



UNIVERSITAS INDONESIA

**OPEN DOMAIN INFORMATION EXTRACTION OTOMATIS DARI TEKS
BAHASA INDONESIA**

TESIS

**YOHANES GULTOM
1506706345**

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI MAGISTER ILMU KOMPUTER
DEPOK
JUNI 2017**



UNIVERSITAS INDONESIA

**OPEN DOMAIN INFORMATION EXTRACTION OTOMATIS DARI TEKS
BAHASA INDONESIA**

TESIS

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Magister Ilmu Komputer**

YOHANES GULTOM

1506706345

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI MAGISTER ILMU KOMPUTER
DEPOK
JUNI 2017**

ABSTRAK

Nama : Yohanes Gultom
Program Studi : Magister Ilmu Komputer
Judul : Open Domain Information Extraction Otomatis dari Teks Bahasa Indonesia

Banyaknya jumlah dokumen digital yang tersedia saat ini sudah melebihi kapasitas manusia untuk memprosesnya secara manual. Hal ini mendorong munculnya kebutuhan akan metode ekstraksi informasi (*information extraction*) otomatis dari teks atau dokumen digital dari berbagai domain (*open domain*). Sayangnya, sistem *open domain information extraction* (*open IE*) yang ada saat ini hanya berlaku untuk bahasa tertentu saja. Selain itu belum ada sistem *open IE* untuk bahasa Indonesia yang dipublikasikan. Pada penelitian ini Penulis memperkenalkan sebuah sistem untuk mengekstraksi relasi antar entitas dari teks bahasa Indonesia dari berbagai domain. Sistem ini menggunakan sebuah NLP *pipeline*, pembangkit kandidat *triple* (*triple candidates generator*) dan pengembang token (*token expander*) berbasis aturan serta pemilih *triple* berbasis *machine learning*. Setelah melakukan *cross-validation* terhadap empat kandidat model: *logistic regression*, SVM, MLP dan *Random Forest*, Penulis menemukan bahwa *Random Forest* adalah *classifier* yang terbaik untuk dijadikan *triple selector* dengan skor F1 0.58 (*precision* 0.62 dan *recall* 0.58). Penyebab utama skor yang masih rendah ini adalah aturan pembangkitan kandidat yang masih sederhana dan cakupan pola *dataset* yang masih rendah.

Kata Kunci:

information extraction, open domain, natural language processing, bahasa Indonesia

ABSTRACT

Name : Yohanes Gultom
Program : Magister Ilmu Komputer
Title : Automatic Open Domain Information Extraction from Indonesian Text

The vast amount of digital documents, that have surpassed human processing capability, calls for an automatic information extraction method from any text document regardless of their domain. Unfortunately, open domain information extraction (open IE) systems are language-specific and there is no published system for Indonesian language. This paper introduces a system to extract entity relations from Indonesian text in triple format using an NLP pipeline, rule-based candidates generator, token expander and machine-learning-based triple selector. We cross-validate four candidates: logistic regression, SVM, MLP, Random Forest using our dataset to discover that Random Forest is the best classifier for the triple selector achieving 0.58 F1 score (0.62 precision and 0.58 recall). The low score is largely due to the simplistic candidate generation rules and the coverage of dataset.

Keywords:

information extraction, open domain, natural language processing, Indonesian language

DAFTAR ISI

HALAMAN JUDUL	i
ABSTRAK	ii
Daftar Isi	iv
Daftar Gambar	vi
Daftar Tabel	vii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	3
1.2.1 Definisi Permasalahan	3
1.2.2 Batasan Permasalahan	3
1.3 Tujuan dan Manfaat	3
1.4 Sistematika Penulisan	4
2 TINJAUAN PUSTAKA	5
2.1 L ^A T _E X Secara Singkat	5
2.2 L ^A T _E X Kompiler dan IDE	6
2.3 Bold, Italic, dan Underline	6
2.4 Memasukan Gambar	7
2.5 Membuat Tabel	7
3 METODE PENELITIAN	9
3.1 Satu Persamaan	9
3.2 Lebih dari Satu Persamaan	9
4 HASIL DAN ANALISIS	11
4.1 thesis.tex	11
4.2 laporan_setting.tex	11
4.3 istilah.tex	11
4.4 hype.indonesia.tex	11
4.5 pustaka.tex	12

4.6	bab[1 - 6].tex	12
4.7	Penulisan <i>code</i> atau <i>pseudocode</i> program	12
4.7.1	<i>Inline</i>	12
4.7.2	<i>Multiline</i>	12
5	PENUTUP	14
5.1	Mengubah Tampilan Teks	14
5.2	Memberikan Catatan	14
5.3	Menambah Isi Daftar Isi	15
5.4	Memasukan PDF	15
5.5	Membuat Perintah Baru	19
	Daftar Referensi	20
	LAMPIRAN	1
	Lampiran 1	2

DAFTAR GAMBAR

1.1	Contoh input dan output yang diharapkan dari sistem open IE untuk bahasa Indonesia	2
2.1	<i>Creative Common License 1.0 Generic.</i>	7

DAFTAR TABEL

1.1	Perbandingan antara <i>information extraction</i> tradisional (IE), <i>open domain extraction</i> (open IE) dan <i>knowledge extraction</i> (KE)	1
2.1	Contoh Tabel	7
2.2	An Example of Rows Spanning Multiple Columns	8
2.3	An Example of Columns Spanning Multiple Rows	8
2.4	An Example of Spanning in Both Directions Simultaneously	8

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Di masa sekarang ketersediaan dokumen digital berbahasa natural seperti berita, jurnal dan buku elektronik (*e-book*) sudah sangat banyak dan terus meningkat dengan cepat karena didorong oleh meningkatnya pemanfaatan komputer, *smartphone* dan *internet*. Jumlah dokumen digital tersebut telah melampaui batas kemampuan manusia untuk memproses secara manual sehingga menimbulkan kebutuhan akan proses otomatis untuk melakukannya (Banko et al., 2007). Salah satu proses yang dikembangkan adalah *information extraction* (IE) yang secara selektif menyusun dan mengkombinasikan data yang ditemukan di dalam teks atau dokumen menjadi informasi (Cowie and Lehnert, 1996).

Meskipun IE sudah mampu manusia untuk memproses dokumen digital dengan lebih efisien, metode yang digunakan umumnya hanya berlaku untuk kelompok dokumen yang homogen atau berada dalam satu domain (*closed-domain*). Hal ini terjadi karena umumnya teknik yang dipakai dibuat sedemikian rupa untuk memanfaatkan pola tertentu pada teks atau dokumen (Cowie and Lehnert, 1996). Sebagai contoh untuk mengekstraksi nama penulis dari berita elektronik, salah satu cara paling mudah adalah mencari nama orang di awal atau akhir dokumen. Cara yang sama tidak bisa digunakan untuk mencari nama penulis dari dokumen lain seperti jurnal karena struktur dokumen yang berbeda. Hal ini mendorong berkembangnya metode lain yang mampu mengekstraksi informasi dari berbagai domain (*open domain*) yang disebut *open domain information extraction* (*open IE*) (Banko et al., 2007).

Tabel 1.1: Perbandingan antara *information extraction* tradisional (IE), *open domain extraction* (open IE) dan *knowledge extraction* (KE)

Aspek	IE	Open IE	KE
Domain	Tertutup	Terbuka	Terbuka
Format	Tergantung domain	Triples	RDF Triples
Ontologi	Tidak tersedia	Opsional	Wajib

Metode *open domain information extraction* (*open IE*) mengekstrak informasi dari dokumen dalam format *tuple* (x, r, y) di mana r adalah relasi antara dua buah

entitas x dan y (Etzioni et al., 2011). Format informasi dari tiga nilai tersebut, yang disebut juga *triple*, berlaku umum untuk semua dokumen yang berisi bahasa natural sehingga dapat diterapkan pada dokumen dari berbagai domain. Format *triple* yang digunakan open IE memiliki kemiripan dengan format yang lazim digunakan pada *knowledge extraction* (KE), yaitu Resource Data Format (RDF) specification¹ (Auer et al., 2007; Exner and Nugues, 2014). Namun, open IE umumnya tidak mengikuti seluruh spesifikasi RDF dan tidak memiliki set ontologi tetap. Ringkasan perbandingan antara open IE dengan IE tradisional dan KE ditunjukkan pada Tabel 1.1.

Seiring dengan berkembangnya waktu, beberapa sistem open IE sudah dikembangkan (Schmitz et al., 2012) untuk bahasa Inggris. Bahkan penelitian terkait melaporkan kesuksesan aplikasi open IE untuk *task question answering* (Fader et al., 2011) dan *information retrieval* (Etzioni, 2011). Akan tetapi karena sistem open IE menggunakan satu atau lebih *task natural language processing* (NLP) dan aturan/heuristik yang hanya berlaku untuk bahasa tertentu, maka sistem yang berkembang tidak dapat dipakai untuk memproses teks atau dokumen dalam bahasa lain seperti bahasa Indonesia. Oleh karena itu dalam penelitian ini, Penulis memperkenalkan sistem open IE untuk bahasa Indonesia.

<p>Input</p> <p>”Sembungan adalah sebuah desa yang terletak di kecamatan Kejajar, kabupaten Wonosobo, Jawa Tengah, Indonesia.”</p> <p>Output</p> <ol style="list-style-type: none"> 1. (Sembungan, adalah, desa) 2. (Sembungan, terletak di, kecamatan Kejajar)

Gambar 1.1: Contoh input dan output yang diharapkan dari sistem open IE untuk bahasa Indonesia

Sistem open IE yang Penulis ajukan bertujuan untuk mengekstrak sejumlah *triple* dari satu atau lebih teks bahasa Indonesia seperti contoh pada Gambar 1.1. Sistem ini terdiri dari sebuah NLP *pipeline*, pembangkit kandidat *triple* (*triple candidates generator*), pengembang token (*token expander*) dan sebuah model *machine learning* untuk memilih *triple* (*triple selector*). Untuk melatih model *triple selector* tersebut, Penulis juga membuat dataset berisi 1.611 kandidat *triple* bahasa Indonesia yang valid dan yang tidak valid. Sistem ini diharapkan dapat menjadi referensi

¹Resource Data Format W3C <https://www.w3.org/RDF/>

dalam pengembangan open IE untuk bahasa Indonesia dan juga digunakan untuk kebutuhan aplikasi yang lebih kompleks seperti pendeteksian plagiarisme, *question answering* dan *knowledge extraction*.

1.2 Permasalahan

Pada bagian ini akan dijelaskan mengenai definisi permasalahan yang ingin diselesaikan pada penelitian ini serta batasan yang ditetapkan.

1.2.1 Definisi Permasalahan

Permasalahan yang ditemukan dan ingin diselesaikan pada penelitian ini:

1. Bagaimana merancang sistem *open IE* yang cocok untuk bahasa Indonesia?
2. Bagaimana implementasi sistem *open IE* tersebut?

1.2.2 Batasan Permasalahan

Batasan permasalahan pada penelitian ini adalah:

1. Proses dibatasi pada dokumen teks bahasa Indonesia yang setiap barisnya hanya berisi satu kalimat. Praproses yang dibutuhkan untuk mengubah dokumen dari format yang berbeda tidak dibahas di penelitian ini.
2. Penelitian ini hanya berfokus untuk menghasilkan *triple* yang eksplisit secara sintaktik. Contoh *triple* yang eksplisit dari kalimat "Universitas Indonesia berada di Depok, Jawa Barat, Indonesia" adalah (*Universitas Indonesia, terletak di, Depok*). Sedangkan *triple* yang implisit seperti (*Depok, terletak di, Jawa Barat*) belum ditangani pada penelitian ini.
3. Penelitian ini tidak berfokus untuk mencapai kinerja sistem yang sebanding dengan sistem open IE untuk bahasa Inggris pada penelitian terkait.

1.3 Tujuan dan Manfaat

Tujuan dan manfaat dari penelitian ini adalah:

Tujuan

1. Merancang sistem open IE untuk teks bahasa Indonesia.

2. Mengimplementasikan sistem open IE untuk teks bahasa Indonesia.

Manfaat

1. Menghasilkan sistem *open IE* yang dapat digunakan untuk mengekstrak entitas relasi dari teks bahasa Indonesia
2. Memberikan acuan untuk pengembangan sistem *open IE* untuk bahasa Indonesia
3. Memberikan kontribusi terhadap perkembangan sumber daya bahasa (*language resources*) Indonesia

1.4 Sistematika Penulisan

Sistematika penulisan laporan adalah sebagai berikut:

- **Bab 1 PENDAHULUAN**
Bab ini akan menjelaskan mengenai latar belakang permasalahan, rumusan masalah, tujuan, manfaat dan batasan penelitian.
- **Bab 2 TINJAUAN PUSTAKA**
Bab ini akan menjelaskan landasan teori yang digunakan pada penelitian ini serta memaparkan kajian pustaka terhadap penelitian-penelitian terkait.
- **Bab 3 METODE PENELITIAN**
Bab ini akan menjelaskan mengenai tahapan, rancangan & implementasi sistem, pengumpulan & pengolahan data dan teknik evaluasi yang digunakan pada penelitian ini.
- **Bab 4 HASIL DAN ANALISIS**
Bab ini akan menjelaskan tentang hasil eksperimen dan analisis hasil eksperimen.
- **Bab 5 PENUTUP**
Bab ini akan menjelaskan tentang kesimpulan dari penelitian yang telah dilakukan dan saran untuk penelitian berikutnya.

BAB 2

TINJAUAN PUSTAKA

@todo

tambahkan kata-kata pengantar bab 2 disini

2.1 L^AT_EX Secara Singkat

Definisi dari LaTeX (?) adalah:

LaTeX is a family of programs designed to produce publication-quality typeset documents. It is particularly strong when working with mathematical symbols.

The history of LaTeX begins with a program called TEX. In 1978, a computer scientist by the name of Donald Knuth grew frustrated with the mistakes that his publishers made in typesetting his work. He decided to create a typesetting program that everyone could easily use to typeset documents, particularly those that include formulae, and made it freely available. The result is TEX. Knuth's product is an immensely powerful program, but one that does focus very much on small details. A mathematician and computer scientist by the name of Leslie Lamport wrote a variant of TEX called LaTeX that focuses on document structure rather than such details.

Contoh sitasi lainnya menggunakan \citep adalah saat kita mau mensitasi pekerjaan tentang *machine learning* (?) dan *dynamic programming* (?).

Dokumen L^AT_EX sangat mudah, seperti halnya membuat dokumen teks biasa. Ada beberapa perintah yang diawali dengan tanda '\'. Seperti perintah \\\ yang digunakan untuk memberi baris baru. Perintah tersebut juga sama dengan perintah \newline. Pada bagian ini akan sedikit dijelaskan cara manipulasi teks dan perintah-perintah L^AT_EX yang mungkin akan sering digunakan. Jika ingin belajar hal-hal dasar mengenai L^AT_EX, silahkan kunjungi:

- <http://frodo.elon.edu/tutorial/tutorial/>, atau

- <http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/>

2.2 L^AT_EX Kompiler dan IDE

Agar dapat menggunakan L^AT_EX (pada konteks hanya sebagai pengguna), Anda tidak perlu banyak tahu mengenai hal-hal didalamnya. Seperti halnya pembuatan dokumen secara visual (contohnya Open Office (OO) Writer), Anda dapat menggunakan L^AT_EX dengan cara yang sama. Orang-orang yang menggunakan L^AT_EX relatif lebih teliti dan terstruktur mengenai cara penulisan yang dia gunakan, L^AT_EX memaksa Anda untuk seperti itu.

Kembali pada bahasan utama, untuk mencoba L^AT_EX Anda cukup mendownload kompiler dan IDE. Saya menyarankan menggunakan Texlive dan Texmaker. Texlive dapat didownload dari <http://www.tug.org/texlive/>. Sedangkan Texmaker dapat didownload dari <http://www.xmlmath.net/texmaker/>. Untuk pertama kali, coba buka berkas thesis.tex dalam template yang Anda miliki pada Texmaker. Dokumen ini adalah dokumen utama. Tekan F6 (PDFLaTeX) dan Texmaker akan mengkompilasi berkas tersebut menjadi berkas PDF. Jika tidak bisa, pastikan Anda sudah menginstall Texlive. Buka berkas tersebut dengan menekan F7. Hasilnya adalah sebuah dokumen yang sama seperti dokumen yang Anda baca saat ini.

2.3 Bold, Italic, dan Underline

Hal pertama yang mungkin ditanyakan adalah bagaimana membuat huruf tercetak tebal, miring, atau memiliki garis bawah. Pada Texmaker, Anda bisa melakukan hal ini seperti halnya saat mengubah dokumen dengan OO Writer. Namun jika tetap masih tertarik dengan cara lain, ini dia:

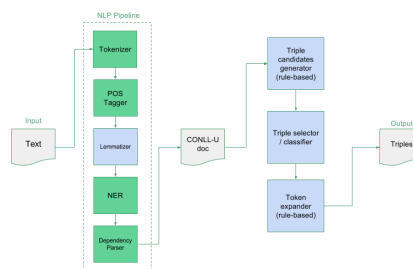
- **Bold**
Gunakan perintah `\textbf{}` atau `\bo{}`.
- *Italic*
Gunakan perintah `\textit{}` atau `\f{}`.
- Underline
Gunakan perintah `\underline{}`.
- Overline
Gunakan perintah `\overline{}`.

- *superscript*
Gunakan perintah `\{ }`.
- *subscript*
Gunakan perintah `_ { }`.

Perintah `\f` dan `\bo` hanya dapat digunakan jika package `uithesis` digunakan.

2.4 Memasukan Gambar

Setiap gambar dapat diberikan caption dan diberikan label. Label dapat digunakan untuk menunjuk gambar tertentu. Jika posisi gambar berubah, maka nomor gambar juga akan diubah secara otomatis. Begitu juga dengan seluruh referensi yang menunjuk pada gambar tersebut. Contoh sederhana adalah Gambar 2.1. Silahkan lihat code `LATEX` dengan nama `bab2.tex` untuk melihat kode lengkapnya. Harap diingat bahwa caption untuk gambar selalu terletak dibawah gambar.



Gambar 2.1: *Creative Common License 1.0 Generic.*

2.5 Membuat Tabel

Seperti pada gambar, tabel juga dapat diberi label dan caption. Caption pada tabel terletak pada bagian atas tabel. Contoh tabel sederhana dapat dilihat pada Tabel 2.1.

Tabel 2.1: Contoh Tabel

	kol 1	kol 2
baris 1	1	2
baris 2	3	4
baris 3	5	6
jumlah	9	12

Ada jenis tabel lain yang dapat dibuat dengan \LaTeX berikut beberapa diantaranya. Contoh-contoh ini bersumber dari <http://en.wikibooks.org/wiki/LaTeX/Tables>

Tabel 2.2: An Example of Rows Spanning Multiple Columns

No	Name	Week 1			Week 2		
		A	B	C	A	B	C
1	Lala	1	2	3	4	5	6
2	Lili	1	2	3	4	5	6
3	Lulu	1	2	3	4	5	6

Tabel 2.3: An Example of Columns Spanning Multiple Rows

Percobaan	Iterasi	Waktu
Pertama	1	0.1 sec
Kedua	1	0.1 sec
	3	0.15 sec
Ketiga	1	0.09 sec
	2	0.16 sec
	3	0.21 sec

Tabel 2.4: An Example of Spanning in Both Directions Simultaneously

		Title			
		A	B	C	D
Type	X	1	2	3	4
	Y	0.5	1.0	1.5	2.0
Resource	I	10	20	30	40
	J	5	10	15	20

BAB 3

METODE PENELITIAN

@todo

tambahkan kata-kata pengantar bab 1 disini

3.1 Satu Persamaan

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1} \quad (3.1)$$

Persamaan 3.1 diatas adalah persamaan garis. Persamaan 3.1 dan 3.2 sama-sama dibuat dengan perintah `\align`. Perintah ini juga dapat digunakan untuk menulis lebih dari satu persamaan.

$$\underbrace{|\overline{ab}|}_{\text{pada bola } |\overline{ab}| = r} = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2 + ||(z_b - z_a)^2} \quad (3.2)$$

3.2 Lebih dari Satu Persamaan

$$|\overline{a} * \overline{b}| = |\overline{a}| |\overline{b}| \sin \theta \quad (3.3)$$

$$\begin{aligned} \overline{a} * \overline{b} &= \begin{vmatrix} \hat{i} & x_1 & x_2 \\ \hat{j} & y_1 & y_2 \\ \hat{k} & z_1 & z_2 \end{vmatrix} \\ &= \hat{i} \begin{vmatrix} y_1 & y_2 \\ z_1 & z_2 \end{vmatrix} + \hat{j} \begin{vmatrix} z_1 & z_2 \\ x_1 & x_2 \end{vmatrix} + \hat{k} \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} \end{aligned}$$

Pada Persamaan 3.3 dapat dilihat beberapa baris menjadi satu bagian dari Persamaan 3.3. Sedangkan dibawah ini dapat dilihat bahwa dengan cara yang sama, Persamaan 3.4, 3.5, dan 3.6 memiliki nomor persamaannya masing-masing.

$$\int_a^b f(x) dx + \int_b^c f(x) dx = \int_a^c f(x) dx \quad (3.4)$$

$$\lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} = 0 \quad \text{jika pangkat } f(x) < \text{pangkat } g(x) \quad (3.5)$$

$$a^{m^{a^n \log b}} = b^{\frac{m}{n}} \quad (3.6)$$

BAB 4

HASIL DAN ANALISIS

@todo

tambahkan kata-kata pengantar bab 1 disini

4.1 thesis.tex

Berkas ini berisi seluruh berkas Latex yang dibaca, jadi bisa dikatakan sebagai berkas utama. Dari berkas ini kita dapat mengatur bab apa saja yang ingin kita tampilkan dalam dokumen.

4.2 laporan_setting.tex

Berkas ini berguna untuk mempermudah pembuatan beberapa template standar. Anda diminta untuk menuliskan judul laporan, nama, npm, dan hal-hal lain yang dibutuhkan untuk pembuatan template.

4.3 istilah.tex

Berkas istilah digunakan untuk mencatat istilah-istilah yang digunakan. Fungsinya hanya untuk memudahkan penulisan. Pada beberapa kasus, ada kata-kata yang harus selalu muncul dengan tercetak miring atau tercetak tebal. Dengan menjadikan kata-kata tersebut sebagai sebuah perintah \LaTeX tentu akan mempercepat dan mempermudah pengerjaan laporan.

4.4 hype.indonesia.tex

Berkas ini berisi cara pemenggalan beberapa kata dalam bahasa Indonesia. \LaTeX memiliki algoritma untuk memenggal kata-kata sendiri, namun untuk beberapa kasus algoritma ini memenggal dengan cara yang salah. Untuk memperbaiki pemenggalan yang salah inilah cara pemenggalan yang benar ditulis dalam berkas hype.indonesia.tex.

4.5 pustaka.tex

Berkas pustaka.tex berisi seluruh daftar referensi yang digunakan dalam laporan. Anda bisa membuat model daftar referensi lain dengan menggunakan bibtex. Untuk mempelajari bibtex lebih lanjut, silahkan buka <http://www.bibtex.org/Format>. Untuk merujuk pada salah satu referensi yang ada, gunakan perintah `\cite`, e.g. `\cite{latex.intro}` yang akan akan memunculkan ?

4.6 bab[1 - 6].tex

Berkas ini berisi isi laporan yang Anda tulis. Setiap nama berkas e.g. bab1.tex merepresentasikan bab dimana tulisan tersebut akan muncul. Sebagai contoh, kode dimana tulisan ini dibuat berada dalam berkas dengan nama bab4.tex. Ada enam buah berkas yang telah disiapkan untuk mengakomodir enam bab dari laporan Anda, diluar bab kesimpulan dan saran. Jika Anda tidak membutuhkan sebanyak itu, silahkan hapus kode dalam berkas thesis.tex yang memasukan berkas L^AT_EX yang tidak dibutuhkan; contohnya perintah `\include{bab6.tex}` merupakan kode untuk memasukan berkas bab6.tex kedalam laporan.

4.7 Penulisan *code* atau *pseudocode* program

4.7.1 *Inline*

Dengan perintah `\verb: System.out.println("Hello, World");`

Dengan perintah *custom* `\code: System.out.println("Hello, World");`

4.7.2 *Multiline*

Dengan perintah *verbatim*:

```
public class HelloWorld {
    public static void main(String[] args) {
        // Prints "Hello, World" to the terminal window.
        System.out.println("Hello, World");
    }
}
```

Dengan perintah `lstlisting`:

```
1 public class HelloWorld {  
2     public static void main(String[] args) {  
3         // Prints "Hello, World" to the terminal window.  
4         System.out.println("Hello, World");  
5     }  
6 }
```

Konfigurasi tampilan bisa dilakukan di `uithesis.sty` dengan referensi dokumentasi di https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings

BAB 5

PENUTUP

@todo

Tambahkan kata-kata pengantar bab 5 disini.

5.1 Mengubah Tampilan Teks

Beberapa perintah yang dapat digunakan untuk mengubah tampilan adalah:

- `\f`
Merupakan alias untuk perintah `\textit`, contoh *contoh hasil tulisan*.
- `\bi`
Contoh hasil tulisan.
- `\bo`
Contoh hasil tulisan.
- `\m`
Contohhasiltulisan.
- `\mc`

Contohhasiltulisan

.

- `\code`
Contoh hasil tulisan.

5.2 Memberikan Catatan

Ada dua perintah untuk memberikan catatan penulisan dalam dokumen yang Anda kerjakan, yaitu:

- `\todo`

Contoh:

@todo

Contoh bentuk todo.

- `\todoCite`

Contoh:

@todo
Referensi

5.3 Menambah Isi Daftar Isi

Terkadang ada kebutuhan untuk memasukan kata-kata tertentu kedalam Daftar Isi. Perintah `\addChapter` dapat digunakan untuk judul bab dalam Daftar isi. Contohnya dapat dilihat pada berkas `thesis.tex`.

5.4 Memasukan PDF

Untuk memasukan PDF dapat menggunakan perintah `\inpdf` yang menerima satu buah argumen. Argumen ini berisi nama berkas yang akan digabungkan dalam laporan. PDF yang dimasukan dengan cara ini akan memiliki header dan footer seperti pada halaman lainnya.

Untitled

Ini adalah berkas pdf yang dimasukan dalam dokumen laporan.

Cara lain untuk memasukan PDF adalah dengan menggunakan perintah `\putpdf` dengan satu argumen yang berisi nama berkas pdf. Berbeda dengan perintah sebelumnya, PDF yang dimasukan dengan cara ini tidak akan memiliki footer atau header seperti pada halaman lainnya.

Untitled

Ini adalah berkas pdf yang dimasukan dalam dokumen laporan.

5.5 Membuat Perintah Baru

Ada dua perintah yang dapat digunakan untuk membuat perintah baru, yaitu:

- `\Var`
Digunakan untuk membuat perintah baru, namun setiap kata yang diberikan akan diproses dahulu menjadi huruf kapital. Contoh jika perintahnya adalah `\Var{adalah}` maka ketika perintah `\Var` dipanggil, yang akan muncul adalah ADALAH.
- `\var`
Digunakan untuk membuat perintah atau baru.

DAFTAR REFERENSI

- Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. *The semantic web*, pages 722–735.
- Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676.
- Cowie, J. and Lehnert, W. (1996). Information extraction. *Communications of the ACM*, 39(1):80–91.
- Etzioni, O. (2011). Search needs a shake-up. *Nature*, 476(7358):25–26.
- Etzioni, O., Fader, A., Christensen, J., Soderland, S., et al. (2011). Open information extraction: The second generation. In *Twenty-Second International Joint Conference on Artificial Intelligence*.
- Exner, P. and Nugues, P. (2014). Refractive: An open source tool to extract knowledge from syntactic and semantic relations. In *LREC*, pages 2584–2589.
- Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.
- Schmitz, M., Bart, R., Soderland, S., Etzioni, O., et al. (2012). Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534. Association for Computational Linguistics.

LAMPIRAN

LAMPIRAN 1