebuah array berisikan sejumlah parameter \\textit{method} dari kelas
118  */
119 private static void ParameterMethod(BufferedWriter out, Parameter[] paramMethod, ParamTag[] paramTags){
120     try {
121         if (paramTags.length == 0) {
122             out.write("\\textbf{Parameter:}\n");
123             out.write("\\begin{itemize}\n");
124             if (paramMethod.length != 0) {
125                 for (int i = 0; i < paramMethod.length; i++) {
126                     out.write("\\item_\\texttt{" + paramMethod[i].toString() + "_-\n");
127                 }
128             } else {
129                 out.write("\\item_Tidak_miliki_parameter_\\textit{method}\n");
130             }
131             out.write("\\end{itemize}\n");
132         } else {
133             out.write("\\textbf{Parameter:}\n");
134             out.write("\\begin{itemize}\n");
135             for (int k = 0; k < paramTags.length; k++) {
136                 Type type = paramMethod[k].type();
137                 out.write("\\item_\\texttt{" + type.typeName() + "_-_" + paramTags[k].parameterName() + "_-\n");
138                 Tag[] inlineTagsInParameter = paramTags[k].inlineTags();
139                 for (int i = 0; i < inlineTagsInParameter.length; i++) {
140                     if (i == 1) {
141                         out.write("\\texttt{" + inlineTagsInParameter[i].text().replace("#", "") + "}");
142                     } else {
143                         out.write(inlineTagsInParameter[i].text());
144                     }
145                 }
146                 out.write("\n");
147             }
148             out.write("\\end{itemize}\n");
149         }
150     } catch (IOException e) { }
151 }
152
153 /**
154  * \\textit{Method} ini akan menampilkan \\textit{return type} dari \\textit{method-method} yang dimiliki
155  * oleh sebuah kelas
156  *
157  * @param out turunan dari kelas \\texttt{Writer} yang digunakan untuk menulis file text
158  * @param type sebuah objek Type
159  * @param returnTags sebuah array berisikan sejumlah \\textit{return type} dari \\textit{method} dari kelas
160  */
161 private static void ReturnTypeInfo(BufferedWriter out, Type type, Tag[] returnTags){
162     try {
163         if (type.typeName().equals("void") || returnTags.length == 0) {
164             out.write("\\textbf{Return_Value}:_Tidak_miliki_\\textit{return_value}\n");
165         } else {
166             out.write("\\textbf{Return_Value}:_");
167             Tag[] inlineTagsInReturnValue = returnTags[0].inlineTags();
168             for (int i = 0; i < inlineTagsInReturnValue.length; i++) {
169                 if (i == 1) {
170                     out.write(inlineTagsInReturnValue[i].text().replace("#", ""));
171                 } else {
172                     out.write(inlineTagsInReturnValue[i].text().replace("#{link", "").replace("}", "").replace("#", ""));
173                 }
174             }

```

```

175         out.write("\n\n");
176     }
177 } catch (IOException e) { }
178 }
179
180 /**
181  * \textit{Method} ini akan menampilkan \textit{return type} dari \textit{method-method} yang dimiliki
182  * oleh sebuah kelas
183  *
184  * @param out turunan dari kelas \texttt{Writer} yang digunakan untuk menulis file text
185  * @param throwTags sebuah array berisikan sejumlah \textit{exception} dari \textit{method} dari kelas
186  */
187 private static void ExceptionMethod(BufferedWriter out, Tag[] throwTags) throws IOException {
188     try {
189         if (throwTags.length == 0) {
190             out.write("\\textbf{Exception}: Tidak memiliki \\textit{exception}");
191             out.write("\n\n");
192         } else {
193             out.write("\\textbf{Exception}: " + throwTags[0].text());
194             out.write("\n\n");
195         }
196     } catch (IOException e) { }
197 }
198 }

```

## B.2 Hasil Latex

Listing B.5: javadoctolatex.tex

```

1 \documentclass{article}
2 \begin{document}
3 \begin{enumerate}
4 \item \texttt{AttributeClassExtractor}
5
6 Kelas ini merupakan kelas untuk mengambil informasi sebuah atribut yang
7 terdapat pada kelas
8
9 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
10 \begin{itemize}
11 \item \texttt{public static void extractAttributeClassContent(FieldDoc[] fields, java.io.BufferedWriter out)} \textit{Method} ini
12 akan menampilkan atribut-atribut yang dimiliki oleh
13 sebuah kelas
14 \textbf{Parameter:}
15 \begin{itemize}
16 \item \texttt{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{Return Value:} Tidak memiliki \textit{return value}
23
24 \textbf{Exception:} Tidak memiliki \textit{exception}
25
26 \end{itemize}
27 \item \texttt{ClassExtractor}
28
29 Kelas ini merupakan kelas untuk mengambil informasi dari sebuah kelas
30
31 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
32 \begin{itemize}
33 \item \texttt{public static void extractClassContent(ClassDoc[] classes, java.io.BufferedWriter out)} \textit{Method} ini akan
34 menampilkan nama kelas berserta penjelasan dari
35 sebuah kelas
36 \textbf{Parameter:}
37 \begin{itemize}
38 \item \texttt{ClassDoc classes} -
39 sebuah array berisikan sejumlah kelas
40 \item \texttt{BufferedWriter out} -
41 turunan dari kelas \texttt{Writer} yang digunakan untuk menulis
42 file text
43 \end{itemize}
44 \textbf{Return Value:} Tidak memiliki \textit{return value}
45
46 \textbf{Exception:} Tidak memiliki \textit{exception}
47
48 \end{itemize}
49 \item \texttt{Extractor}
50
51 Kelas ini merupakan kelas untuk menjalankan \textit{custom doclet}
52
53 Atribut yang dimiliki kelas ini adalah sebagai berikut.
54 \begin{itemize}
55 \item \texttt{String fileName} - atribut untuk nama \textit{file}
56 \end{itemize}
57 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
58 \begin{itemize}
59 \item \texttt{public static boolean start(RootDoc root)} \textit{Method} ini berperan sebagai \textit{method} untuk menjalankan
60 \textit{custom doclet}
61
62 \textbf{Parameter:}
63 \begin{itemize}
64 \item \texttt{RootDoc root} -

```

```

65 berperan sebagai mengambil seluruh informasi spesifik dari
66     \textit{option} yang terdapat pada \textit{command-line} sebuah
67     \textit{terminal}. Selain itu berperan juga untuk mengambil informasi dari
68     sekumpulan \textit{file java} yang akan di proses.
69 \end{itemize}
70 \textbf{Return Value}: kondisi true
71
72 \textbf{Exception}: Tidak memiliki \textit{exception}
73
74 \item \texttt{private static void init(ClassDoc[] classes)}\textit{Method} ini berperan untuk menulis kedalam sebuah \textit{file}
75 saat \textit{javadoc} berjalan.
76
77 \textbf{Parameter:}
78 \begin{itemize}
79 \item \texttt{ClassDoc classes} -
80 sebuah array yang berisikan sekumpulan \textit{file java}
81     yang akan di proses.
82 \end{itemize}
83 \textbf{Return Value}: Tidak memiliki \textit{return value}
84
85 \textbf{Exception}: Tidak memiliki \textit{exception}
86
87 \item \texttt{public static int optionLength(java.lang.String option)}\textit{Method} untuk menghitung banyak option yang digunakan pada
88 \textit{command-line}
89
90 \textbf{Parameter:}
91 \begin{itemize}
92 \item \texttt{String option} -
93 sebuah option
94 \end{itemize}
95 \textbf{Return Value}: panjang setiap option
96
97 \textbf{Exception}: Tidak memiliki \textit{exception}
98
99 \item \texttt{public static boolean validOptions(java.lang.String[][] args, DocErrorReporter err)}\textit{Pengecekan option valid}
100
101 \textbf{Parameter:}
102 \begin{itemize}
103 \item \texttt{String args} -
104 String array 2 dimensi dari option
105 \item \texttt{DocErrorReporter err} -
106 sebuah error jika tidak terdapat option tersebut.
107 \end{itemize}
108 \textbf{Return Value}: bernilai true jika option tersebut dikenali, false jika option
109     tersebut tidak dikenali
110
111 \textbf{Exception}: Tidak memiliki \textit{exception}
112
113 \end{itemize}
114 \item \texttt{MethodClassExtractor}
115
116 Kelas ini merupakan kelas untuk mengambil informasi sebuah \textit{method}
117 terdapat pada kelas
118
119 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
120 \begin{itemize}
121 \item \texttt{public static void extractMethodClassContent(ClassDoc superclass, MethodDoc[] methods, java.io.BufferedWriter out)}\textit{Method}
122     \textit{Method} ini akan menampilkan \textit{method-method} yang dimiliki
123     oleh sebuah kelas
124
125 \textbf{Parameter:}
126 \begin{itemize}
127 \item \texttt{ClassDoc superclass} -
128 sebuah objek ClassDoc
129 \item \texttt{MethodDoc methods} -
130 sebuah array berisikan sejumlah \textit{method} dari kelas
131 \item \texttt{BufferedWriter out} -
132 turunan dari kelas \textit{Writer} yang digunakan untuk menulis
133     file text
134 \end{itemize}
135 \textbf{Return Value}: Tidak memiliki \textit{return value}
136
137 \textbf{Exception}: Tidak memiliki \textit{exception}
138
139 \item \texttt{private static void ParameterMethod(java.io.BufferedWriter out, Parameter[] paramMethod, ParamTag[] paramTags)}\textit{Method}
140     \textit{Method} ini akan menampilkan parameter \textit{method-method} yang dimiliki
141     oleh sebuah kelas
142
143 \textbf{Parameter:}
144 \begin{itemize}
145 \item \texttt{BufferedWriter out} -
146 turunan dari kelas \textit{Writer} yang digunakan untuk menulis file text
147 \item \texttt{Parameter paramMethod} -
148 sebuah array berisikan sejumlah \textit{method} dari kelas
149 \item \texttt{ParamTag paramTags} -
150 sebuah array berisikan sejumlah parameter \textit{method} dari kelas
151 \end{itemize}
152 \textbf{Return Value}: Tidak memiliki \textit{return value}
153
154 \textbf{Exception}: Tidak memiliki \textit{exception}
155
156 \item \texttt{private static void ReturnTypeInfoMethod(java.io.BufferedWriter out, Type type, Tag[] returnTags)}\textit{Method} ini
157     akan menampilkan \textit{return type} dari \textit{method-method} yang dimiliki
158     oleh sebuah kelas
159
160 \textbf{Parameter:}
161 \begin{itemize}
162 \item \texttt{BufferedWriter out} -
163 turunan dari kelas \textit{Writer} yang digunakan untuk menulis file text

```

```

161 \item \texttt{Type type} -
162 sebuah objek Type
163 \item \texttt{Tag returnTags} -
164 sebuah array berisikan sejumlah \textit{return type} dari \textit{method} dari kelas
165 \end{itemize}
166 \textbf{Return Value}: Tidak memiliki \textit{return value}
167
168 \textbf{Exception}: Tidak memiliki \textit{exception}
169
170 \item \texttt{private static void ExceptionMethod(java.io.BufferedWriter out, Tag[] throwTags)}\textit{Method} ini akan
171 menampilkan \textit{return type} dari \textit{method-method} yang dimiliki
172 oleh sebuah kelas
173 \textbf{Parameter:}
174 \begin{itemize}
175 \item \texttt{BufferedWriter out} -
176 turunan dari kelas \texttt{Writer} yang digunakan untuk menulis file text
177 \item \texttt{Tag throwTags} -
178 sebuah array berisikan sejumlah \textit{exception} dari \textit{method} dari kelas
179 \end{itemize}
180 \textbf{Return Value}: Tidak memiliki \textit{return value}
181
182 \textbf{Exception}: Tidak memiliki \textit{exception}
183
184 \end{itemize}
185 \end{enumerate}
186 \end{document}

```

## B.3 Hasil PDF

Hasil PDF pada listing [B.5](#) dapat dilihat pada halaman selanjutnya.

### 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(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(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(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(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, 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(ClassDoc superclass, 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, Parameter[] paramMethod, 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, Type type, 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, 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*

# LAMPIRAN C

## PENGUJIAN TERHADAP KODE PROGRAM SIAMODELS

Listing C.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 \textbf{Parameter:}
```

```

74 \begin{itemize}
75 \item \texttt{String kode} -
76 kode mata kuliah
77 \end{itemize}
78 \textbf{Return Value}: objek mata kuliah
79
80 \textbf{Exception}: IllegalStateException
81     jika sks dan tidak sesuai dengan yang ada di kode
82
83 \end{itemize}
84 \item \texttt{Semester}
85
86
87
88 Atribut yang dimiliki kelas ini adalah sebagai berikut.
89 \begin{itemize}
90 \item \texttt{Semester UNKNOWN5} -
91 \item \texttt{Semester TRANSFER} -
92 \item \texttt{Semester PENDEK} -
93 \item \texttt{Semester GANJIL} -
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 implements Comparable}
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()}}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 |
194 | \item \texttt{public int compareTo(id.ac.unpar.siamodels.TahunSemester o)}
195 |
196 | \textbf{Parameter:}
197 | \begin{itemize}
198 | \item \texttt{TahunSemester o} -
199 | \end{itemize}
200 | \textbf{Return Value:} Tidak memiliki \textit{return value}
201 |
202 | \textbf{Exception:} Tidak memiliki \textit{exception}
203 |
204 | \textbf{Override:} \texttt{compareTo} dari kelas \texttt{Object}
205 |
206 | \item \texttt{public boolean equals(java.lang.Object arg0)}
207 |
208 | \textbf{Parameter:}
209 | \begin{itemize}
210 | \item \texttt{Object arg0} -
211 | \end{itemize}
212 | \textbf{Return Value:} Tidak memiliki \textit{return value}
213 |
214 | \textbf{Exception:} Tidak memiliki \textit{exception}
215 |
216 | \item \texttt{public String toString()}}
217 |
218 | \textbf{Parameter:}
219 | \begin{itemize}
220 | \item Tidak memiliki parameter \textit{method}
221 | \end{itemize}
222 | \textbf{Return Value:} Tidak memiliki \textit{return value}
223 |
224 | \textbf{Exception:} Tidak memiliki \textit{exception}
225 |
226 | \end{itemize}
227 | \item \texttt{Mahasiswa}
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 getNama()}}
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 setNama(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 </ul>
402 Sebelum memanggil method ini, \texttt{getRiwayatNilai()} harus sudah mengandung nilai per mata kuliah!
403
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 </ul>
417
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 </ul>
432 Sebelum memanggil method ini, \texttt{getRiwayatNilai()} harus sudah mengandung nilai per mata kuliah!
433
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
    kuliah tertentu. Kompleksitas  $O(n)$ .
479 Sebelum memanggil method ini, \texttt{getRiwayatNilai()} harus sudah mengandung nilai per mata kuliah!
480 Note: jika yang dimiliki adalah MataKuliah, gunakanlah MataKuliahgetKode().
481
482 \textbf{Parameter:}
483 \begin{itemize}
484 \item \texttt{String kodeMataKuliah} -
485 kode mata kuliah yang ingin diperiksa kelulusannya.
486 \end{itemize}
487 \textbf{Return Value:} true jika sudah pernah mengambil dan lulus, false jika belum
488
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 MataKuliahgetKode().
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 implements Comparator}
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 getLokasi()}
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 setLokasi(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 getKode()}
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 getNama()}
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 implements HasPrasyarat}
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 |
983 | \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
984 |
985 | \end{itemize}
986 | \item \texttt{AIF311 implements HasPrasyarat}
987 |
988 | \item \texttt{AIF311 implements HasPraktikum}
989 |
990 | Kuliah Pemrograman Fungsional bertujuan untuk: 1. memperkenalkan paradigma
991 | pemrograman fungsional, yaitu sebuah pemrograman yang didasarkan pada konsep
992 | pemetaan dan fungsi matematika. Penyelesaian suatu masalah didasari atas
993 | aplikasi dari fungsi-fungsi tersebut. 2. memberikan dasar-dasar pemrograman
994 | fungsional dengan menggunakan bahasa fungsional Haskell.
995 |
996 | Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
997 | \begin{itemize}
998 | \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
999 |
1000 | \textbf{Parameter:}
1001 | \begin{itemize}
1002 | \item \texttt{Mahasiswa mahasiswa} -
1003 | \item \texttt{java.util.List reasonsContainer} -
1004 | \end{itemize}
1005 | \textbf{Return Value:} Tidak memiliki \textit{return value}
1006 |
1007 | \textbf{Exception:} Tidak memiliki \textit{exception}
1008 |
1009 | \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1010 |
1011 | \end{itemize}
1012 | \item \texttt{AIF192}
1013 |
1014 |
1015 |
1016 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF468}
1017 |
1018 |
1019 |
1020 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{IIE103}
1021 |
1022 |
1023 |
1024 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF385}
1025 |
1026 |
1027 |
1028 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF106}
1029 |
1030 | Mata kuliah ini memberikan pengetahuan tentang cara kerja komputer, dimulai
1031 | dari representasi data dan berbagai macam operasinya. Selanjutnya, juga
1032 | diperkenalkan bagaimana merepresentasikan suatu fungsi dalam rangkaian
1033 | gerbang logika, dan bagaimana menyederhanakannya. Berbagai rangkaian dasar
1034 | yang digunakan di dalam komputer juga dipekenalkan. Mahasiswa juga akan
1035 | mempelajari komponen komputer, misalnya register dan memori.
1036 |
1037 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF281}
1038 |
1039 |
1040 |
1041 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{EAA102}
1042 |
1043 |
1044 |
1045 | Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF405 implements HasPrasyarat}
1046 |
1047 | \item \texttt{AIF405 implements HasPraktikum}
1048 |
1049 | Mata kuliah ini merupakan lanjutan dari Projek Sistem Informasi 1 dan
1050 | memberikan kesempatan bagi mahasiswa untuk melanjutkan/mengembangkan
1051 | perancangan sitem pada organisasi studi kasus, mengimplementasikan rancangan
1052 | dan melakukan pengujian perangkat lunak;
1053 |
1054 | Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1055 | \begin{itemize}
1056 | \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1057 |
1058 | \textbf{Parameter:}
1059 | \begin{itemize}
1060 | \item \texttt{Mahasiswa mahasiswa} -

```

```

1061 \item \texttt{java.util.List reasonsContainer} -
1062 \end{itemize}
1063 \textbf{Return Value}: Tidak memiliki \textit{return value}
1064
1065 \textbf{Exception}: Tidak memiliki \textit{exception}
1066
1067 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1068
1069 \end{itemize}
1070 \item \texttt{APS182}
1071
1072
1073 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{MKU004}
1074
1075 Fenomenologi Agama merupakan bagian yang tak terpisahkan dari kajian filosofis, kritis,
1076 rasional, dan obyektif mengenai substansi ajaran agama. Fenomenologi merupakan sebuah
1077 disiplin ilmu yang secara kritis-rasional mengkaji fenomena dan dinamika kehidupan manusia
1078 beragama, dari upaya menjadikan Tuhan sebagai tujuan sesembahan sampai menempatkan Tuhan
1079 sebagai instrumen legitimasi untuk melakukan tindakan yang justru bertolak belakang dengan
1080 kehendak Tuhan yang disembah. Sehubungan dengan itu, kritik konstruktif terhadap perilaku
1081 manusia beragama menjadi salah satu poin utama dalam mata kuliah ini. Kesiadaan untuk
1082 melakukan otoritik terhadap agama sendiri erat terkait dengan upaya menemukan kembali nilai
1083 sejati agama atau otentisitas hidup beragama.
1084
1085 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{MKU012}
1086
1087 Perkuliahan logika ditujukan untuk memberikan dasar-dasar ketrampilan berpikir rasional dan
1088 sistematis. Isinya mencakup ketrampilan berpikir deduktif dan induktif, seperti silogisme,
1089 argumen analogikal dan generalisasi induktif. Pembahasan teoretis disertai pula dengan
1090 pelatihan praktis yang diarahkan pada proses berpikir. Untuk menajamkan kemampuan berpikir
1091 tersebut, mahasiswa dilatih pula mengidentifikasi kerancuan-kerancuan (fallacies) yang sering
1092 dijumpai baik dalam kehidupan sehari-hari maupun dalam konteks akademik.
1093
1094 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF389}
1095
1096
1097
1098 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AMS190}
1099
1100
1101
1102 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AMS191}
1103
1104
1105
1106 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AMS200}
1107
1108
1109
1110 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF330}
1111
1112
1113
1114 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF388}
1115
1116
1117
1118 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF465}
1119
1120
1121
1122 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF453 implements HasPrasyarat}
1123
1124 Mata kuliah ini memperkenalkan kebutuhan organisasi terhadap sistem business
1125 intelligent (BI) dan pemanfaatan BI untuk organisasi; memperkenalkan konsep
1126 sistem business intelligent dan komponennya; Mempelajari teknik-teknik
1127 analisis data bisnis dan visualisasi hasil analisis; Mempelajari konsep data
1128 warehouse dan perancangannya dan fungsi OLAP; Mempraktekkan teknik-teknik
1129 analisis data dan visualisasi hasil analisis.
1130
1131 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1132 \begin{itemize}
1133 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1134
1135 \textbf{Parameter}:
1136 \begin{itemize}
1137 \item \texttt{Mahasiswa mahasiswa} -
1138 \item \texttt{java.util.List reasonsContainer} -
1139 \end{itemize}
1140 \textbf{Return Value}: Tidak memiliki \textit{return value}
1141
1142 \textbf{Exception}: Tidak memiliki \textit{exception}
1143
1144 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1145
1146 \end{itemize}
1147 \item \texttt{AIF280}
1148
1149
1150
1151 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF445 implements HasPrasyarat}
1152
1153
1154
1155 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1156 \begin{itemize}
1157 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1158
1159

```

```

1160 \textbf{Parameter:}
1161 \begin{itemize}
1162 \item \texttt{Mahasiswa mahasiswa} -
1163 \item \texttt{java.util.List reasonsContainer} -
1164 \end{itemize}
1165 \textbf{Return Value:} Tidak memiliki \textit{return value}
1166
1167 \textbf{Exception:} Tidak memiliki \textit{exception}
1168
1169 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1170
1171 \end{itemize}
1172 \item \texttt{AIF469 implements HasPrasyarat}
1173
1174 Mata kuliah ini mengajarkan kepada mahasiswa teknik-teknik untuk membuat
1175 layanan berbasis web. Mahasiswa diperkenalkan dengan standar-standar seperti
1176 HTTP, XML, JSON dan diajarkan untuk memanfaatkannya dalam membuat maupun
1177 menggunakan layanan pihak ketiga. Dalam kuliah ini, juga akan diperkenalkan
1178 minimal satu layanan pihak ketiga yang dapat dimanfaatkan mahasiswa, seperti
1179 Google Places Web Service.
1180
1181 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1182 \begin{itemize}
1183 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1184
1185 \textbf{Parameter:}
1186 \begin{itemize}
1187 \item \texttt{Mahasiswa mahasiswa} -
1188 \item \texttt{java.util.List reasonsContainer} -
1189 \end{itemize}
1190 \textbf{Return Value:} Tidak memiliki \textit{return value}
1191
1192 \textbf{Exception:} Tidak memiliki \textit{exception}
1193
1194 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1195
1196 \end{itemize}
1197 \item \texttt{AIF486}
1198
1199
1200
1201 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{IIE207}
1202
1203
1204
1205 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF202 implements HasPrasyarat}
1206
1207 \item \texttt{AIF202 implements HasResponsi}
1208
1209 \item \texttt{AIF202 implements HasPraktikum}
1210
1211 Mata kuliah ini memperkenalkan kepada mahasiswa beberapa algoritma dan
1212 struktur data, alternatif cara implementasinya, dan analisis kompleksitas
1213 waktunya. Mahasiswa diberikan beberapa masalah komputasi yang harus
1214 diselesaikan dengan menggunakan algoritma atau struktur data yang sudah
1215 diperkenalkan dan mengimplementasikannya dalam bahasa pemrograman Java.
1216
1217 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1218 \begin{itemize}
1219 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1220
1221 \textbf{Parameter:}
1222 \begin{itemize}
1223 \item \texttt{Mahasiswa mahasiswa} -
1224 \item \texttt{java.util.List reasonsContainer} -
1225 \end{itemize}
1226 \textbf{Return Value:} Tidak memiliki \textit{return value}
1227
1228 \textbf{Exception:} Tidak memiliki \textit{exception}
1229
1230 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1231
1232 \end{itemize}
1233 \item \texttt{AIF347}
1234
1235
1236
1237 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{MKU009}
1238
1239 Tujuan dari mata kuliah ini adalah untuk mendalami keterampilan berbahasa Indonesia, agar
1240 mampu mengkomunikasikan hasil pemikiran serta meningkatkan keterampilan dalam menyusun karya
1241 ilmiah. Mata kuliah Bahasa Indonesia ini dimulai dengan mempelajari penulisan kata baku dan
1242 non baku serta pengungkapan pikiran dengan punctuation yang benar. Selanjutnya dipelajari
1243 penyusunan kalimat yang baku serta menghubungkan kalimat-kalimat yang padu dalam menuangkan
1244 gagasan dalam sebuah paragraf. Selain itu, dalam matakuliah ini dipelajari cara menyusun
1245 surat dinas yang jelas dan komunikatif. Di akhir kuliah ini, mahasiswa diberi tugas
1246 penyusunan makalah dengan benar.
1247
1248 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{ESM203}
1249
1250
1251
1252 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{APS402 implements HasPrasyarat}
1253
1254
1255
1256 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1257 \begin{itemize}
1258 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}

```

```

1259 \textbf{Parameter:}
1260 \begin{itemize}
1261 \item \texttt{Mahasiswa mahasiswa} -
1262 \item \texttt{java.util.List reasonsContainer} -
1263 \end{itemize}
1264 \textbf{Return Value}: Tidak memiliki \textit{return value}
1265
1266 \textbf{Exception}: Tidak memiliki \textit{exception}
1267
1268 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1269
1270 \end{itemize}
1271 \item \texttt{AIF306 implements HasPrasyarat}
1272
1273 Mata kuliah ini bertujuan untuk memberikan pengalaman bagi mahasiswa dalam
1274 mengerjakan proyek dengan teknologi-teknologi terkini, secara berkelompok.
1275 Teknologi-teknologi yang digunakan pada kuliah ini tidak spesifik dan dapat
1276 berubah seiring perkembangan teknologi maupun disesuaikan dengan kompetensi
1277 dosen pengajar. Beberapa teknologi yang dapat dimanfaatkan antara lain: DVCS
1278 tool menggunakan Git + Github, Mobile native app (Android, iOS, dll), dan
1279 responsive web design.
1280
1281 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1282 \begin{itemize}
1283 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1284 \end{itemize}
1285
1286 \textbf{Parameter:}
1287 \begin{itemize}
1288 \item \texttt{Mahasiswa mahasiswa} -
1289 \item \texttt{java.util.List reasonsContainer} -
1290 \end{itemize}
1291 \textbf{Return Value}: Tidak memiliki \textit{return value}
1292
1293 \textbf{Exception}: Tidak memiliki \textit{exception}
1294
1295 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1296
1297 \end{itemize}
1298 \item \texttt{AKS122}
1299
1300 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{MKU002}
1301
1302 Pendidikan Kewarganegaraan menjelaskan pentingnya pemahaman tentang identitas nasional
1303 Indonesia, hak dan kewajiban warga negara Indonesia serta hubungannya dengan hak dan
1304 kewajiban asasi manusia. Materi kuliah mencakup juga wawasan nusantara, ketahanan nasional,
1305 politik dan strategi nasional, serta implementasinya dalam kehidupan bermasyarakat, berbangsa
1306 dan bernegara kesatuan Republik Indonesia.
1307
1308 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF462}
1309
1310
1311
1312
1313
1314 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF360 implements HasPrasyarat}
1315
1316
1317
1318 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1319 \begin{itemize}
1320 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1321 \end{itemize}
1322 \textbf{Parameter:}
1323 \begin{itemize}
1324 \item \texttt{Mahasiswa mahasiswa} -
1325 \item \texttt{java.util.List reasonsContainer} -
1326 \end{itemize}
1327 \textbf{Return Value}: Tidak memiliki \textit{return value}
1328
1329 \textbf{Exception}: Tidak memiliki \textit{exception}
1330
1331 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1332
1333 \end{itemize}
1334 \item \texttt{AIF337}
1335
1336
1337
1338 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF458 implements HasPrasyarat}
1339
1340
1341
1342 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1343 \begin{itemize}
1344 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1345 \end{itemize}
1346 \textbf{Parameter:}
1347 \begin{itemize}
1348 \item \texttt{Mahasiswa mahasiswa} -
1349 \item \texttt{java.util.List reasonsContainer} -
1350 \end{itemize}
1351 \textbf{Return Value}: Tidak memiliki \textit{return value}
1352
1353 \textbf{Exception}: Tidak memiliki \textit{exception}
1354
1355 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1356
1357 \end{itemize}

```



---

```

1358 \item \texttt{AIF301 implements HasPrasyarat}
1359
1360 Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar tentang *sistem
1361 cerdas dan komponen-komponennya. " "Terdapat 4 topik utama yang dibahas yaitu
1362 teknik pencarian untuk *penyelesaian masalah, representasi pengetahuan dalam
1363 sistem *cerdas, pemodelan ketidakpastian dalam masalah dan teknik
1364 pembelajaran mesin.
1365
1366 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1367 \begin{itemize}
1368 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1369
1370 \textbf{Parameter:}
1371 \begin{itemize}
1372 \item \texttt{Mahasiswa mahasiswa} -
1373 \item \texttt{java.util.List reasonsContainer} -
1374 \end{itemize}
1375 \textbf{Return Value:} Tidak memiliki \textit{return value}
1376
1377 \textbf{Exception:} Tidak memiliki \textit{exception}
1378
1379 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1380
1381 \end{itemize}
1382 \item \texttt{AIF182}
1383
1384
1385
1386 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{ESM204}
1387
1388
1389
1390 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF205 implements HasPrasyarat}
1391
1392 Mata kuliah ini memperkenalkan kepada mahasiswa arsitektur komputer
1393 sederhana, modern, dan Advance. Perbedaan, kelebihan dan kekurangan untuk
1394 masing-masing arsitektur. Selain itu mahasiswa juga mempelajari cara kerja
1395 dari komponen-komponen komputer, terutama memory, cache, system BUS dan
1396 input/output.
1397
1398 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1399 \begin{itemize}
1400 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1401
1402 \textbf{Parameter:}
1403 \begin{itemize}
1404 \item \texttt{Mahasiswa mahasiswa} -
1405 \item \texttt{java.util.List reasonsContainer} -
1406 \end{itemize}
1407 \textbf{Return Value:} Tidak memiliki \textit{return value}
1408
1409 \textbf{Exception:} Tidak memiliki \textit{exception}
1410
1411 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1412
1413 \end{itemize}
1414 \item \texttt{AIF317 implements HasPrasyarat}
1415
1416
1417
1418 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1419 \begin{itemize}
1420 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1421
1422 \textbf{Parameter:}
1423 \begin{itemize}
1424 \item \texttt{Mahasiswa mahasiswa} -
1425 \item \texttt{java.util.List reasonsContainer} -
1426 \end{itemize}
1427 \textbf{Return Value:} Tidak memiliki \textit{return value}
1428
1429 \textbf{Exception:} Tidak memiliki \textit{exception}
1430
1431 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1432
1433 \end{itemize}
1434 \item \texttt{AIF383}
1435
1436
1437
1438 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF442 implements HasPraktikum}
1439
1440 \item \texttt{AIF442 implements HasPrasyarat}
1441
1442
1443
1444 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1445 \begin{itemize}
1446 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1447
1448 \textbf{Parameter:}
1449 \begin{itemize}
1450 \item \texttt{Mahasiswa mahasiswa} -
1451 \item \texttt{java.util.List reasonsContainer} -
1452 \end{itemize}
1453 \textbf{Return Value:} Tidak memiliki \textit{return value}
1454
1455 \textbf{Exception:} Tidak memiliki \textit{exception}
1456

```

```

1457 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1458
1459 \end{itemize}
1460 \item \texttt{ESM101}
1461
1462
1463
1464 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF403 implements HasPrasyarat}
1465
1466 1. Memberikan wawasan kepada mahasiswa tentang kemunculan dan pemanfaatan teknologi baru,
1467 khususnya yang berkaitan dengan komputer, dan dampaknya terhadap masyarakat luas.
1468 2. Memberikan kesadaran dan panduan bersikap kepada mahasiswa dalam menghadapi gejolak yang
1469 disebabkan oleh munculnya teknologi baru, khususnya yang berkaitan dengan komputer.
1470
1471 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1472 \begin{itemize}
1473 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1474
1475 \textbf{Parameter:}
1476 \begin{itemize}
1477 \item \texttt{Mahasiswa mahasiswa} -
1478 \item \texttt{java.util.List reasonsContainer} -
1479 \end{itemize}
1480 \textbf{Return Value}: Tidak memiliki \textit{return value}
1481
1482 \textbf{Exception}: Tidak memiliki \textit{exception}
1483
1484 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1485
1486 \end{itemize}
1487 \item \texttt{AIF402 implements HasPrasyarat}
1488
1489
1490
1491 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1492 \begin{itemize}
1493 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1494
1495 \textbf{Parameter:}
1496 \begin{itemize}
1497 \item \texttt{Mahasiswa mahasiswa} -
1498 \item \texttt{java.util.List reasonsContainer} -
1499 \end{itemize}
1500 \textbf{Return Value}: Tidak memiliki \textit{return value}
1501
1502 \textbf{Exception}: Tidak memiliki \textit{exception}
1503
1504 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1505
1506 \end{itemize}
1507 \item \texttt{AIF455}
1508
1509
1510
1511 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF443}
1512
1513
1514
1515 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF101 implements HasPraktikum}
1516
1517 Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar pemrograman
1518 seperti pengulangan dan percabangan, konsep dasar penyimpanan data kontigu
1519 menggunakan array, konsep dasar pemrograman berorientasi objek seperti kelas
1520 & objek, method, dll, termasuk di dalamnya 4 prinsip dasar pemrograman
1521 berorientasi objek : data abstraction, encapsulation, inheritance dan
1522 polymorphism. Selain, itu diberikan masalah-masalah komputasi sederhana
1523 yang harus diselesaikan menggunakan konsep-konsep yang sudah diperkenalkan
1524 dan mengimplementasikannya menggunakan bahasa pemrograman Java
1525
1526 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF382}
1527
1528
1529
1530 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF480}
1531
1532
1533
1534 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF316 implements HasPrasyarat}
1535
1536 \item \texttt{AIF316 implements HasPraktikum}
1537
1538 Mata kuliah ini memperkenalkan konsep-konsep dasar komputasi paralel, dimana sebuah
1539 program yang berjalan secara paralel harus memiliki safety property dan liveness property.
1540 Mahasiswa dikenalkan dengan beberapa teknik pemrograman multi-thread
1541 seperti lock, monitor, barrier, thread pool, dan sebagainya, yang diimplementasikan
1542 dalam bahasa pemrograman Java. Mahasiswa juga dikenalkan dengan beberapa metode untuk
1543 menganalisis kebenaran program baik secara matematis maupun secara praktis dengan bantuan
1544 model checker.
1545
1546 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1547 \begin{itemize}
1548 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1549
1550 \textbf{Parameter:}
1551 \begin{itemize}
1552 \item \texttt{Mahasiswa mahasiswa} -
1553 \item \texttt{java.util.List reasonsContainer} -
1554 \end{itemize}
1555 \textbf{Return Value}: Tidak memiliki \textit{return value}

```

```

1556 \textbf{Exception}: Tidak memiliki \textit{exception}
1557
1558
1559 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1560
1561 \end{itemize}
1562 \item \texttt{AIF438 implements HasPrasyarat}
1563
1564 Mata kuliah ini: Memperkenalkan karakteristik dan teknik visualisasi dari
1565 berbagai jenis data yang dapat dianalisis dengan teknik-teknik data mining;
1566 mempelajari teknik-teknik penyiapan data untuk berbagai jenis data dan teknik
1567 data mining; mempraktekkan teknik-teknik penyiapan data untuk menganalisis
1568 data nyata/simulasi dengan memanfaatkan perangkat lunak aplikasi.
1569
1570 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1571 \begin{itemize}
1572 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1573
1574 \textbf{Parameter}:
1575 \begin{itemize}
1576 \item \texttt{Mahasiswa mahasiswa} -
1577 \item \texttt{java.util.List reasonsContainer} -
1578 \end{itemize}
1579 \textbf{Return Value}: Tidak memiliki \textit{return value}
1580
1581 \textbf{Exception}: Tidak memiliki \textit{exception}
1582
1583 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1584
1585 \end{itemize}
1586 \item \texttt{AIF204 implements HasPrasyarat}
1587
1588 \item \texttt{AIF204 implements HasPraktikum}
1589
1590 Mata kuliah ini memperkenalkan konsep dan arsitektur DBMS, mengajarkan
1591 aljabar relasional dan SQL serta pemanfaatannya pada pemrograman kueri
1592 sederhana s/d relatif kompleks. Selain itu, mata kuliah ini juga mengajarkan
1593 dan mempraktekkan perancangan basisdata untuk masalah sederhana
1594 (lingkup kecil) termasuk pengembangan program aplikasinya;
1595
1596 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1597 \begin{itemize}
1598 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1599
1600 \textbf{Parameter}:
1601 \begin{itemize}
1602 \item \texttt{Mahasiswa mahasiswa} -
1603 \item \texttt{java.util.List reasonsContainer} -
1604 \end{itemize}
1605 \textbf{Return Value}: Tidak memiliki \textit{return value}
1606
1607 \textbf{Exception}: Tidak memiliki \textit{exception}
1608
1609 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1610
1611 \end{itemize}
1612 \item \texttt{AIF341 implements HasPraktikum}
1613
1614 Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar
1615 jaringan dan aplikasinya di kehidupan sehari-hari. Mahasiswa
1616 dikenalkan dengan teknologi-teknologi terbaru di bidang jaringan,
1617 sehingga mahasiswa memiliki pengetahuan yang dapat digunakan
1618 dalam kehidupan sehari-hari. Selain itu mahasiswa juga
1619 diperkenalkan dengan NetAcad, sebuah layanan dari Cisco yang
1620 dapat digunakan untuk memenuhi segala macam kebutuhan terkait
1621 dengan Cisco Academy.
1622
1623 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF183}
1624
1625
1626
1627 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AKS124}
1628
1629
1630
1631 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF459}
1632
1633
1634
1635 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF336}
1636
1637 Mata kuliah ini merupakan mata kuliah lanjutan dari mata kuliah Keamanan
1638 Informasi, dengan titik berat pada materi kriptografi. Mata kuliah ini
1639 memperkenalkan tambahan konsep kriptografi, misalnya tentang otentikasi
1640 yaitu otentikasi entitas, manajemen kunci, dan bentuk lain dari metode
1641 merahasiakan pesan, yaitu dengan menggunakan secret sharing. Selanjutnya,
1642 diperkenalkan juga penggunaan kriptografi pada protokol-protokol yang
1643 sebenarnya banyak digunakan sehari-hari, misalnya pada e-cash, auction,
1644 dan electronic voting.
1645
1646 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF463}
1647
1648
1649
1650 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF208 implements HasPrasyarat}
1651
1652 Mata kuliah ini memperkenalkan kepada mahasiswa tahapan rekayasa perangkat
1653 lunak, terutama dengan paradigma berorientasi objek, dilengkapi dengan
1654 pengenalan tentang manajemen proyek perangkat lunak.

```

```

1655 Selain, itu diberikan deskripsi proyek berskala kecil yang harus dikerjakan
1656 oleh mahasiswa dalam kelompok dengan menerapkan teori yang telah
1657 dipelajarinya.
1658
1659 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1660 \begin{itemize}
1661 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1662
1663 \textbf{Parameter:}
1664 \begin{itemize}
1665 \item \texttt{Mahasiswa mahasiswa} -
1666 \item \texttt{java.util.List reasonsContainer} -
1667 \end{itemize}
1668 \textbf{Return Value:} Tidak memiliki \textit{return value}
1669
1670 \textbf{Exception:} Tidak memiliki \textit{exception}
1671
1672 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1673
1674 \end{itemize}
1675 \item \texttt{MKU003}
1676
1677 Mata kuliah ini membentuk karakteristik mahasiswa sebagai manusia yang memiliki religiusitas
1678 melalui pendalaman akan makna agama dan beragama, mendeteksi dinamika Wahyu Tuhan dan iman
1679 mereka, memahami relasi dengan Tuhan dan sesama, mengenal makna keselamatan dalam konteks
1680 Kerajaan Allah, dan mampu menyatakan ajaran Gereja dalam pelayanan terhadap orang miskin dan
1681 terlanjar.
1682
1683 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{APS309}
1684
1685 APS302 atau APS309 ?
1686
1687 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF335}
1688
1689
1690
1691 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF362 implements HasPrasyarat}
1692
1693
1694
1695 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1696 \begin{itemize}
1697 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1698
1699 \textbf{Parameter:}
1700 \begin{itemize}
1701 \item \texttt{Mahasiswa mahasiswa} -
1702 \item \texttt{java.util.List reasonsContainer} -
1703 \end{itemize}
1704 \textbf{Return Value:} Tidak memiliki \textit{return value}
1705
1706 \textbf{Exception:} Tidak memiliki \textit{exception}
1707
1708 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1709
1710 \end{itemize}
1711 \item \texttt{AIF460}
1712
1713
1714
1715 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF358}
1716
1717
1718
1719 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AMS100}
1720
1721 Sistem Bilangan, Fungsi, Limit dan Kekontinuan Fungsi, Turunan, Integral,
1722 Penggunaan Integral, Sistem Persamaan Linear, Determinan, Vektor, Nilai dan
1723 Vektor Eigen.
1724
1725 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF401 implements HasPrasyarat}
1726
1727
1728
1729 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1730 \begin{itemize}
1731 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1732
1733 \textbf{Parameter:}
1734 \begin{itemize}
1735 \item \texttt{Mahasiswa mahasiswa} -
1736 \item \texttt{java.util.List reasonsContainer} -
1737 \end{itemize}
1738 \textbf{Return Value:} Tidak memiliki \textit{return value}
1739
1740 \textbf{Exception:} Tidak memiliki \textit{exception}
1741
1742 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1743
1744 \end{itemize}
1745 \item \texttt{AIF456}
1746
1747
1748
1749 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{SIR104}
1750
1751
1752
1753 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF339 implements HasPrasyarat}

```

```

1754
1755
1756
1757 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1758 \begin{itemize}
1759 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1760
1761 \textbf{Parameter:}
1762 \begin{itemize}
1763 \item \texttt{Mahasiswa mahasiswa} -
1764 \item \texttt{java.util.List reasonsContainer} -
1765 \end{itemize}
1766 \textbf{Return Value:} Tidak memiliki \textit{return value}
1767
1768 \textbf{Exception:} Tidak memiliki \textit{exception}
1769
1770 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1771
1772 \end{itemize}
1773 \item \texttt{AIF102 implements HasPrasyarat}
1774
1775 \item \texttt{AIF102 implements HasPraktikum}
1776
1777 Mata kuliah ini memperkenalkan berbagai algoritma dan teknik-teknik
1778 penyelesaian masalah komputasi seperti rekursif, sorting, teknik divide dan
1779 conquer, serta exhaustive search. Selain itu, pada kuliah ini juga
1780 dikenalkan berbagai struktur data yang dapat digunakan untuk mendukung
1781 penyelesaian masalah komputasi seperti ADT List, Stack dan Queue. Baik
1782 algoritma maupun struktur data yang dikenalkan harus dapat diimplementasikan
1783 dan digunakan oleh mahasiswa untuk menyelesaikan masalah dengan menggunakan
1784 suatu bahasa pemrograman berorientasi objek.
1785
1786 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1787 \begin{itemize}
1788 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1789
1790 \textbf{Parameter:}
1791 \begin{itemize}
1792 \item \texttt{Mahasiswa mahasiswa} -
1793 \item \texttt{java.util.List reasonsContainer} -
1794 \end{itemize}
1795 \textbf{Return Value:} Tidak memiliki \textit{return value}
1796
1797 \textbf{Exception:} Tidak memiliki \textit{exception}
1798
1799 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1800
1801 \end{itemize}
1802 \item \texttt{AIF381}
1803
1804
1805
1806 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF483}
1807
1808
1809
1810 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF315 implements HasPrasyarat}
1811
1812 \item \texttt{AIF315 implements HasPraktikum}
1813
1814 Mata kuliah ini memperkenalkan konsep dan lingkungan pemrograman berbasis web,
1815 kemudian belajar membuat aplikasi berbasis web menggunakan HTML5, CSS, Java Script
1816 dan PHP. Untuk meningkatkan keterampilan pemrograman dilengkapi dengan praktikum.
1817 Sedangkan untuk mendapatkan pengalaman penerapan konsep diberikan tugas besar membuat
1818 program berbasis web dengan kasus yang ditentukan oleh mahasiswa.
1819
1820 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1821 \begin{itemize}
1822 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1823
1824 \textbf{Parameter:}
1825 \begin{itemize}
1826 \item \texttt{Mahasiswa mahasiswa} -
1827 \item \texttt{java.util.List reasonsContainer} -
1828 \end{itemize}
1829 \textbf{Return Value:} Tidak memiliki \textit{return value}
1830
1831 \textbf{Exception:} Tidak memiliki \textit{exception}
1832
1833 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1834
1835 \end{itemize}
1836 \item \texttt{AIF342 implements HasPraktikum}
1837
1838 \item \texttt{AIF342 implements HasPrasyarat}
1839
1840
1841
1842 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1843 \begin{itemize}
1844 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1845
1846 \textbf{Parameter:}
1847 \begin{itemize}
1848 \item \texttt{Mahasiswa mahasiswa} -
1849 \item \texttt{java.util.List reasonsContainer} -
1850 \end{itemize}
1851 \textbf{Return Value:} Tidak memiliki \textit{return value}
1852

```

```

1853 \textbf{Exception}: Tidak memiliki \textit{exception}
1854
1855 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1856
1857 \end{itemize}
1858 \item \texttt{IIE214}
1859
1860
1861
1862 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF303 implements HasPrasyarat}
1863
1864 Mempelajari Konsep Data, Informasi, Pengetahuan, Sistem Informasi, proses dan
1865 pemodelan bisnis, jenis-jenis sistem informasi, untuk mendukung pengambilan
1866 keputusan. Mempelajari trend Teknologi Informasi, tahap-tahap pembangunan
1867 sistem informasi. Mempelajari pengantar : EIS, e-bisnis/e-commerce, Business
1868 Intelligence, Cloud Computing dan Mobile Applications
1869
1870 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1871 \begin{itemize}
1872 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1873
1874 \textbf{Parameter:}
1875 \begin{itemize}
1876 \item \texttt{Mahasiswa mahasiswa} -
1877 \item \texttt{java.util.List reasonsContainer} -
1878 \end{itemize}
1879 \textbf{Return Value}: Tidak memiliki \textit{return value}
1880
1881 \textbf{Exception}: Tidak memiliki \textit{exception}
1882
1883 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1884
1885 \end{itemize}
1886 \item \texttt{AIF302 implements HasPrasyarat}
1887
1888 Mata kuliah ini melatih mahasiswa dalam menulis ilmiah serta memperkenalkan
1889 metodologi penelitian serta kakas untuk menulis ilmiah.
1890
1891 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1892 \begin{itemize}
1893 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1894
1895 \textbf{Parameter:}
1896 \begin{itemize}
1897 \item \texttt{Mahasiswa mahasiswa} -
1898 \item \texttt{java.util.List reasonsContainer} -
1899 \end{itemize}
1900 \textbf{Return Value}: Tidak memiliki \textit{return value}
1901
1902 \textbf{Exception}: Tidak memiliki \textit{exception}
1903
1904 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1905
1906 \end{itemize}
1907 \item \texttt{AIF181}
1908
1909
1910
1911 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF210}
1912
1913
1914
1915 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF343}
1916
1917
1918
1919 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF206 implements HasPrasyarat}
1920
1921 Mata kuliah ini memperkenalkan kepada mahasiswa mengenai konsep sistem
1922 operasi, jenis-jenis sistem operasi yang digunakan dalam kehidupan
1923 sehari-hari dan beberapa perangkat keras yang dibutuhkan pada komputer.
1924 Selain itu juga mempelajari mengenai teknik dan algoritma yang digunakan
1925 dalam pengelolaan sistem operasi.
1926
1927 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1928 \begin{itemize}
1929 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1930
1931 \textbf{Parameter:}
1932 \begin{itemize}
1933 \item \texttt{Mahasiswa mahasiswa} -
1934 \item \texttt{java.util.List reasonsContainer} -
1935 \end{itemize}
1936 \textbf{Return Value}: Tidak memiliki \textit{return value}
1937
1938 \textbf{Exception}: Tidak memiliki \textit{exception}
1939
1940 \textbf{Override}: \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1941
1942 \end{itemize}
1943 \item \texttt{AIF314 implements HasPrasyarat}
1944
1945 \item \texttt{AIF314 implements HasPraktikum}
1946
1947 Kuliah ini merupakan kelanjutan dari kuliah Manajemen Informasi Basisdata.
1948 Pada perkuliahan ini, mahasiswa akan mempelajari teknik-teknik pengelolaan
1949 basis data dan membuat program dengan basis data yang optimal/efisien.
1950
1951 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.

```

```

1952 \begin{itemize}
1953 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1954
1955 \textbf{Parameter:}
1956 \begin{itemize}
1957 \item \texttt{Mahasiswa mahasiswa} -
1958 \item \texttt{java.util.List reasonsContainer} -
1959 \end{itemize}
1960 \textbf{Return Value:} Tidak memiliki \textit{return value}
1961
1962 \textbf{Exception:} Tidak memiliki \textit{exception}
1963
1964 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
1965
1966 \end{itemize}
1967 \item \texttt{AIF380}
1968
1969
1970
1971 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF103}
1972
1973 Mata kuliah ini merupakan salah satu cara untuk mencapai kompetensi dasar
1974 tentang matematika diskrit yang prinsipnya banyak digunakan dalam bidang
1975 ilmu komputer. Selain itu, kuliah ini juga merupakan cara untuk membentuk
1976 pola pikir logis yang dibutuhkan untuk menempuh kuliah-kuliah di tingkat
1977 yang lebih tinggi.
1978
1979 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF292}
1980
1981
1982
1983 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF441 implements HasPraktikum}
1984
1985 \item \texttt{AIF441 implements HasPrasyarat}
1986
1987 Mata kuliah ini memperkenalkan kepada mahasiswa konsep jaringan lanjut
1988 terutama di layer data link dan layer network. Materi utama dari mata kuliah
1989 ini adalah pengembangan jaringan dan pengenalan fungsi-fungsi yang terdapat
1990 pada alat jaringan Cisco yang berkaitan dengan layer 2 dan layer 3.
1991
1992 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
1993 \begin{itemize}
1994 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
1995
1996 \textbf{Parameter:}
1997 \begin{itemize}
1998 \item \texttt{Mahasiswa mahasiswa} -
1999 \item \texttt{java.util.List reasonsContainer} -
2000 \end{itemize}
2001 \textbf{Return Value:} Tidak memiliki \textit{return value}
2002
2003 \textbf{Exception:} Tidak memiliki \textit{exception}
2004
2005 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
2006
2007 \end{itemize}
2008 \item \texttt{AIF457 implements HasPrasyarat}
2009
2010 Mata kuliah ini memperkenalkan konsep kewirausahaan dengan memanfaatkan teknologi, khususnya
2011 teknologi informasi, sebagai basis usaha dan inovasi produk/jasa; Mempelajari
2012 teknik mencari peluang dan merumuskan bidang usaha spesifik yang akan
2013 diterjuni; Mempelajari konsep manajemen pemasaran, keuangan dan SDM dalam
2014 kaitannya dengan berwira-usaha di bidang TI; Menyusun proposal bisnis untuk
2015 berwira-usaha di bidang TI dan mempresentasikannya.
2016
2017 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
2018 \begin{itemize}
2019 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
2020
2021 \textbf{Parameter:}
2022 \begin{itemize}
2023 \item \texttt{Mahasiswa mahasiswa} -
2024 \item \texttt{java.util.List reasonsContainer} -
2025 \end{itemize}
2026 \textbf{Return Value:} Tidak memiliki \textit{return value}
2027
2028 \textbf{Exception:} Tidak memiliki \textit{exception}
2029
2030 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
2031
2032 \end{itemize}
2033 \item \texttt{MKU001}
2034
2035 Mata Kuliah Pendidikan Pancasila berupaya menelaah/mengkaji berbagai fenomena kehidupan
2036 bangsa dan Negara Indonesia sebagai sebuah ruang publik dengan menggunakan pendekatan
2037 hermeneutika (filsafat) dan pendidikan nilai (pedagogik). Dengan bantuan hermeneutika
2038 mahasiswa diajak berpikir kritis terhadap segala bentuk ideologisme Pancasila dan melalui
2039 pendidikan nilai mahasiswa dilatih untuk memiliki nilai Pancasila. Nilai pengembangan diri
2040 intra-personal dan relasi inter-personal dapat tertanam melalui pendidikan Pancasila yang
2041 tujuannya adalah membangun kepribadian (character building) manusia Indonesia yang utuh,
2042 baik menyangkut aspek kognitif, afektif, maupun psikomotor. Dengan demikian, Pendidikan
2043 Pancasila mengajak mahasiswa menilai realitas ruang publik sehari-hari secara mandiri
2044 dengan panduan nilai-nilai etis Pancasila.
2045
2046 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{ESA101}
2047
2048
2049
2050 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AMS290}

```



```

2051
2052
2053
2054 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF461}
2055
2056
2057
2058 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF318 implements HasPrasyarat}
2059
2060 \item \texttt{AIF318 implements HasPraktikum}
2061
2062 Mata kuliah ini memperkenalkan konsep perangkat mobile dan pemrograman pada perangkat
2063 mobile. Pemrograman dikhususkan pada lingkungan J2ME dan Android.
2064 Untuk meningkatkan keterampilan pemrograman dilengkapi dengan praktikum.
2065 Sedangkan untuk mendapatkan pengalaman penerapan konsep diberikan tugas implementasi suatu
2066 kasus pada lingkungan mobile-cloud dengan kasus yang sudah ditentukan.
2067
2068 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
2069 \begin{itemize}
2070 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
2071
2072 \textbf{Parameter:}
2073 \begin{itemize}
2074 \item \texttt{Mahasiswa mahasiswa} -
2075 \item \texttt{java.util.List reasonsContainer} -
2076 \end{itemize}
2077 \textbf{Return Value:} Tidak memiliki \textit{return value}
2078
2079 \textbf{Exception:} Tidak memiliki \textit{exception}
2080
2081 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
2082
2083 \end{itemize}
2084 \item \texttt{AIF334}
2085
2086
2087
2088 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{ESM105}
2089
2090
2091
2092 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF450}
2093
2094
2095
2096 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF446}
2097
2098
2099
2100 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF104}
2101
2102 Mata kuliah ini memberikan pengetahuan tentang logika yang digunakan di
2103 dalam ilmu komputer. Dalam kuliah ini, mahasiswa belajar untuk bisa
2104 memodelkan suatu kalimat dalam kehidupan sehari-hari, ke dalam kalimat
2105 dengan sintaks tertentu, yang hanya memiliki satu arti. Lalu, diperkenalkan
2106 juga, bagaimana mengartikan suatu kalimat (benar atau salah) dan bagaimana
2107 menentukan sifat dari kalimat tersebut.
2108
2109 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF387}
2110
2111
2112
2113 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF313 implements HasPraktikum}
2114
2115 Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar pembuatan grafik
2116 dengan menggunakan komputer seperti mengenal berbagai algoritma pembuatan
2117 primitif 2 dimensi seperti titik, garis, lingkaran, elips, berbagai macam
2118 bentuk kurva, fraktal, konsep warna (RGB), dasar-dasar grafika 3 dimensi
2119 seperti pewarnaan, pencahayaan, pemberian tekstur pada objek, transformasi,
2120 animasi, dan sebagainya. Selain, itu diberikan masalah-masalah komputasi
2121 sederhana yang harus diselesaikan menggunakan konsep-konsep yang sudah
2122 diperkenalkan dan mengimplementasikannya menggunakan bahasa pemrograman Java.
2123
2124 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF344 implements HasPrasyarat}
2125
2126
2127
2128 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
2129 \begin{itemize}
2130 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
2131
2132 \textbf{Parameter:}
2133 \begin{itemize}
2134 \item \texttt{Mahasiswa mahasiswa} -
2135 \item \texttt{java.util.List reasonsContainer} -
2136 \end{itemize}
2137 \textbf{Return Value:} Tidak memiliki \textit{return value}
2138
2139 \textbf{Exception:} Tidak memiliki \textit{exception}
2140
2141 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
2142
2143 \end{itemize}
2144 \item \texttt{AIF201 implements HasPrasyarat}
2145
2146 \item \texttt{AIF201 implements HasPraktikum}
2147
2148 \item \texttt{AIF201 implements HasResponsi}
2149

```



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. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.

```

\begin{itemize}
\item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
\textbf{Parameter:}
\begin{itemize}
\item \texttt{Mahasiswa mahasiswa} -
\item \texttt{java.util.List reasonsContainer} -
\end{itemize}
\textbf{Return Value:} Tidak memiliki \textit{return value}
\textbf{Exception:} Tidak memiliki \textit{exception}
\textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
\end{itemize}
\item \texttt{AIF352}

```

Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF305 implements HasPrasyarat}

Mata kuliah ini memperkenalkan kepada mahasiswa konsep dasar jaringan komputer dengan menggunakan top-down. Selain itu mengajarkan juga kepada mahasiswa mengenai 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. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.

```

\begin{itemize}
\item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
\textbf{Parameter:}
\begin{itemize}
\item \texttt{Mahasiswa mahasiswa} -
\item \texttt{java.util.List reasonsContainer} -
\end{itemize}
\textbf{Return Value:} Tidak memiliki \textit{return value}
\textbf{Exception:} Tidak memiliki \textit{exception}
\textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
\end{itemize}
\item \texttt{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. \item \texttt{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. \item \texttt{AIF332 implements HasPrasyarat}

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

```

\begin{itemize}
\item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
\textbf{Parameter:}
\begin{itemize}
\item \texttt{Mahasiswa mahasiswa} -
\item \texttt{java.util.List reasonsContainer} -
\end{itemize}
\textbf{Return Value:} Tidak memiliki \textit{return value}
\textbf{Exception:} Tidak memiliki \textit{exception}
\textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
\end{itemize}
\item \texttt{AIF304 implements HasPrasyarat}
\item \texttt{AIF304 implements HasPraktikum}
\item \texttt{AIF304 implements HasResponsi}

```

Mata kuliah ini memberikan kesempatan bagi mahasiswa untuk memperdalam konsep

```

2249 tentang pengembangan sistem informasi dan mempraktekkan analisis kebutuhan,
2250 analisis sistem dan perancangan sitem pada organisasi studi kasus;
2251
2252 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
2253 \begin{itemize}
2254 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
2255
2256 \textbf{Parameter:}
2257 \begin{itemize}
2258 \item \texttt{Mahasiswa mahasiswa} -
2259 \item \texttt{java.util.List reasonsContainer} -
2260 \end{itemize}
2261 \textbf{Return Value:} Tidak memiliki \textit{return value}
2262
2263 \textbf{Exception:} Tidak memiliki \textit{exception}
2264
2265 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
2266
2267 \end{itemize}
2268 \item \texttt{ESM201}
2269
2270
2271 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF312 implements HasPrasyarat}
2272
2273 \item \texttt{AIF312 implements HasPraktikum}
2274
2275 Mata kuliah ini memberikan pengetahuan awal tentang keamanan informasi. Pada
2276 beberapa pertemuan awal, dibahas keamanan informasi secara matematis, yaitu
2277 di materi-materi seputar kriptografi dan serangannya. Lalu, dibahas pula
2278 konsep keamanan informasi pada jaringan komputer dan pada software.
2279
2280 Kelas ini tidak memiliki atribut. \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
2281 \begin{itemize}
2282 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
2283
2284 \textbf{Parameter:}
2285 \begin{itemize}
2286 \item \texttt{Mahasiswa mahasiswa} -
2287 \item \texttt{java.util.List reasonsContainer} -
2288 \end{itemize}
2289 \textbf{Return Value:} Tidak memiliki \textit{return value}
2290
2291 \textbf{Exception:} Tidak memiliki \textit{exception}
2292
2293 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{MataKuliah}
2294
2295 \end{itemize}
2296 \item \texttt{AIF484}
2297
2298
2299
2300 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF191}
2301
2302
2303
2304 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF386}
2305
2306
2307
2308 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF105}
2309
2310 Mata kuliah ini memperkenalkan kepada mahasiswa terminologi dan konsep dasar
2311 yang akan banyak dipakai sepanjang kuliah di Teknik Informatika. Selain itu
2312 mata kuliah ini juga mempersiapkan dan membiasakan mahasiswa dengan suasana
2313 akademik yang khas perguruan tinggi seperti kedisiplinan, kerja sama,
2314 kemampuan menggunakan teknologi informasi dalam pembuatan tugas, kemampuan
2315 komunikasi, dsb.
2316
2317 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF294}
2318
2319
2320
2321 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF282}
2322
2323
2324
2325 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{AIF451}
2326
2327
2328
2329 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{APS302}
2330
2331
2332
2333 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{EAA101}
2334
2335
2336
2337 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{HasPrasyarat}
2338
2339 Mendefinisikan kelas-kelas yang memiliki prasyarat, terkustomisasi
2340 untuk seorang \texttt{Mahasiswa}. Jika ada tambahan, jangan lupa untuk
2341 mendaftarkannya di \texttt{DEFAULT\_HASPRASYARAT\_CLASSES}. Jika berubah package,
2342 jangan lupa mengupdate \texttt{DEFAULT\_PRASYARAT\_PACKAGE}.
2343
2344 Atribut yang dimiliki kelas ini adalah sebagai berikut.
2345 \begin{itemize}
2346 \item \texttt{String DEFAULT\_HASPRASYARAT\_CLASSES} - Daftar dari nama kelas default seluruh turunan interface ini. Perlu

```

```

2348     manual, karena Java reflection tidak dapat mendeteksi otomatis.
2349 \item \texttt{String DEFAULT\_PRASYARAT\_PACKAGE} - Package tempat menyimpan seluruh turunan standar interface ini. Perlu
    didefinisikan
2350 manual, karena Java reflection tidak dapat mendeteksi otomatis.
2351 \end{itemize}
2352 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
2353 \begin{itemize}
2354 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}Memeriksa
    prasyarat-prasyarat dari kuliah, spesifik untuk mahasiswa
2355 yang dituju. Jika ada pesan-pesan khusus, akan ditambahkan pada parameter
2356 reasonsContainer.
2357
2358 \textbf{Parameter:}
2359 \begin{itemize}
2360 \item \texttt{Mahasiswa mahasiswa} -
2361 prasyarat kuliah akan diperiksa spesifik pada mahasiswa ini
2362 \item \texttt{List reasonsContainer} -
2363 pesan-pesan terkait prasyarat akan ditambahkan di sini, jika ada.
2364 \end{itemize}
2365 \textbf{Return Value:} true jika seluruh prasyarat dipenuhi, false jika tidak.
2366
2367 \textbf{Exception:} Tidak memiliki \textit{exception}
2368
2369 \end{itemize}
2370 \item \texttt{HasPraktikum}
2371
2372
2373
2374 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{HasResponsi}
2375
2376
2377
2378 Kelas ini tidak memiliki atribut. Kelas ini tidak memiliki method. \item \texttt{Kelulusan implements HasPrasyarat}
2379
2380
2381
2382 Atribut yang dimiliki kelas ini adalah sebagai berikut.
2383 \begin{itemize}
2384 \item \texttt{String PILIHAN\_WAJIB} -
2385 \item \texttt{String WAJIB} -
2386 \item \texttt{String AGAMA} -
2387 \item \texttt{int MIN\_SKS\_LULUS} -
2388 \item \texttt{int MIN\_PILIHAN\_WAJIB} -
2389 \end{itemize}
2390 \textit{Method-method} yang dimiliki kelas ini adalah sebagai berikut.
2391 \begin{itemize}
2392 \item \texttt{public boolean checkPrasyarat(id.ac.unpar.siamodels.Mahasiswa mahasiswa, java.util.List reasonsContainer)}
2393
2394 \textbf{Parameter:}
2395 \begin{itemize}
2396 \item \texttt{Mahasiswa mahasiswa} -
2397 \item \texttt{java.util.List reasonsContainer} -
2398 \end{itemize}
2399 \textbf{Return Value:} Tidak memiliki \textit{return value}
2400
2401 \textbf{Exception:} Tidak memiliki \textit{exception}
2402
2403 \textbf{Override:} \texttt{checkPrasyarat} dari kelas \texttt{Object}
2404
2405 \end{itemize}
2406 \end{enumerate}
2407 \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 implements Comparable

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<sub>i</sub> <sub>j</sub>li<sub>i</sub>2 digit pertama berupa tahun, 2 digit terakhir <sub>j</sub>li<sub>i</sub> digit terakhir: 1 untuk ganjil, 2 untuk genap, 4 untuk pendek. <sub>i</sub>/ul<sub>i</sub>

*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: `juli` `li` Kuliah yang tidak lulus tidak dihitung `li` Jika pengambilan beberapa kali, diambil `jem` nilai terbaik `j/em`. `i/ul` 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: `juli` `li` Kuliah yang tidak lulus tidak dihitung `li` Jika pengambilan beberapa kali, diambil `jem` nilai terbaik `j/em`. `i/ul` 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: `juli` `li` Perhitungan kuliah yang tidak lulus ditentukan parameter `li` Jika pengambilan beberapa kali, diambil `jem` nilai terbaik `j/em`. `i/ul`

**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: `juli` `li` Jika pengambilan beberapa kali, diambil semua. `i/ul` 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  $\text{jem}_i$  nilai terbaik  $i/\text{em}_i$ .

**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:  $\sum_i \text{li}_i$  Kuliah yang tidak lulus  $\text{jem}_i$  dihitung  $i/\text{em}_i$ . 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` implements `Comparator`  
Pembanding 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.LocalTime 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 setName(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 implements HasPrasyarat
- 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 implements HasPrasyarat

15. AIF311 implements HasPraktikum

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`

16. AIF192

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

17. AIF468

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

18. IIE103

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

19. AIF385

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

20. 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.

21. AIF281

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

22. EAA102

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

23. AIF405 implements HasPrasyarat

24. AIF405 implements HasPraktikum

Mata kuliah ini merupakan lanjutan dari Projek Sistem Informasi 1 dan memberikan kesempatan bagi mahasiswa untuk melanjutkan/mengembangkan perancangan sitem 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

25. APS182

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

26. 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. Kesiediaan 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.

27. 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.

28. AIF389

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

29. AMS190

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

30. AMS191

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

31. AMS200

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

32. AIF330

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

33. AIF388

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

34. AIF465

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

35. AIF453 implements HasPrasyarat

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`

36. AIF280

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

37. AIF445 implements HasPrasyarat

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

38. AIF469 implements HasPrasyarat

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

39. AIF486

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

40. IIE207

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

41. AIF202 implements HasPrasyarat

42. AIF202 implements HasResponsi

43. AIF202 implements HasPraktikum

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

44. AIF347

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

45. 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.

46. ESM203

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

47. APS402 implements HasPrasyarat

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

#### 48. AIF306 implements HasPrasyarat

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

#### 49. AKS122

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

#### 50. 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.

#### 51. AIF462

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

#### 52. AIF360 implements HasPrasyarat

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

#### 53. AIF337

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

#### 54. AIF458 implements HasPrasyarat

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. AIF301 implements HasPrasyarat

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

56. AIF182

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

57. ESM204

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

58. AIF205 implements HasPrasyarat

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

59. AIF317 implements HasPrasyarat

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. AIF383

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

61. AIF442 implements HasPraktikum

62. AIF442 implements HasPrasyarat

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

#### 63. ESM101

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

#### 64. AIF403 implements HasPrasyarat

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

#### 65. AIF402 implements HasPrasyarat

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

66. AIF455

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

67. AIF443

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

68. AIF101 implements HasPraktikum

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.

69. AIF382

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

70. AIF480

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

71. AIF316 implements HasPrasyarat

72. AIF316 implements HasPraktikum

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`

73. AIF438 implements `HasPrasyarat`

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`

74. AIF204 implements `HasPrasyarat`

75. AIF204 implements `HasPraktikum`

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`

76. AIF341 implements HasPraktikum

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.

77. AIF183

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

78. AKS124

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

79. AIF459

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

80. 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-protolol yang sebenarnya banyak digunakan sehari-hari, misalnya pada e-cash, auction, dan electronic voting.

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

81. AIF463

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

82. AIF208 implements HasPrasyarat

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`

#### 83. 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.

#### 84. APS309

APS302 atau APS309 ?

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

#### 85. AIF335

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

#### 86. AIF362 implements HasPrasyarat

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. AIF460

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

#### 88. AIF358

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

89. 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.

90. AIF401 implements HasPrasyarat

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. AIF456

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

92. SIR104

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

93. AIF339 implements HasPrasyarat

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. AIF102 implements HasPrasyarat

95. AIF102 implements HasPraktikum

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

96. AIF381

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

97. AIF483

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

98. AIF315 implements HasPrasyarat

99. AIF315 implements HasPraktikum

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

100. AIF342 implements HasPraktikum

101. AIF342 implements HasPrasyarat

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

102. IIE214

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

103. AIF303 implements HasPrasyarat

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

104. AIF302 implements HasPrasyarat

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

105. AIF181

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

106. AIF210

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

107. AIF343

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

108. AIF206 implements HasPrasyarat

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

109. AIF314 implements HasPrasyarat

110. AIF314 implements HasPraktikum

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`

#### 111. AIF380

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

#### 112. 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.

#### 113. AIF292

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

#### 114. AIF441 implements HasPraktikum

#### 115. AIF441 implements HasPrasyarat

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`

116. AIF457 implements HasPrasyarat

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`

117. 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.

118. ESA101

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

119. AMS290

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

120. AIF461

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

121. AIF318 implements HasPrasyarat

122. AIF318 implements HasPraktikum

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

123. AIF334

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

124. ESM105

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

125. AIF450

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

126. AIF446

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

127. 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.

128. AIF387

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

129. AIF313 implements HasPraktikum

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.

130. AIF344 implements HasPrasyarat

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

131. AIF201 implements HasPrasyarat

132. AIF201 implements HasPraktikum

133. AIF201 implements HasResponsi

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

134. AIF352

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

135. AIF305 implements HasPrasyarat

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

136. 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.

137. MKU011

Mata kuliah estetika memberi pemahaman konseptual filosofis "seni" dalam khasanah keilmuan, pembentukan kesadaran ekologis juga dalam proses kebudayaan 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.

#### 138. AIF332 implements HasPrasyarat

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

#### 139. AIF304 implements HasPrasyarat

#### 140. AIF304 implements HasPraktikum

#### 141. AIF304 implements HasResponsi

Mata kuliah ini memberikan kesempatan bagi mahasiswa untuk memperdalam konsep tentang pengembangan sistem informasi dan mempraktekkan analisis kebutuhan, analisis sistem dan perancangan sitem 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

#### 142. ESM201

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

#### 143. AIF312 implements HasPrasyarat

144. AIF312 implements HasPraktikum

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`

145. AIF484

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

146. AIF191

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

147. AIF386

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

148. 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.

149. AIF294

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

150. AIF282

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

151. AIF451

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

152. APS302

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

153. EAA101

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

154. 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*

155. HasPraktikum

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

156. HasResponsi

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

157. Kelulusan implements HasPrasyarat

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

## **LAMPIRAN D**

### **HASIL PDF TEXDOCLET**

Hasil PDF dapat dilihat pada halaman selanjutnya.

# TeXDoclet Java Documentation

Created with Javadoc TeXDoclet Doclet

Greg Wonderly

Søren Caspersen

Stefan Marx

October 3, 2012

# Contents

<b>Class Hierarchy</b>	<b>2</b>
<b>1 Package org.stfm.texdoclet</b>	<b>3</b>
1.1 Interface ClassFilter	4
1.1.1 Declaration	4
1.1.2 All known subinterfaces	4
1.1.3 All classes known to implement interface	4
1.1.4 Method summary	4
1.1.5 Methods	4
1.2 Class ClassHierachy	4
1.2.1 Declaration	4
1.2.2 Field summary	4
1.2.3 Constructor summary	4
1.2.4 Method summary	5
1.2.5 Fields	5
1.2.6 Constructors	5
1.2.7 Methods	5
1.3 Class HelpOutput	5
1.3.1 Declaration	5
1.3.2 Constructor summary	6
1.3.3 Method summary	6
1.3.4 Constructors	6
1.3.5 Methods	6
1.4 Class HTMLtoLaTeXBackEnd	6
1.4.1 See also	6
1.4.2 Declaration	6
1.4.3 Constructor summary	6
1.4.4 Method summary	6
1.4.5 Constructors	7
1.4.6 Methods	7
1.4.7 Members inherited from class HTMLEditorKit.ParserCallback	8
1.5 Class InterfaceHierachy	8
1.5.1 Declaration	8
1.5.2 Field summary	8
1.5.3 Constructor summary	8
1.5.4 Method summary	8

1.5.5	Fields	8
1.5.6	Constructors	8
1.5.7	Methods	8
1.6	Class <code>MarkdownTest</code>	9
1.6.1	Declaration	13
1.6.2	Constructor summary	13
1.6.3	Constructors	13
1.7	Class <code>Package</code>	13
1.7.1	See also	13
1.7.2	Declaration	13
1.7.3	Field summary	14
1.7.4	Constructor summary	14
1.7.5	Method summary	14
1.7.6	Fields	14
1.7.7	Constructors	14
1.7.8	Methods	15
1.8	Class <code>TableInfo</code>	15
1.8.1	Declaration	16
1.8.2	Constructor summary	16
1.8.3	Method summary	16
1.8.4	Constructors	17
1.8.5	Methods	17
1.9	Class <code>TestFilter</code>	18
1.9.1	Declaration	18
1.9.2	Constructor summary	18
1.9.3	Method summary	18
1.9.4	Constructors	19
1.9.5	Methods	19
1.10	Class <code>TeXDoclet</code>	19
1.10.1	See also	21
1.10.2	Declaration	21
1.10.3	Field summary	21
1.10.4	Constructor summary	22
1.10.5	Method summary	22
1.10.6	Fields	22
1.10.7	Constructors	22
1.10.8	Methods	22
1.10.9	Members inherited from class <code>Doclet</code>	23



# Class Hierarchy

## Classes

- `java.lang.Object`
  - `com.sun.javadoc.Doclet`
    - `org.stfm.texdoclet.TeXDoclet` (in [1.10](#), page [19](#))
  - `javax.swing.text.html.HTMLEditorKit.ParserCallback`
    - `org.stfm.texdoclet.HTMLtoLaTeXBackEnd` (in [1.4](#), page [6](#))
  - `org.stfm.texdoclet.ClassHierarchy` (in [1.2](#), page [4](#))
  - `org.stfm.texdoclet.HelpOutput` (in [1.3](#), page [5](#))
  - `org.stfm.texdoclet.InterfaceHierarchy` (in [1.5](#), page [8](#))
  - `org.stfm.texdoclet.MarkdownTest` (in [1.6](#), page [9](#))
  - `org.stfm.texdoclet.Package` (in [1.7](#), page [13](#))
  - `org.stfm.texdoclet.TableInfo` (in [1.8](#), page [15](#))
  - `org.stfm.texdoclet.TestFilter` (in [1.9](#), page [18](#))

## Interfaces

- `org.stfm.texdoclet.ClassFilter` (in [1.1](#), page [4](#))

# Chapter 1

## Package org.stfm.texdoclet

<i>Package Contents</i>	<i>Page</i>
<b>Interfaces</b>	
<b>ClassFilter</b> .....	4
This interface can be implemented and a class name provided to the Doclet to filter which classes are and are not included in the output document.	
<b>Classes</b>	
<b>ClassHierarchy</b> .....	4
Manages and prints a class hierarchy.	
<b>HelpOutput</b> .....	5
<b>HTMLtoLaTeXBackEnd</b> .....	6
This class implements a <code>ParserCallback</code> that translates HTML to the corresponding $\text{\LaTeX}$ .	
<b>InterfaceHierarchy</b> .....	8
Manages and prints a interface hierarchy.	
<b>MarkdownTest</b> .....	9
This class is just for testing the Markdown processing output.	
<b>Package</b> .....	13
This class is used to manage the contents of a Java package.	
<b>TableInfo</b> .....	15
This class provides support for converting HTML tables into $\text{\LaTeX}$ tables.	
<b>TestFilter</b> .....	18
This class filters out classes beginning with "Test" when applied to the Doclet.	
<b>TeXDoclet</b> .....	19
This class provides a Java <code>javadoc</code> Doclet which generates a $\text{\LaTeX}$ 2 $\epsilon$ document out of the java classes that it is used on.	

This doclet is based on the doclet originally created by Greg Wonderly of [C2 technologies Inc.](http://www.c2technologies.com) and its revision by [XO Software](http://www.xo-software.com). The project of Greg Wonderly is available here : <http://java.net/projects/texdoclet>.

## 1.1 Interface ClassFilter

This interface can be implemented and a class name provided to the Doclet to filter which classes are and are not included in the output document.

### 1.1.1 Declaration

```
public interface ClassFilter
```

### 1.1.2 All known subinterfaces

TestFilter (in [1.9](#), page [18](#))

### 1.1.3 All classes known to implement interface

TestFilter (in [1.9](#), page [18](#))

### 1.1.4 Method summary

[includeClass\(ClassDoc\)](#) Filters the ClassDoc passed.

### 1.1.5 Methods

- **includeClass**  
`boolean includeClass(com.sun.javadoc.ClassDoc cd)`
  - **Description**  
Filters the ClassDoc passed. If true is returned, the passed class will be included into the output. If false is returned, this document will not be included.

## 1.2 Class ClassHierarchy

Manages and prints a class hierarchy. Use `add` to add another class to the hierarchy. Use `printTree` to print the corresponding L<sup>A</sup>T<sub>E</sub>X.

### 1.2.1 Declaration

```
public class ClassHierarchy  
extends java.lang.Object
```

### 1.2.2 Field summary

[root](#)

### 1.2.3 Constructor summary

[ClassHierarchy\(\)](#) Creates new ClassHierarchy

### 1.2.4 Method summary

**add(ClassDoc)** Adds another class to the hierarchy  
**printBranch(RootDoc, SortedMap, double, double)** Prints a branch of the tree.  
**printTree(RootDoc, double)** Prints the L<sup>A</sup>T<sub>E</sub>X corresponding to the tree.

### 1.2.5 Fields

- public java.util.SortedMap **root**

### 1.2.6 Constructors

- **ClassHierarchy**  
 public **ClassHierarchy()**
  - **Description**  
 Creates new **ClassHierarchy**

### 1.2.7 Methods

- **add**  
 protected java.util.SortedMap **add**(com.sun.javadoc.ClassDoc **cls**)
  - **Description**  
 Adds another class to the hierarchy
- **printBranch**  
 protected void **printBranch**(com.sun.javadoc.RootDoc **rootDoc**, java.util.SortedMap **map**, double **indent**, double **overviewindent**)
  - **Description**  
 Prints a branch of the tree. The branch is printed using **TeXDoclet.os**.
- **printTree**  
 public void **printTree**(com.sun.javadoc.RootDoc **rootDoc**, double **overviewindent**)
  - **Description**  
 Prints the L<sup>A</sup>T<sub>E</sub>X corresponding to the tree. The tree is printed using **TeXDoclet.os**.

## 1.3 Class HelpOutput

### 1.3.1 Declaration

```
public class HelpOutput
extends java.lang.Object
```

### 1.3.2 Constructor summary

[HelpOutput\(\)](#)

### 1.3.3 Method summary

[printHelp\(\)](#)

### 1.3.4 Constructors

- **HelpOutput**  
public HelpOutput()

### 1.3.5 Methods

- **printHelp**  
protected static void printHelp()

## 1.4 Class HTMLtoLaTeXBackEnd

This class implements a `ParserCallback` that translates HTML to the corresponding  $\text{\LaTeX}$ . Not all tags are processed but the most common are.

HTML links to files located in the doc-files directory (appendix\_a.html, appendix\_b.txt) are transformed to references to the appendix, whereby the referenced files themselves are included in the appendix.

### 1.4.1 See also

- [javax.swing.text.html.parser.ParserDelegator](#)

### 1.4.2 Declaration

```
public class HTMLtoLaTeXBackEnd
extends javax.swing.text.html.HTMLEditorKit.ParserCallback
```

### 1.4.3 Constructor summary

[HTMLtoLaTeXBackEnd\(StringBuffer\)](#) Constructs a new instance.

### 1.4.4 Method summary

[fixText\(String\)](#) Converts a HTML string into  $\text{\LaTeX}$  using an instance of `HTMLtoLaTeXBackEnd`.

[handleEndTag\(HTML.Tag, int\)](#) This method handles HTML tags that mark an ending (e.g.

[handleSimpleTag\(HTML.Tag, MutableAttributeSet, int\)](#) This method handles simple HTML tags (e.g.

**handleStartTag(HTML.Tag, MutableAttributeSet, int)** This method handles HTML tags that mark a beginning (e.g. **handleText(char[], int)** This method handles all other text.

#### 1.4.5 Constructors

- **HTMLtoLaTeXBackEnd**  
`public HTMLtoLaTeXBackEnd(java.lang.StringBuffer ret)`
  - **Description**  
Constructs a new instance.
  - **Parameters**  
    - \* `StringBuffer` – The `StringBuffer` where the translated HTML is appended.

#### 1.4.6 Methods

- **fixText**  
`public static java.lang.String fixText(java.lang.String str)`
  - **Description**  
Converts a HTML string into  $\text{\LaTeX}$  using an instance of `HTMLtoLaTeXBackEnd`.
- **handleEndTag**  
`public void handleEndTag(javax.swing.text.html.HTML.Tag tag, int pos)`
  - **Description**  
This method handles HTML tags that mark an ending (e.g. `</P>`-tags). It is called by the parser whenever such a tag is encountered.
- **handleSimpleTag**  
`public void handleSimpleTag(javax.swing.text.html.HTML.Tag tag, javax.swing.text.MutableAttributeSet attrSet, int pos)`
  - **Description**  
This method handles simple HTML tags (e.g. `<HR>`-tags). It is called by the parser whenever such a tag is encountered.
- **handleStartTag**  
`public void handleStartTag(javax.swing.text.html.HTML.Tag tag, javax.swing.text.MutableAttributeSet attrSet, int pos)`
  - **Description**  
This method handles HTML tags that mark a beginning (e.g. `<P>`-tags). It is called by the parser whenever such a tag is encountered.
- **handleText**  
`public void handleText(char[] data, int pos)`
  - **Description**  
This method handles all other text.

### 1.4.7 Members inherited from class `HTMLEditorKit.ParserCallback`

`javax.swing.text.html.HTMLEditorKit.ParserCallback`  
`flush`, `handleComment`, `handleEndOfLineString`, `handleEndTag`, `handleError`, `handleSimpleTag`,  
`handleStartTag`, `handleText`, `IMPLIED`

## 1.5 Class `InterfaceHierarchy`

Manages and prints a interface hierarchy. Use `add` to add another interface to the hierarchy. Use `printTree` to print the corresponding  $\text{\LaTeX}$ .

### 1.5.1 Declaration

```
public class InterfaceHierarchy
extends java.lang.Object
```

### 1.5.2 Field summary

[`root`](#)

### 1.5.3 Constructor summary

[`InterfaceHierarchy\(\)`](#) Creates new `InterfaceHierarchy`

### 1.5.4 Method summary

[`add\(ClassDoc\)`](#) Adds another interface to the hierarchy  
[`printBranch\(RootDoc, SortedMap, double, double\)`](#) Prints a branch of the  
tree.  
[`printTree\(RootDoc, double\)`](#) Prints the  $\text{\LaTeX}$  corresponding to the tree.

### 1.5.5 Fields

- `public java.util.SortedMap` **`root`**

### 1.5.6 Constructors

- **`InterfaceHierarchy`**  
`public` **`InterfaceHierarchy()`**  
  - **Description**  
Creates new `InterfaceHierarchy`

### 1.5.7 Methods

- **`add`**  
`protected` `java.util.SortedMap` **`add(com.sun.javadoc.ClassDoc cls)`**

- Description

Adds another interface to the hierarchy

- printBranch

```
protected void printBranch(com.sun.javadoc.RootDoc rootDoc,  
java.util.SortedMap map, double indent, double overviewindent)
```

- Description

Prints a branch of the tree. The branch is printed using `TeXDoclet.os`.

- printTree

```
public void printTree(com.sun.javadoc.RootDoc rootDoc, double
overviewindent)
```

- Description

Prints the L<sup>A</sup>T<sub>E</sub>X corresponding to the tree. The tree is printed using TeXDoclet.os.

## 1.6 Class MarkdownTest

This class is just for testing the Markdown processing output.

a) Some text

### Markdown code :

some text some text some text some text some text some text some  
text some text some text some text some text some text some text  
some text some text some text with 2 ending spaces

```
text some text some text some text some text some text some text
some text some text some text some text some text some text some text
```

```
text some text some text some text some text some text some text
some text some text
```

results in :

some text some text some text some text some text some text some text some text some  
text some text some text some text some text some text some text some text with 2 ending spaces

text some text some text some text some text some text some text some text some text  
some text some text some text some text some text

text some text some text some text some text some text some text some text some text

## b) Lists

Markdown code :

```
unsorted :
```



```
- item1
  - item11
  - item12
- item2
```

or :

```
+ item1
  + item12
    + item13
```

sorted :

```
1. item1
  1. item11
  2. item12
2. item2
  - item21
  - item22
3. item3
```

lists with paragraphs :

```
1. some text some text some text some text some text some text some text some
text some text some text
```

```
    some text some text some text some text some text some text
```

```
2. some text some text some text some text some text some text
```

**results in :**

unsorted :

```
• item1
  - item11
  - item12
• item2
```

or :

```
• item1
  - item12
    * item13
```

sorted :

```
1. item1
```

- (a) item11
- (b) item12

## 2. item2

- item21
- item22

## 3. item3

lists with paragraphs :

1. some text some text some text some text some text some text some text some text some text some text  
some text some text some text some text some text some text
2. some text some text some text some text some text some text

### c) Blockquotes

**Markdown code :**

```
some text some text some text some text some text some text
```

```
> some quoting text
>
> > some nested quoting text
>
> some quoting text
>
> ##### header in blockquote
>
> a list in blockquote :
>
> 1. item1
> 2. item2
>   1. item21
>   2. item22
> 3. item3
>
> some quoting text
>
>   code in blockquote
```

**results in :**

```
some text some text some text some text some text some text
some quoting text
some nested quoting text
some quoting text
```

**header in blockquote**

a list in blockquote :

1. item1
2. item2
  - (a) item21
  - (b) item22
3. item3

some quoting text  
code in blockquote

**d) Preformatted text**

**Markdown code :**

some preformatted :

```
code line 1
code line 2
```

**results in :**

```
some preformatted :
code line 1
code line 2
```

**e) Horizontal rules**

**Markdown code :**

```
***
```

**results in :**

---

**f) Emphasis**

**Markdown code :**

```
*single asterisks* (em)
```

```
_single underscores_ (em)
```

```
**double asterisks** (strong)
```

```
__double underscores__ (strong)
```

results in :

*single asterisks* (em)  
*single underscores* (em)  
**double asterisks** (strong)  
**double underscores** (strong)

## h) Code

Markdown code :

```
some code : 'TeXDoclet extends Doclet' and ''There is a literal backtick (')
here.'''
```

results in :

```
some code : TeXDoclet extends Doclet and There is a literal backtick (') here.
```

### 1.6.1 Declaration

```
public class MarkdownTest
extends java.lang.Object
```

### 1.6.2 Constructor summary

[MarkdownTest\(\)](#)

### 1.6.3 Constructors

- **MarkdownTest**  
 public **MarkdownTest()**

## 1.7 Class Package

This class is used to manage the contents of a Java package. It accepts ClassDoc objects and examines them and groups them according to whether they are classes, interfaces, exceptions or errors. The accumulated Vectors can then be processed to get to all of the elements of the package that fall into each category. If needed the classes, interfaces, exceptions and errors can be sorted using the `sort` method.

### 1.7.1 See also

- [Package.sort\(\)](#) (in 1.7.8, page 15)

### 1.7.2 Declaration

```
public class Package
extends java.lang.Object
```

### 1.7.3 Field summary

**classes** The classes this package has in it  
**errors** The errors this package has in it  
**exceptions** The exceptions this package has in it  
**interfaces** The interfaces this package has in it  
**pkg** The name of the package this object is for  
**pkgDoc**

### 1.7.4 Constructor summary

**Package(String, PackageDoc)** Construct a new object corresponding to the passed package name.

### 1.7.5 Method summary

**addElement(ClassDoc)** Adds a ClassDoc element to this package.  
**sort()** Sorts the vectors of classes, interfaces exceptions and errors.

### 1.7.6 Fields

- protected com.sun.javadoc.PackageDoc **pkgDoc**
- protected java.lang.String **pkg**
  - The name of the package this object is for
- protected java.util.Vector **classes**
  - The classes this package has in it
- protected java.util.Vector **interfaces**
  - The interfaces this package has in it
- protected java.util.Vector **exceptions**
  - The exceptions this package has in it
- protected java.util.Vector **errors**
  - The errors this package has in it

### 1.7.7 Constructors

- **Package**  
 public **Package**(java.lang.String **pkg**, com.sun.javadoc.PackageDoc **doc**)
  - **Description**  
 Construct a new object corresponding to the passed package name.
  - **Parameters**
    - \* **pkg** – the package name to use

### 1.7.8 Methods

- **addElement**

```
public void addElement(com.sun.javadoc.ClassDoc cd)
```

- **Description**

Adds a ClassDoc element to this package.

- **Parameters**

\* *cd* – the object to add to this package

- **sort**

```
public void sort()
```

- **Description**

Sorts the vectors of classes, interfaces exceptions and errors.

## 1.8 Class **TableInfo**

This class provides support for converting HTML tables into  $\text{\LaTeX}$  tables. Some of the things **NOT** implemented include the following:

- valign attributes are not processed, but align= is.
- rowspan attributes are not processed, but colspan= is.
- the argument to border= in the table tag is not used to control line size

Here is an example table.

Column 1 Heading	Column two heading	Column three heading																					
data	Span two columns																						
<i>more data</i>	right	left																					
<div><div>A nested table exam- ple</div><table><tr><th>Column one Head- ing</th><th>Column two head- ing</th><th>Column three head- ing</th></tr><tr><td>data</td><td colspan="2">Span two columns</td></tr><tr><td><i>more data</i></td><td>right</td><td>left</td></tr><tr><td>1</td><td>first line</td><td></td></tr><tr><td>2</td><td>second line</td><td></td></tr><tr><td>3</td><td>third line</td><td></td></tr><tr><td>4</td><td>fourth line</td><td></td></tr></table></div>			Column one Head- ing	Column two head- ing	Column three head- ing	data	Span two columns		<i>more data</i>	right	left	1	first line		2	second line		3	third line		4	fourth line	
			Column one Head- ing	Column two head- ing	Column three head- ing																		
			data	Span two columns																			
			<i>more data</i>	right	left																		
			1	first line																			
			2	second line																			
			3	third line																			
			4	fourth line																			

1.8.1 Declaration

```
public class TableInfo
extends java.lang.Object
```

1.8.2 Constructor summary

```
TableInfo()
```

1.8.3 Method summary

- `endCol()` Ends the current column.
- `endRow()` Ends the current row.
- `endTable()` Ends the table, closing the last row as needed
- `startCol(MutableAttributeSet)` Starts a new column, possibly closing the current column if needed
- `startHeadCol(MutableAttributeSet)` Starts a new Heading column, possibly closing the current column if needed.
- `startRow(MutableAttributeSet)` Starts a new row, possibly closing the current row if needed

**startTable(StringBuffer, MutableAttributeSet)** Constructs a new table object and starts processing of the table by scanning the `<table>` passed to count columns.

#### 1.8.4 Constructors

- **TableInfo**  
`public TableInfo()`

#### 1.8.5 Methods

- **endCol**  
`public void endCol()`
  - **Description**  
Ends the current column.
  - **Parameters**  
    - \* **ret** – The output buffer to put  $\text{\LaTeX} 2_{\epsilon}$  into.
- **endRow**  
`public void endRow()`
  - **Description**  
Ends the current row.
  - **Parameters**  
    - \* **ret** – The output buffer to put  $\text{\LaTeX} 2_{\epsilon}$  into.
- **endTable**  
`public java.lang.StringBuffer endTable()`
  - **Description**  
Ends the table, closing the last row as needed
  - **Parameters**  
    - \* **ret** – The output buffer to put  $\text{\LaTeX} 2_{\epsilon}$  into.
- **startCol**  
`public void startCol(javax.swing.text.MutableAttributeSet attrSet)`
  - **Description**  
Starts a new column, possibly closing the current column if needed
  - **Parameters**  
    - \* **ret** – The output buffer to put  $\text{\LaTeX} 2_{\epsilon}$  into.
    - \* **p** – the properties from the `<td>` tag
- **startHeadCol**  
`public void startHeadCol(javax.swing.text.MutableAttributeSet attrSet)`



- **Description**

Starts a new Heading column, possibly closing the current column if needed. A Heading column has a Bold Face font directive around it.

- **Parameters**

- \* **ret** – The output buffer to put  $\text{\LaTeX} 2_{\epsilon}$  into.
- \* **p** – The properties from the `<th>` tag

- **startRow**

```
public void startRow(javax.swing.text.MutableAttributeSet attrSet)
```

- **Description**

Starts a new row, possibly closing the current row if needed

- **Parameters**

- \* **ret** – The output buffer to put  $\text{\LaTeX}$  into.
- \* **p** – The properties from the `<tr>` tag

- **startTable**

```
public java.lang.StringBuffer startTable(java.lang.StringBuffer org,
    javax.swing.text.MutableAttributeSet attrSet)
```

- **Description**

Constructs a new table object and starts processing of the table by scanning the `<table>` passed to count columns.

- **Parameters**

- \* **p** – properties found on the `<table>` tag
- \* **ret** – the result buffer that will contain the output
- \* **table** – the input string that has the entire table definition in it.
- \* **off** – the offset into `<table>` where scanning should start

## 1.9 Class TestFilter

This class filters out classes beginning with "Test" when applied to the Doclet.

### 1.9.1 Declaration

```
public class TestFilter
    extends java.lang.Object
    implements ClassFilter
```

### 1.9.2 Constructor summary

[TestFilter\(\)](#)

### 1.9.3 Method summary

[includeClass\(ClassDoc\)](#) Returns false if class name starts with "Test".

### 1.9.4 Constructors

- **TestFilter**  
`public TestFilter()`

### 1.9.5 Methods

- **includeClass**  
`public boolean includeClass(com.sun.javadoc.ClassDoc cd)`
  - **Description**  
Returns false if class name starts with "Test".

## 1.10 Class *TeXDoclet*

This class provides a Java *javadoc* Doclet which generates a  $\text{\LaTeX} 2_{\epsilon}$  document out of the java classes that it is used on. This is convenient for creating printable documentation complete with cross reference information.

### Supported HTML tags

`<a>` including an additional attribut "doprinturl". Since the output of the doclet should be printable, the href attribut of tags is printed in parentheses following the link if attribut "doprinturl" is set. Sometimes this is undesirable, and omitting "doprinturl" attribut will prevent this.

`<dl>` with the associated `<dt><dd></dl>` tags

`<p>` but not `align=center...yet`

`<br>` but not `clear=xxx`

`<table>` including all the associated `<td><th><tr></td></th></tr>`

`<ol>` ordered lists

`<ul>` unordered lists

`<font>` font coloring

`<pre>` preformatted text

`<code>` fixed point fonts

`<i>` italicized fonts

`<b>` bold fonts

`<sub>` subscript

`<sup>` superscript

<center> center

<img> image located in java sources ()

1. example converted from JPG: (image file not found)
2. example converted from GIF: (image file not found)

<img> image located in the www: (see image at <http://upload.wikimedia.org/wikipedia/commons/9/92/LaTeX...>)

### Extra tags

#### <TEX>

A new tag is defined: <TEX>. This tag is useful for passing T<sub>E</sub>X code directly to the T<sub>E</sub>X compiler. The following code:

```
<TEX txt="\[ F\left( x \right) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz \]">
<BR><BR><B>This alternative text will appear if the javadoc/HTML is parsed
by any other doclet/browser</B><BR><BR></TEX>
```

will produce the following result:

$$F(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz$$

The "alternative" text is ignored by the TeXDoclet, but useful if you want to use both the TeXDoclet and a regular HTML based doclet.

#### <PRE format="markdown">

Instead of writing your java documentation in often hard to read HTML code you can make use of **Markdown** syntax. The HTML <PRE> tag is used therefore to prevent your IDE from automatically reordering your Markdown documentation text. Markdown parsing is based on the **Pegdown** implementation. The following code :

```
<PRE format="markdown">

some text some text some text some text some text some text some text

#### Lists

- item1
    1. item11
    2. item12
- item1

#### Text formatting
```

`_emphasis_` and `--strong--` and some 'code' :

```
code line 1
code line 2
```

some text some text some text some text some text some text some text

<PRE>

will produce the following :

some text some text some text some text some text some text some text

## Lists

- item1
  1. item11
  2. item12
- item1

## Text formatting

*emphasis* and **strong** and some code :

```
code line 1
code line 2
```

some text some text some text some text some text some text some text

### 1.10.1 See also

- [HTMLtoLaTeXBackEnd](#) (in 1.4, page 6)
- [TeXDoclet.start\(RootDoc\)](#) (in 1.10.8, page 23)

### 1.10.2 Declaration

```
public class TeXDoclet
extends com.sun.javadoc.Doclet
```

### 1.10.3 Field summary

```
BOLD
CHAPTER_LEVEL
ITALIC
os Writer for writing to output file
SECTION_LEVEL
SUBSECTION_LEVEL
TRUETYPE
```

#### 1.10.4 Constructor summary

**TeXDoclet()**

#### 1.10.5 Method summary

**finish()**

**init()**

**initSections()**

**main(String[])**

**optionLength(String)** Returns how many arguments would be consumed if **option** is a recognized option.

**start(RootDoc)** Called by the framework to format the entire document

**validOptions(String[], DocErrorReporter)** Checks the passed options and their arguments for validity.

#### 1.10.6 Fields

- public static final java.lang.String **SECTION\_LEVEL**
- public static final java.lang.String **CHAPTER\_LEVEL**
- public static final java.lang.String **SUBSECTION\_LEVEL**
- public static final java.lang.String **BOLD**
- public static final java.lang.String **TRUETYPE**
- public static final java.lang.String **ITALIC**
- public static java.io.PrintWriter **os**
  - Writer for writing to output file

#### 1.10.7 Constructors

- **TeXDoclet**  
public **TeXDoclet()**

#### 1.10.8 Methods

- **finish**  
public static void **finish()**
- **init**  
public static void **init()**
- **initSections**  
public static void **initSections()**
- **main**  
public static void **main**(java.lang.String[] **args**)

- **optionLength**

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

- **Description**

Returns how many arguments would be consumed if `option` is a recognized option.

- **Parameters**

- \* `option` – the option to check

- **start**

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

- **Description**

Called by the framework to format the entire document

- **Parameters**

- \* `root` – the root of the starting document

- **validOptions**

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

- **Description**

Checks the passed options and their arguments for validity.

- **Parameters**

- \* `args` – the arguments to check

- \* `err` – the interface to use for reporting errors

### 1.10.9 Members inherited from class `Doclet`

`com.sun.javadoc.Doclet`

`languageVersion`, `optionLength`, `start`, `validOptions`