

## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Organisasi

Peran yang didapatkan di dalam proyek U-Tapis 2022 ini adalah sebagai salah satu pengembang algoritma *machine learning* yang disupervisi oleh Bapak Abdullah Ahmad dari divisi *IT Operation Department* Tribun News, Bapak Moeljono Widjaja, B.Sc., M.Sc., Ph.D dari UMN, Ibu Niknik Mediyawati, S.Pd., M.Hum dari UMN, dan Ibu Marlinda Vasty Overbeek, S.Kom, M.Kom dari UMN. Tugas yang harus dilakukan oleh peran ini, yaitu mengembangkan algoritma pengecekan sintaksis kalimat yang terdiri dari algoritma *preprocessing*, algoritma *Conditional Random Field* (CRF), dan algoritma *Context-Free Grammar* (CFG). Algoritma-algoritma ini kemudian diintegrasikan dengan aplikasi website yang telah dibuat oleh tim website U-Tapis 2022 agar bisa dimanfaatkan oleh para wartawan Tribun News.

Pengerjaan tugas magang ini dilakukan secara WFH (*work from home*). Perkembangan proyek dilaporkan secara berkala ke Bapak Moeljono Widjaja, B.Sc., M.Sc., Ph.D.; Ibu Niknik Mediyawati, S.Pd., M.Hum.; dan Bapak Abdullah Ahmad. Proyek U-Tapis ini secara keseluruhan (bersama *front-end website* dan model *machine learning* lainnya) akan diujicobakan bersama dengan Redaksi Tribun di kantor Tribun Jakarta, Solo, dan Makassar.

#### 3.2 Tugas yang Dilakukan

Berikut adalah uraian tugas-tugas yang harus dikerjakan selama kegiatan Magang U-Tapis ini:

- Membuat algoritma *preprocessing* artikel berita dan membagi artikel menjadi kalimat-kalimat.
- Membuat algoritma *Conditional Random Field/CRF* untuk memberikan *tag* ke tiap kata/symbol dalam kalimat-kalimat berita tersebut.
- Membuat algoritma *Context Free Grammar/CFG* untuk melakukan parsing pada tiap kalimat-kalimat berita tersebut (sintaksis benar bila parsing berhasil dan sintaksis salah bila parsing gagal).

- Membuat *API Web Service* menggunakan *Flask Python Framework* sehingga algoritma pengecekan sintaksis kalimat dapat digunakan di *front-end website*.
- Membantu tim U-Tapis lainnya sehingga proyek U-Tapis ini dapat selesai sesuai dengan waktu yang telah ditentukan.

Saat kegiatan magang dimulai, dilakukan koordinasi dengan para dosen proyek U-Tapis terkait tugas yang perlu dilakukan beserta cakupan tugas. Kemudian, dilakukan pengembangan algoritma mulai dari eksplorasi data berita, penentuan label/tag untuk *part-of-speech tagging*, membuat rule-rule *Context Free Grammar*, dan melakukan *training* model *Conditional Random Field*. Setelah itu, dilakukan *deployment* dengan *Python Flask Web API (Application Programming Interface)* ke VPS (*Virtual Private Server* milik UMN, evaluasi performa algoritma, serta pengujian bersama tim Website U-Tapis. Pada akhir kegiatan magang ini, algoritma pengecekan sintaksis kalimat (bersama dengan komponen-komponen proyek U-Tapis lainnya) akan diujicobakan di Kantor Tribun News Jakarta (8 Desember 2022), Surakarta (10 Desember 2022), dan Makassar (10 Desember 2022) sebagai bentuk pertanggungjawaban dan pelaporan progres proyek U-Tapis ini.

### 3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang U-Tapis ini telah diuraikan di dalam Tabel 3.1.

Tabel 3.1. Pelaksanaan kerja magang tiap minggu

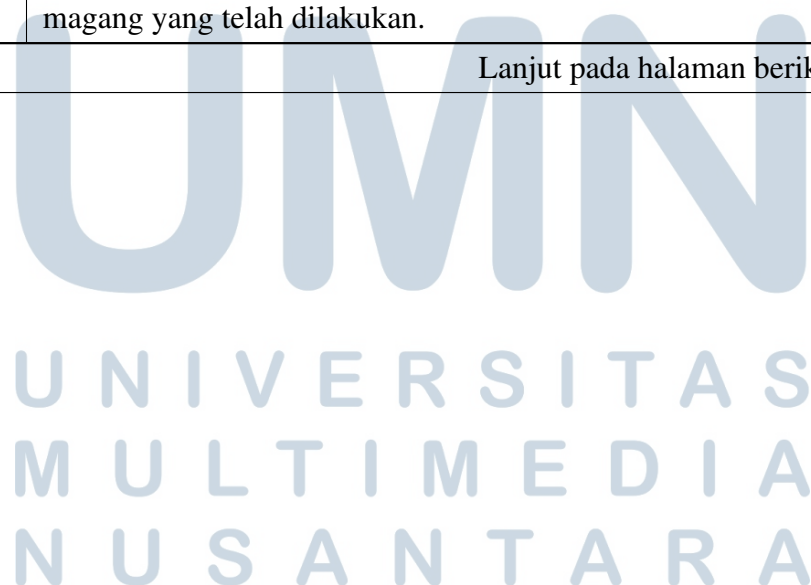
Minggu	Pekerjaan yang dilakukan
1	<ul style="list-style-type: none"> <li>- Perkenalan dengan dosen-dosen pembimbing magang dari jurusan Jurnalistik, Sistem Informatika, Informatika.</li> <li>- Mengeksplorasi data mentah berita-berita Tribun News secara manual (tanpa kode pemrograman).</li> </ul>
2	<ul style="list-style-type: none"> <li>- Mengeksplorasi data mentah berita-berita Tribun News secara manual (tanpa kode pemrograman).</li> <li>- Mengikuti pertemuan <i>online</i> terkait penjelasan penggunaan bahasa Indonesia yang baik dan benar.</li> </ul>
Lanjut pada halaman berikutnya	

Tabel 3.1 Pelaksanaan kerja magang tiap minggu (lanjutan)

Minggu	Pekerjaan yang dilakukan
3	<ul style="list-style-type: none"> <li>- Mengikuti briefing tugas-tugas yang tersedia untuk mahasiswa Informatika bersama Bapak Moeljono Widjaja, B.Sc., M.Sc., Ph.D.</li> <li>- Membuat kode Python untuk melakukan <i>parsing</i> data-data berita Tribun News dalam bentuk <i>file .eml (file email)</i>.</li> <li>- Membuat algoritma <i>preprocessing</i> di Python untuk memecah data berita menjadi per kalimat, membuang semua tanda tidak penting (<code>\n</code>, <code>\t</code>, dan lain-lain), serta menyimpan output ke dalam <i>file excel (.xlsx)</i>.</li> <li>- Mempelajari dan melakukan <i>testing</i> pada algoritma <i>part-of-speech tagging</i> milik <code>nltk (nltk.tag.CRFTagger)</code>.</li> <li>- Mempelajari dan melakukan <i>testing</i> pada algoritma <i>Context Free Grammar</i> milik <code>nltk (nltk.data dan nltk.parse.chart.ChartParser)</code>.</li> </ul>
4	<ul style="list-style-type: none"> <li>- Mempelajari kaidah bahasa Indonesia yang baik dan benar untuk menentukan <i>tag-tag</i> yang dibutuhkan algoritma CRF dan CFG.</li> <li>- Mempelajari dan melakukan <i>testing</i> pada algoritma <i>part-of-speech tagging (PoS Tagging)</i> milik <code>nltk (nltk.tag.CRFTagger)</code>.</li> <li>- Mempelajari dan melakukan <i>testing</i> pada algoritma <i>Context Free Grammar</i> milik <code>nltk (nltk.data dan nltk.parse.chart.ChartParser)</code>.</li> <li>- Memperbaiki <i>bug</i> pada algoritma <i>preprocessing</i>.</li> <li>- Berdiskusi dengan dosen pemimpin proyek U-Tapis (Ibu Niknik Mediyawati, S.Pd., M.Hum dan Bapak Moeljono Widjaja, B.Sc., M.Sc., Ph.D) terkait cakupan perbaikan sintaksis kalimat dengan algoritma <i>Context Free Grammar (CFG)</i>.</li> <li>- Menyusun <i>rule-rule</i> dari <i>Context Free Grammar (CFG)</i>.</li> </ul>
Lanjut pada halaman berikutnya	

Tabel 3.1 Pelaksanaan kerja magang tiap minggu (lanjutan)

Minggu	Pekerjaan yang dilakukan
5	<ul style="list-style-type: none"> <li>- Mengevaluasi dan memperbaiki <i>tag-tag part-of-speech tagging</i> (PoS Tagging) sesuai dengan kebutuhan <i>Context Free Grammar</i> (CFG).</li> <li>- Mempelajari frasa-frasa khusus di Bahasa Indonesia yang tidak boleh tertukar, seperti 'oleh karena itu', 'tidak...', 'tetapi...', 'bukan...', 'melainkan...', dan lain-lain. Frasa-frasa ini akan dibuat label PoS Tagging khusus dan dibuat aturan CFG khusus.</li> <li>- Menyusun <i>rule-rule</i> dari <i>Context Free Grammar</i> (CFG).</li> <li>- Memperbaiki <i>bug</i> yang ditemukan pada <i>Context Free Grammar</i> (CFG).</li> <li>- Memperbaiki algoritma <i>preprocessing</i> untuk mengabaikan isi kutipan karena konten di dalam kutipan termasuk bahasa lisan yang tidak wajib mengikuti kaidah bahasa Indonesia yang baik dan benar.</li> </ul>
6	<ul style="list-style-type: none"> <li>- Menyusun <i>rule-rule</i> dari <i>Context Free Grammar</i> (CFG).</li> <li>- Memperbaiki <i>bug</i> yang ditemukan pada <i>Context Free Grammar</i> (CFG).</li> <li>- Mengikuti Rapat Koordinasi U-Tapis dan melaporkan progres magang yang telah dilakukan.</li> </ul>
Lanjut pada halaman berikutnya	



Tabel 3.1 Pelaksanaan kerja magang tiap minggu (lanjutan)

Minggu	Pekerjaan yang dilakukan
7	<ul style="list-style-type: none"> <li>- Menambahkan dokumentasi berupa <i>comment-comment</i> rinci pada <i>file</i> CFG untuk memberikan penjelasan terkait <i>rule-rule</i> yang telah dibuat.</li> <li>- Menyederhanakan <i>tag-tag</i> untuk <i>part of speech tagging</i> (PoS Tagging).</li> <li>- Melakukan <i>manual tagging</i> pada data-data berita setelah <i>preprocessing</i> sebagai <i>training data</i> untuk algoritma <i>Conditional Random Field</i> (CRF).</li> <li>- Memperbaiki <i>Context Free Grammar</i> (CFG) agar sesuai dengan <i>part of speech tag</i> (PoS Tag) baru tersebut.</li> <li>- Mengembangkan <i>rule</i> milik CFG terkait penggunaan predikat kata sifat, predikat kata benda, konjungsi subordinatif, konjungsi koordinatif tanpa koma, konjungsi koordinatif dengan koma, konjungsi korelatif, konjungsi antar kalimat, tanda kurung, dan tanda tengah kalimat.</li> <li>- Membuat laporan monitoring dan evaluasi magang U-Tapis untuk pihak internal UMN dalam bentuk <i>slide</i> presentasi dan video perkembangan.</li> <li>- Mempelajari buku "Tata Baku Bahasa Indonesia Edisi Keempat" yang telah diberikan oleh Ibu Niknik Mediyawati, S.Pd., M.Hum untuk meningkatkan performa dari <i>tag-tag</i> dan <i>rule-rule</i> CFG yang sudah ada.</li> </ul>
8	<ul style="list-style-type: none"> <li>- Mempelajari buku "Tata Baku Bahasa Indonesia Edisi Keempat" yang telah diberikan oleh Ibu Niknik Mediyawati, S.Pd., M.Hum untuk meningkatkan performa dari <i>tag-tag</i> dan <i>rule-rule</i> CFG yang sudah ada.</li> <li>- Merombak ulang algoritma <i>Context Free Grammar</i> (CFG) untuk mencakup elemen-elemen Subjek, Predikat, Objek, Pelengkap (sebelumnya <i>rule</i> untuk membuat kalimat masih berupa frasa nominal, frasa verbal, dan lain-lain; bukan elemen S-P-O-K).</li> </ul>
Lanjut pada halaman berikutnya	

Tabel 3.1 Pelaksanaan kerja magang tiap minggu (lanjutan)

Minggu	Pekerjaan yang dilakukan
9	<ul style="list-style-type: none"> <li>- Mengembangkan algoritma CFG yang telah dibuat sesuai dengan informasi tambahan dari buku "Tata Baku Bahasa Indonesia Edisi Keempat".</li> <li>- Menuliskan dokumentasi untuk algoritma CFG.</li> <li>- Melakukan <i>testing</i> pada algoritma CFG yang telah dibuat dan memperbaiki bug yang ditemukan selama <i>testing</i> algoritma CFG.</li> </ul>
10	<ul style="list-style-type: none"> <li>- Melakukan <i>manual tagging</i> pada data-data berita Tribun News sesuai dengan <i>tag-tag</i> yang sudah diperbaharui.</li> <li>- Membuat data contoh-contoh kalimat yang mengandung konjungsi koordinatif, konjungsi subordinatif, konjungsi antar kalimat, konjungsi antar paragraf, dan preposisi.</li> <li>- Menyiapkan kode untuk membaca data <i>manual tagging</i> tersebut dan melakukan <i>training</i> algoritma CRF.</li> </ul>
11	<ul style="list-style-type: none"> <li>- Melakukan <i>manual tagging</i> pada data-data berita Tribun News.</li> <li>- Memperbaiki algoritma CFG sesuai dengan isu yang ditemukan selama melakukan <i>manual tagging</i>.</li> <li>- Melakukan <i>training</i> pada algoritma CRF dengan data <i>manual tagging</i> tersebut.</li> <li>- Melakukan <i>testing</i> terhadap algoritma CRF tersebut.</li> </ul>
12	<ul style="list-style-type: none"> <li>- Memperbaiki <i>bug</i> yang ditemukan pada fungsi <i>preprocessing</i>, algoritma CRF, dan algoritma CFG.</li> <li>- Melakukan <i>testing</i> terhadap algoritma CRF tersebut.</li> <li>- Membuat <i>Python Flask API Web Service</i> untuk <i>deployment</i> algoritma U-Tapis pengecekan sintaksis kalimat.</li> <li>- Menyiapkan data <i>manual tagging</i> untuk evaluasi performa algoritma CRF dan CFG.</li> <li>- Mengikuti pertemuan laporan monitoring dan evaluasi tahap satu Matching Fund Kedaireka 2022.</li> </ul>
Lanjut pada halaman berikutnya	

Tabel 3.1 Pelaksanaan kerja magang tiap minggu (lanjutan)

Minggu	Pekerjaan yang dilakukan
13	<ul style="list-style-type: none"> <li>- Memperbaiki <i>bug</i> yang ditemukan pada fungsi <i>preprocessing</i>, algoritma CRF, dan algoritma CFG.</li> <li>- Menyiapkan data <i>manual tagging</i> untuk evaluasi performa algoritma CRF dan CFG.</li> <li>- Memperbaharui <i>Python Flask API Web Service</i> untuk menyesuaikan dengan perbaikan <i>bug</i> pada fungsi <i>preprocessing</i>, algoritma CRF, dan algoritma CFG.</li> <li>- Mengisi buku panduan U-Tapis terkait algoritma pengecekan sintaksis kalimat.</li> <li>- Mempelajari dan melakukan <i>hosting</i> pada Python Anywhere sebagai solusi sementara untuk <i>deployment</i> kode <i>Python Flask API Web Service</i> karena <i>Virtual Private Server</i> (VPS) internal milik UMN masih belum siap dan <i>deadline</i> pengujian U-Tapis di kantor Tribun News Jakarta sudah dekat.</li> </ul>
14	<ul style="list-style-type: none"> <li>- Melakukan <i>hosting</i> kode <i>Python Flask API Web Service</i> pada <i>Virtual Private Server</i> (VPS) milik UMN.</li> <li>- Berkoordinasi dengan Tim <i>front-end website</i> U-Tapis terkait pengaplikasian algoritma pengecekan sintaksis kalimat di <i>website</i> tersebut.</li> <li>- Melakukan uji coba U-Tapis di Tribun News Jakarta.</li> <li>- Membuat video perkembangan algoritma pengecekan sintaksis kalimat U-Tapis.</li> <li>- Mengikuti pertemuan laporan monitoring dan evaluasi tahap dua Matching Fund Kedaireka 2022.</li> <li>- Melakukan optimasi pada kode <i>Python Flask API Web Service</i> dan men-<i>deploy</i> ulang di <i>Virtual Private Server</i> (VPS) milik UMN.</li> <li>- Menambahkan fitur <i>log message</i> yang akan ditampilkan pada <i>server</i> yang menjalankan kode <i>Python Flask API Web Service</i> tersebut setiap kali terdapat <i>request</i> yang masuk.</li> <li>- Memantau performa <i>server</i> di VPS UMN dan melaporkan permasalahan VPS yang ditemukan ke Tim Dosen U-Tapis.</li> </ul>
Lanjut pada halaman berikutnya	

Tabel 3.1 Pelaksanaan kerja magang tiap minggu (lanjutan)

Minggu	Pekerjaan yang dilakukan
15	<ul style="list-style-type: none"> <li>- Memantau performa <i>server</i> di VPS UMN dan melaporkan permasalahan VPS yang ditemukan ke Tim Dosen U-Tapis</li> <li>- Melakukan optimasi pada algoritma <i>CFG parsing</i> dengan mengganti <i>parser</i> dari <i>ChartParser</i> menjadi <i>SteppingChartParser</i> dengan <i>Left-Corner Strategy</i> dan menerapkan <i>early stopping</i> bila ditemukan minimal satu kalimat yang berhasil di-<i>parsing</i> (untuk menghemat penggunaan RAM).</li> <li>- Menambahkan <i>query parameter</i> untuk <i>Python Flask API Web Service</i> sehingga <i>request</i> bisa meminta data <i>boolean</i> hasil <i>parsing</i> saja (<i>bool-only</i>) atau data kalimat yang memiliki sintaksis salah saja (<i>false-only</i>).</li> <li>- Melakukan rapat perkembangan progres dengan tim <i>front-end website</i> dan tim <i>machine learning</i> lainnya. Rapat ini dilakukan bersama dengan Ibu Marlinda Vasty Overbeek, S.Kom, M.Kom, Bapak Moeljono Widjaja, B.Sc., M.Sc., Ph.D, dan Bapak Rudi Sutomo, S.Kom., M.Si., M.Kom.</li> <li>- Menambahkan fitur di algoritma <i>preprocessing</i> untuk melakukan filtrasi <i>HTML tag</i>.</li> </ul>
Lanjut pada halaman berikutnya	





Tabel 3.1 Pelaksanaan kerja magang tiap minggu (lanjutan)

Minggu	Pekerjaan yang dilakukan
16	<ul style="list-style-type: none"> <li>- Memperbaiki <i>bug</i> pada algoritma <i>preprocessing</i> untuk menangani <i>escaped whitespace</i> (lambang <i>whitespace</i> <code>\\n</code> dan <code>\\r</code> akan diubah menjadi <code>\n</code> sedangkan lambang <i>whitespace</i> <code>\\t</code>, <code>\\f</code>, dan <code>\\v</code> akan diubah menjadi tanda spasi " ").</li> <li>- Membantu tim <i>front-end website</i> untuk melakukan <i>deployment</i> di <i>Virtual Private Server</i> terpisah milik UMN.</li> <li>- Mengubah bentuk <i>response</i> yang diberikan oleh <i>Python Flask API Web Service</i>.</li> <li>- Melakukan <i>deployment</i> kode <i>Python Flask API Web Service</i> di VPS UMN terbaru (VPS dengan kapasitas RAM yang lebih besar dari VPS sebelumnya).</li> <li>- Membantu tim <i>front-end website</i> U-Tapis memperbaiki pengelolaan <i>API request</i> untuk API U-Tapis pengecekan sintaksis kalimat.</li> <li>- Melakukan Uji Coba U-Tapis di Kantor Tribun News Jakarta.</li> <li>- Melakukan Uji Coba U-Tapis di Kantor Tribun News Surakarta.</li> </ul>
17	<ul style="list-style-type: none"> <li>- Berdiskusi dengan tim <i>machine learning</i> peluluhan kata bahasa Indonesia terkait <i>deployment</i> di VPS U-Tapis pengecekan sintaksis kalimat.</li> <li>- Melakukan <i>deployment</i> algoritma peluluhan kata bahasa Indonesia di VPS U-Tapis pengecekan sintaksis kalimat.</li> <li>- Melengkapi laporan HAKI (Hak Atas Kekayaan Intelektual) algoritma U-Tapis pengecekan sintaksis kalimat.</li> </ul>

### 3.4 Cara Kerja Algoritma U-Tapis Pengecekan Sintaksis Kalimat

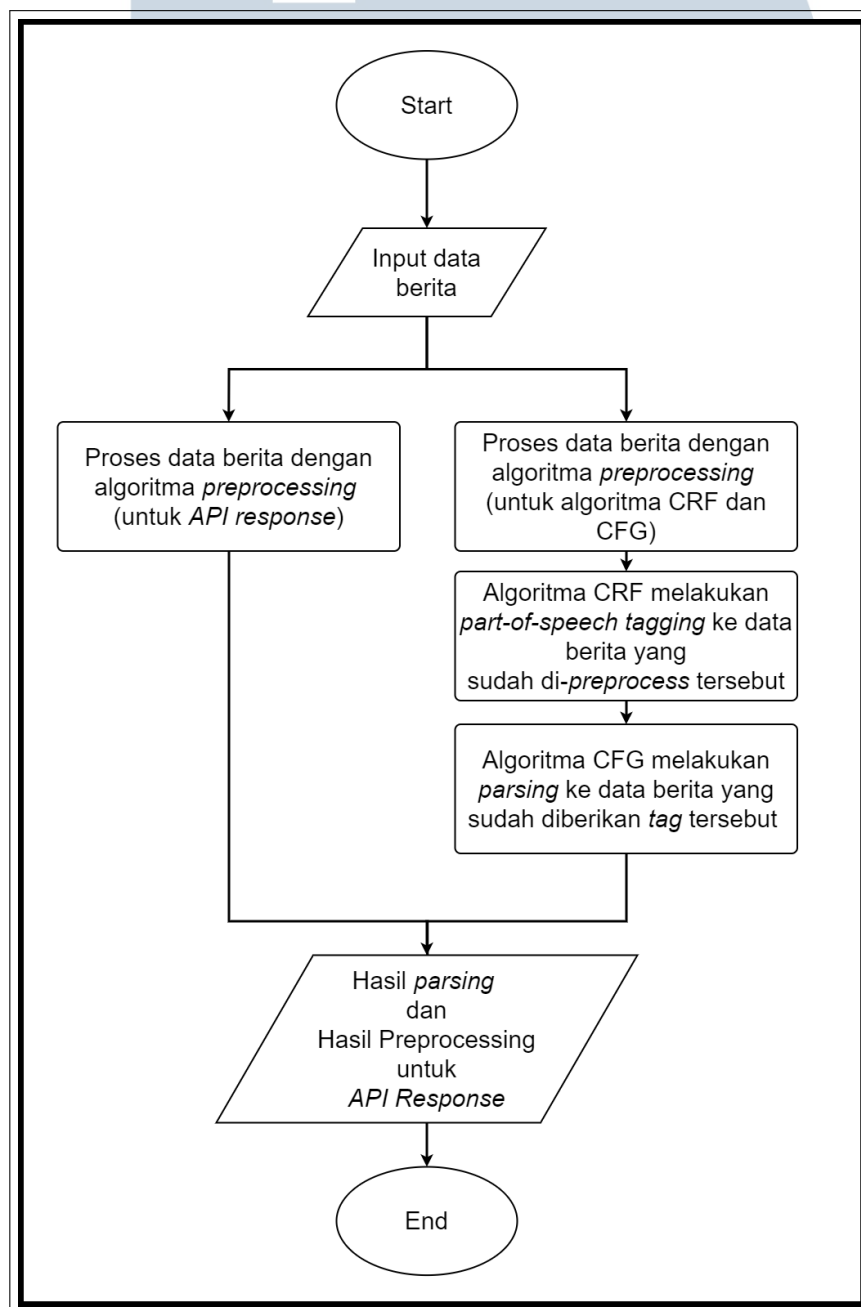
Algoritma U-Tapis pengecekan sintaksis kalimat ini dikembangkan menggunakan bahasa pemrograman Python. Hal ini dilakukan karena Python memiliki banyak *package* yang dapat membantu mempercepat waktu pengembangan algoritma.

Algoritma ini terdiri dari empat algoritma terpisah, yaitu:

1. Algoritma *Preprocessing* (untuk Algoritma CRF dan CFG).

2. Algoritma *Preprocessing* (untuk *API Response*).
3. Algoritma *Conditional Random Field (CRF)*.
4. Algoritma *Context Free Grammar (CFG)*.

Flowchart cara kerja algoritma U-Tapis pengecekan sintaksis kalimat dapat dilihat pada Gambar 3.1.



Gambar 3.1. Flowchart cara kerja algoritma U-Tapis pengecekan sintaksis kalimat

Saat data teks berita di-*input* ke dalam algoritma U-Tapis pengecekan sintaksis kalimat, dua algoritma *preprocessing* akan mengolah data tersebut terlebih dahulu. Algoritma *preprocessing* (untuk algoritma CRF dan CFG) akan membuang semua simbol-simbol yang tidak relevan, memecah data berita menjadi per kalimat, dan memecah tiap kalimat tersebut menjadi per kata/simbol (*tokenization*). Algoritma *preprocessing* (untuk *API response*) akan memecah data berita per kalimat tanpa melakukan perubahan apapun pada data berita tersebut.

Data berita yang telah diolah oleh algoritma *preprocessing* untuk algoritma CRF dan CFG tersebut kemudian dikirimkan ke algoritma *Conditional Random Field (CRF)* untuk dilakukan proses *tagging/labeling*, yaitu proses pemberian label-label pada setiap kata/simbol dari setiap kalimat. Label-label yang diberikan akan sesuai dengan peran kata/simbol tersebut dalam suatu kalimat (nomina, verba, adjektiva, adverbial, dan lain-lain).

Data berita yang telah di-*tagging* ini lalu akan dikirimkan ke algoritma *Context Free Grammar (CFG)* untuk dilakukan proses *parsing*. Algoritma *Context Free Grammar (CFG)* akan menggunakan *rule-rule* kalimat yang baik dan benar untuk menentukan apakah data kalimat-kalimat berita tersebut dapat dibentuk (*parse*) dengan *rule-rule* kalimat tersebut. Bila dapat dibentuk (*parse*), kalimat tersebut memiliki sintaksis kalimat yang sesuai dengan kaidah bahasa Indonesia yang baik dan benar. Bila TIDAK dapat dibentuk (*parse*), kalimat tersebut memiliki sintaksis kalimat yang TIDAK sesuai dengan kaidah bahasa Indonesia yang baik dan benar.

Selama tahap pengembangan ini, algoritma U-Tapis pengecekan sintaksis kalimat di-*deploy* di Virtual Private Server (VPS) milik Universitas Multimedia Nusantara (UMN) dalam bentuk *Application Programming Interface (API)*. Hal ini dilakukan agar komponen U-Tapis lainnya (*website*, dan lain-lain) dapat menggunakan algoritma ini dengan mudah (tanpa perlu kode khusus untuk setiap platform). *Application Programming Interface (API)* akan mengembalikan hasil *parsing* dan hasil *preprocessing* untuk *API response*.

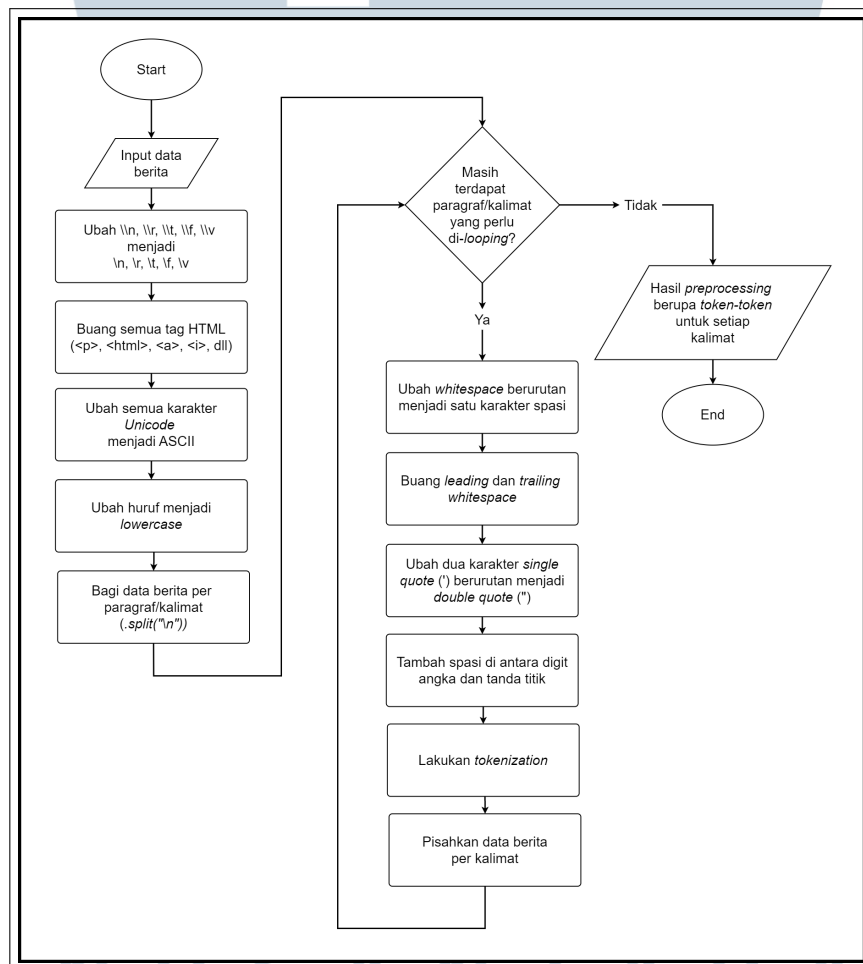
### **3.4.1 Algoritma Preprocessing (untuk Algoritma CRF dan CFG)**

Algoritma *preprocessing* untuk algoritma CRF dan CFG ini digunakan untuk mempersiapkan data berita yang diterima oleh API menjadi bentuk yang dapat digunakan oleh algoritma *Conditional Random Field (CRF)* dan *Context Free Grammar (CFG)*.

Berikut adalah *Python package* yang digunakan untuk algoritma *preprocessing*:

- anyascii [11].
- bs4 (*BeautifulSoup4*) [12].
- nltk (*Natural Language Toolkit*) [13].
- re (*Regular Expression*).

Flowchart dari algoritma *preprocessing* untuk algoritma CRF dan CFG dapat dilihat pada Gambar 3.2.



Gambar 3.2. Flowchart algoritma *preprocessing* untuk CRF dan CFG

Berikut adalah tahapan-tahapan yang dilakukan oleh algoritma *preprocessing*:

**1. Menghilangkan semua escape character (\) dari setiap whitespace character (\n, \r, \t, \f, \v).**

Hal ini dilakukan karena *Flask API framework* (*framework* yang digunakan untuk *deployment* algoritma ini) akan secara otomatis melakukan *escape* (penambahan tanda \) untuk setiap whitespace character (\n, \r, \t, \f, \v). Meski hal ini dilakukan untuk keamanan *framework*, hal ini menyebabkan *whitespace character* tersebut tidak dapat diolah oleh kode-kode setelahnya sehingga harus dikembalikan seperti semula. Proses penghilangan tanda *escape* (\) ini dilakukan dengan kode `re.sub()`.

**2. Menghilangkan semua HTML tag.**

Input data berita yang diberikan bisa jadi mengandung HTML tag yang harus dibuang (<p>, <html>, <a>, <i>, <br>, dan lain-lain). Hal ini dapat dilakukan dengan menginisialisasi *instance* dari *class* `BeautifulSoup(data_berita, "html.parser")` dan memanggil *method* `.get_text()`. *Method* itu akan mengembalikan data teks berita yang sudah tidak mengandung *HTML tag*.

**3. Mengubah semua unicode character menjadi ASCII character.**

Semua *unicode character* di dalam input data berita akan diubah menjadi *ASCII character* yang ekuivalen. Hal ini dikarenakan algoritma CRF dan CFG menganggap karakter-karakter *unicode* sebagai karakter yang berbeda. Sebagai contoh, karakter kutip dua kiri (“), karakter kutip dua kanan (”), dan karakter kutip dua biasa (") akan dianggap sebagai karakter yang berbeda meski memiliki arti yang sama. Proses pengubahan *unicode* menjadi *ASCII* ini dilakukan dengan kode `anyascii()`.

**4. Mengubah data berita menjadi huruf kecil (lowercase).**

Algoritma CRF bersifat *case-sensitive* sehingga kata yang sama akan dianggap berbeda bila mengandung huruf kecil/huruf besar yang berbeda. Oleh karena itu, semua data berita akan diubah menjadi *lowercase* untuk menghilangkan permasalahan ini. Proses ini dilakukan dengan *Python built-in function*, yaitu `.lower()`.

**5. Membagi data berita per paragraf/kalimat dengan separator \n.**

Algoritma *preprocessing* akan mengasumsikan semua data teks yang dipisahkan dengan *newline* (`\n`) sebagai data paragraf/kalimat yang terpisah dengan data paragraf/kalimat sebelumnya. Proses ini dilakukan dengan *Python built-in function*, yaitu `.split("\n")`.

**6. Looping untuk membuang karakter-karakter tidak relevan dan memisahkan data berita menjadi per kalimat.**

Data teks yang telah dipecah dengan `.split("\n")` tersebut masih bisa berupa paragraf. Oleh karena itu, dilakukan *looping* untuk tiap data teks tersebut agar dapat dilakukan *preprocessing* lebih lanjut dan dilakukan pemisahan per kalimat.

Berikut adalah tahapan-tahapan yang dilakukan di dalam tiap iterasi *looping* tersebut:

**(a) Mengubah *whitespace characters* berurutan menjadi satu spasi.**

Menggunakan `re.sub()` untuk menggantikan semua karakter *whitespace* yang berurutan menjadi satu karakter spasi.

**(b) Membuang *leading* dan *trailing whitespace*.**

Membuang karakter *whitespace* yang terdapat di awal dan di akhir data teks berita. Proses ini dilakukan dengan menggunakan *Python built-in function*, yaitu `.strip()`

**(c) Mengubah *double single quote*(') menjadi *double quote* (").**

Beberapa data berita menggunakan dua tanda petik tunggal (*double single quote*) untuk merepresentasikan tanda petik ganda. Tanda ini diubah menjadi tanda petik ganda biasa dengan menggunakan `re.sub()`.

**(d) Menambahkan spasi di antara tanda angka dan tanda titik.**

Kode untuk melakukan pemisahan kata/symbol (*tokenization*) akan menganggap angka dan titik setelahnya sebagai satu kesatuan. Untuk menghindari itu, maka perlu diterapkan algoritma untuk memisahkan angka dari tanda titik di paling akhir (`1234. → 1234 .`). Hal ini dilakukan dengan `re.sub()`.

**(e) Melakukan *tokenization*.**

Dengan menggunakan `nltk.tokenize.word_tokenize()`, data berita akan di-*tokenize*. Proses ini akan memisahkan data kalimat menjadi

kata/symbol.

(f) **Melakukan pemisahan data kalimat.**

Menerapkan *custom code* untuk memisahkan data berita yang telah di-*tokenize* tersebut per kalimat. Bila ditemukan tanda akhir kalimat (.?! ) di dalam atau di luar kutipan, teks tersebut dipisahkan menjadi kalimat tersendiri. Bila sudah tidak lagi ditemukan tanda akhir kalimat dan semua data berita telah habis, data berita tersebut digabungkan menjadi satu kalimat tersendiri.

Karena kalimat di dalam kutipan tidak wajib sesuai dengan kaidah bahasa Indonesia yang baik dan benar (tidak bisa diuji kesesuaiannya secara sintaksis), semua kata-kata dan simbol-simbol di dalam kutipan (kecuali simbol , . ! ? di paling akhir kutipan) akan dibuang.

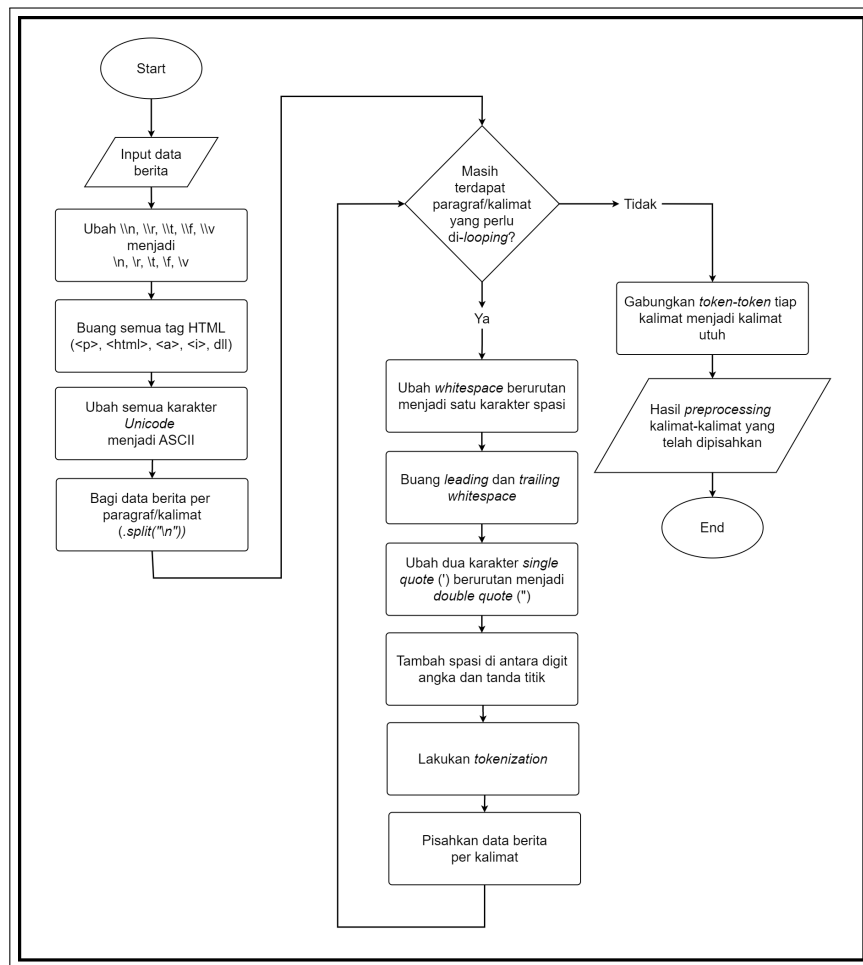
### 3.4.2 Algoritma Preprocessing (untuk API Response)

Algoritma *preprocessing* untuk *API response* memiliki cara kerja yang sama dengan algoritma *preprocessing* untuk algoritma CRF dan CFG. Perbedaan dari algoritma ini yaitu:

- Kata-kata dan simbol-simbol di dalam kutipan tidak dibuang.
- Kata-kata tidak di-*lowercase*.
- Setelah proses pemisahan kalimat selesai, setiap *token* untuk tiap kalimat akan disatukan kembali menjadi satu data teks. Dengan demikian, hasil akhir dari algoritma ini adalah data-data teks kalimat yang sudah dipisahkan.

*Flowchart* dari algoritma *preprocessing* untuk *API response* dapat dilihat pada Gambar 3.3.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.3. Flowchart algoritma *preprocessing* untuk *API response*

API akan menggunakan algoritma *preprocessing* (untuk *API response*) ini untuk mengembalikan hasil pemisahan data berita berupa data teks kalimat-kalimat bersama dengan hasil *parsing* kalimat-kalimat tersebut (hasil pengecekan sintaksis kalimat berupa *list of boolean*).

### 3.4.3 Algoritma Conditional Random Field (CRF)

Algoritma *Conditional Random Field* (CRF) merupakan algoritma yang bertugas untuk melakukan *part-of-speech tagging*, yaitu pemberian label/tag ke setiap *token-token* (kata-kata/symbol-symbol yang telah dipecah untuk setiap data kalimat). Algoritma ini mengaplikasikan *pattern recognition* dalam pemberian label/tag untuk tiap *token*. Hal ini berarti *state/kondisi* dari informasi data di dekatnya (*neighboring data*) akan memengaruhi hasil prediksi. Oleh karena itu, algoritma ini bisa menangani kata-kata/symbol-symbol baru [14].

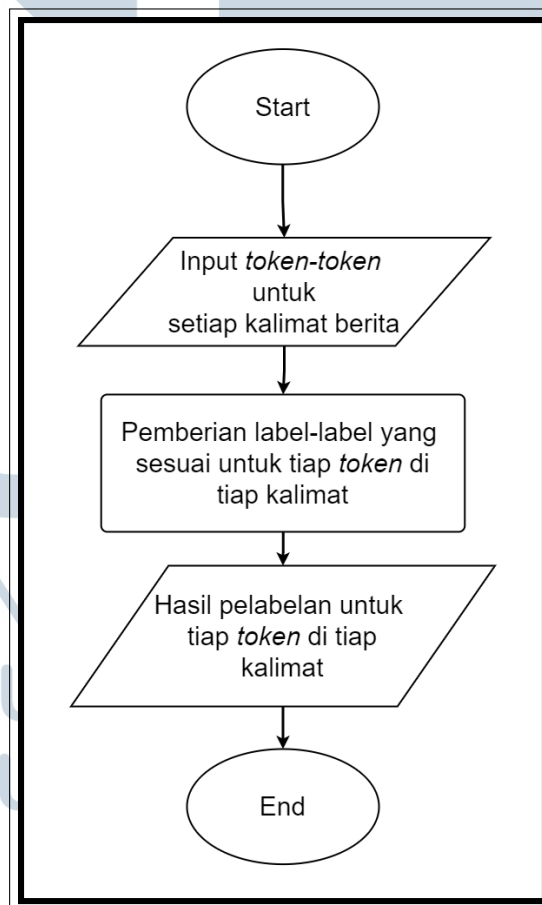


Untuk mengembangkan algoritma ini, perlu dilakukan *preprocessing* pada beberapa data berita Tribun News dan dilakukan pelabelan manual (*manual labeling/tagging*) untuk setiap kata-kata dan simbol-simbol dalam setiap kalimat dari data-data berita tersebut. Data pelabelan manual ini kemudian diberikan ke algoritma *Conditional Random Field (CRF)* sebagai *training data*. Setelah proses *training* selesai, algoritma *Conditional Random Field (CRF)* dapat melakukan proses *tagging/labeling*, baik pada kata-kata dan simbol-simbol yang pernah dipelajari maupun yang belum. *Model CRF* yang sudah di-*training* ini akan disimpan di dalam file `.crf.tagger`.

Berikut adalah *Python package* yang digunakan untuk algoritma *Conditional Random Field (CRF)*:

- `nltk.tag.CRFTagger` [15].

*Flowchart* dari algoritma *Conditional Random Field (CRF)* dapat dilihat pada Gambar 3.4.



Gambar 3.4. Flowchart algoritma *Conditional Random Field (CRF)*

Rincian *tag-tag* yang digunakan oleh algoritma *Conditional Random Field* (CRF) dan *Context Free Grammar* (CFG) dapat dilihat pada Tabel 3.2.

Tabel 3.2. *Tag-tag* CRF dan CFG

Tag	Arti
<nomina>	Kata nomina.
<numeralia>	Kata numeralia.
<verba>	Kata verba.
<adjektiva>	Kata adjektiva.
<adverbia>	Kata adverbia.
<preposisi>	Kata preposisi.
<bukan>	Kata adverbia spesial 'bukan' (hanya bisa diikuti oleh nomina, numeralia, frasa nominal, atau frasa numeralia).
<entah>	Konjungsi korelatif spesial 'entah... entah...'
<jangankan>	Konjungsi korelatif spesial 'jangankan..., ... pun...'
<dengan>	Kata preposisi atau kata konjungsi subordinatif 'dengan'.
<pun>	Artikel atau adverbia 'pun'.
<artikel>	Kata artikula/artikel.
<konj_koor_tak_hingga>	Kata konjungsi koordinatif tak hingga (bisa menghubungkan dua atau lebih kata, frasa, atau klausa), yaitu 'dan', 'serta', 'atau', 'dan/atau'.
<konj_koor_terhingga>	Kata konjungsi koordinatif terhingga (hanya bisa menghubungkan dua kata, frasa, atau klausa), yaitu 'tetapi'.
Lanjut pada halaman berikutnya	

Tabel 3.2 *Tag-tag* CRF dan CFG (lanjutan)

<b>Tag</b>	<b>Arti</b>
<konj_koor_terhingga_klausa	Kata konjungsi koordinatif terhingga klausa (hanya bisa menghubungkan dua klausa), yaitu 'melainkan', 'padahal', 'sedangkan', 'lalu', 'kemudian', 'selanjutnya', 'setelahnya', 'sebelumnya', dan lain-lain.
<konj_sub_tnp_koma_kt_pertama	Kata pertama dari konjungsi subordinatif yang tidak membutuhkan koma ('oleh' karena, 'oleh' sebab, 'sehingga', dan lain-lain).
<konj_sub_tnp_koma_kt_kedua	Kata kedua dari konjungsi subordinatif yang tidak membutuhkan koma (oleh 'karena', oleh 'sebab').
<konj_sub_dgn_koma>	Konjungsi subordinatif yang membutuhkan koma (bahkan, yaitu, yakni, kecuali, selain, asal).
<konj_sub_yang>	Konjungsi subordinatif 'yang'.
<konj_sub_bahwa>	Konjungsi subordinatif 'bahwa'.
<konj_sub_berbeda>	Konjungsi subordinatif spesial 'berbeda dengan...' karena hanya bisa diikuti dengan frasa nominal.
<konj_atr_pertama>	Kata pertama dari konjungsi antar kalimat dan antar paragraf.
<konj_atr_kedua>	Kata kedua dari konjungsi antar kalimat dan antar paragraf.
<konj_atr_ketiga>	Kata ketiga dari konjungsi antar kalimat dan antar paragraf.
<konj_atr_adapun>	Konjungsi antar kalimat dan antar paragraf spesial karena tidak membutuhkan tanda koma.
<td_tengah_kal>	Tanda tengah kalimat : ... - -
<td_akhir_kal>	Tanda akhir kalimat .?!
Lanjut pada halaman berikutnya	

Tabel 3.2 *Tag-tag* CRF dan CFG (lanjutan)

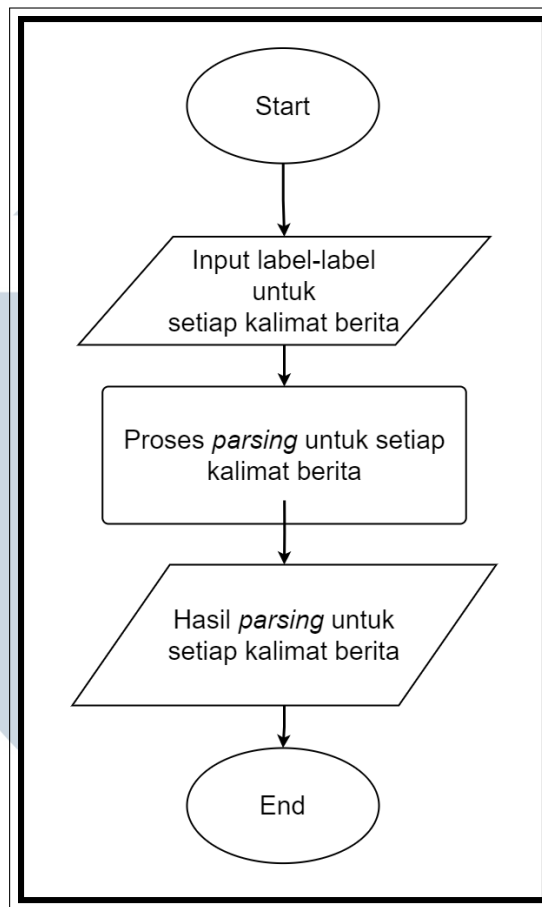
Tag	Arti
<td_koma>	Tanda koma , ;
<kurung_buka>	Kurung buka ( [ { <
<kurung_tutup>	Kurung tutup ) ] } >
<kutip_awal>	Tanda kutip awal "
<kutip_akhir>	Tanda kutip akhir "
<simbol>	Simbol-simbol lainnya @ # \$ % ^ & *

#### 3.4.4 Algoritma Context Free Grammar (CFG)

Setelah data berita diberikan label/tag, data berita tersebut akan di-*parsing* oleh algoritma *Context Free Grammar* (CFG) untuk diuji kesesuaian sintaksisnya dengan kaidah bahasa Indonesia yang baik dan benar (S-P-O-K). Algoritma *Context Free Grammar* adalah kumpulan *rule* yang bisa digunakan untuk menghasilkan *pattern* secara rekursif. Untuk mengatasi jumlah kata-kata bahasa Indonesia yang hampir tak terhingga (bahasa Indonesia dengan kaidah yang benar, bahasa Indonesia dengan kaidah yang salah, bahasa gaul, bahasa asing, bahasa dengan salah tik, dan lain-lain), algoritma CFG ini akan menggunakan *tag-tag* yang telah ditentukan dalam *rule-rule* sintaksis yang baik dan benar (bukan data kata mentah, melainkan *tag-tag* hasil algoritma *Conditional Random Field* (CRF)).

*Flowchart* dari algoritma *Context Free Grammar* (CFG) dapat dilihat pada Gambar 3.5.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.5. Flowchart algoritma *Context Free Grammar (CFG)*

*Rule-rule* dari algoritma CFG ini dituliskan di sebuah *file* terpisah dengan tipe file `.cfg`. File ini kemudian akan dibaca dengan `nltk.data.load()` dan dihubungkan dengan `nltk.parse.chart.SteppingChartParser` yang menggunakan strategi *Left Corner Strategy* (strategi *parsing* yang menelusuri *rule-rule* mulai dari *rule* paling kiri-bawah dan kemudian perlahan mencoba *rule* di kanan-atas). `SteppingChartParser` kemudian akan melakukan *parsing* dan mengembalikan *parse tree* pertama yang ditemukan bila sintaksis benar atau mengembalikan *list* kosong bila sintaksis salah [16].

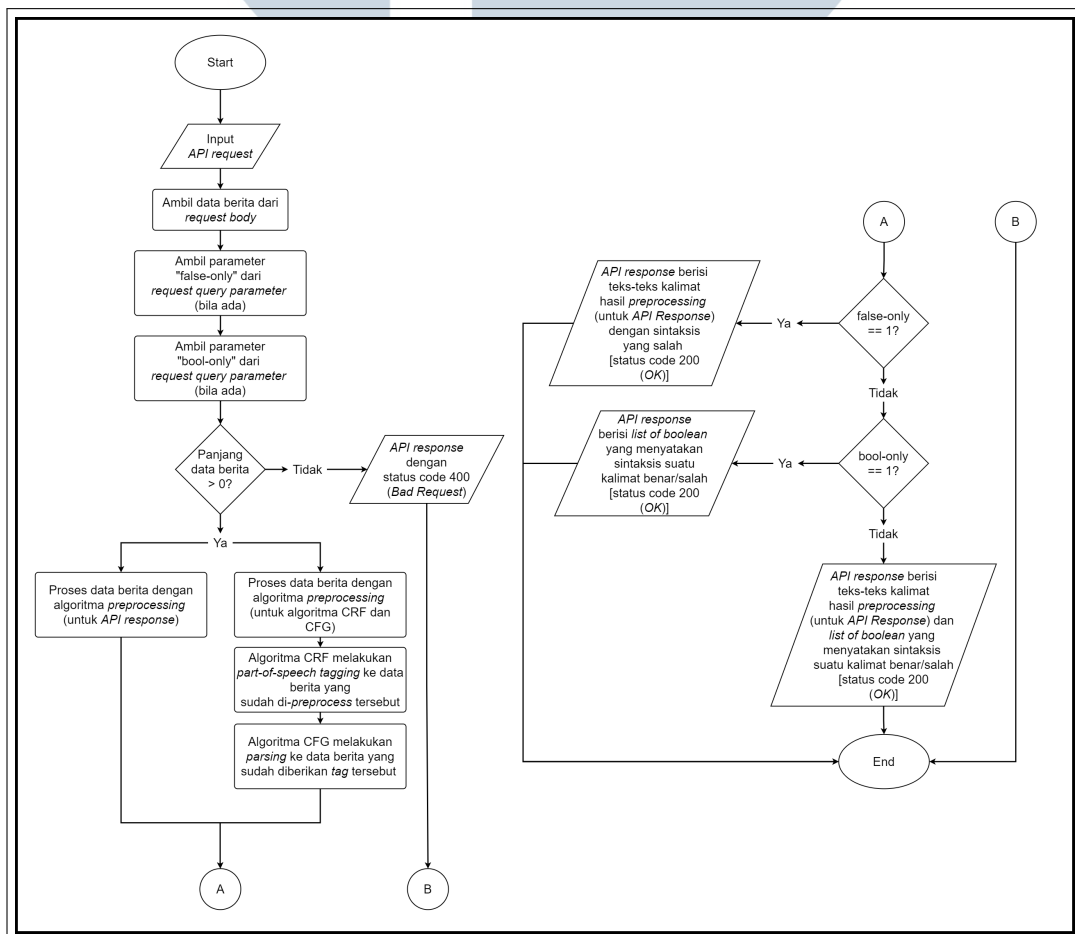
*Rule-rule* dari algoritma CFG ini memiliki 14.242 baris *rule* yang digunakan untuk menentukan apakah suatu kalimat memiliki sintaksis yang benar atau tidak. *Rule-rule* ini dimanfaatkan dengan menggunakan `SteppingChartParser` agar algoritma CFG bisa dihentikan bila berhasil menemukan minimal satu *parse tree* tanpa harus menemukan kemungkinan *parse tree* lainnya. Teknik ini bernama *early stopping* dan teknik ini digunakan untuk menghemat penggunaan waktu dan memori.

Berikut adalah *Python package* yang digunakan untuk algoritma *Context Free Grammar (CFG)*:

- `nltk.tag.chart.SteppingChartParser` [16].
- `nltk.tag.chart.FilteredSingleEdgeFundamentalRule` [16].
- `nltk.tag.chart.LeafInitRule` [16].
- `nltk.tag.chart.FilteredBottomUpPredictCombineRule` [16].
- `nltk.data` [17].

### 3.4.5 Deployment Algoritma sebagai API

*Flowchart* cara kerja API U-Tapis pengecekan sintaksis kalimat dapat dilihat pada Gambar 3.6.



Gambar 3.6. Flowchart cara kerja API U-Tapis pengecekan sintaksis kalimat

Setelah pengembangan algoritma-algoritma U-Tapis pengecekan sintaksis kalimat tersebut selesai dibuat, algoritma-algoritma tersebut di-*deploy* di *Virtual Private Server* (VPS) milik Universitas Multimedia Nusantara (UMN) dengan menggunakan *Python Flask Web Framework*. *Framework* ini dipilih karena bersifat *lightweight* (tidak memakan banyak *resource* untuk menggunakannya) dan dapat digunakan untuk menjalankan algoritma yang ditulis pada bahasa pemrograman Python.

Algoritma ini di-*deploy* di *endpoint* `http://{{hostname}}:{{port}}/utapis-cek-sintaksis-kal`. Nilai dari variabel `hostname` dan `port` bergantung pada pengaturan di *Virtual Private Server* (VPS) saat *deployment*. Karena pengembangan algoritma U-Tapis pengecekan sintaksis kalimat ini menggunakan *Virtual Private Server* (VPS) milik Universitas Multimedia Nusantara (UMN), pengguna API haruslah terhubung dengan *Virtual Private Network* (VPN) milik UMN.

Untuk menggunakan API U-Tapis pengecekan sintaksis kalimat, pengguna harus mengirimkan sebuah *request* ke *endpoint* `http://{{hostname}}:{{port}}/utapis-cek-sintaksis-kal` dengan *method* POST dan menyertakan *request body* yang berisi data form dengan elemen bernama `article` yang berisi data berita yang ingin diuji kebenaran sintaksisnya.

Setelah pengecekan sintaksis data berita tersebut selesai, pengguna akan mendapatkan *API response* berupa data *JavaScript Object Notation* (JSON) (lihat Kode 3.1).

```
1 {
2   "results": [
3     {
4       "is_valid": boolean,
5       "sentence": string
6     },
7     ...
8   ]
9 }
```

Kode 3.1: Format *API response* (default)

API U-Tapis pengecekan sintaksis kalimat juga dapat menerima *query parameter*, yaitu `bool-only` atau `false-only`. Opsi `bool-only` akan membuat *API response* hanya berisi *list-of-boolean* (lihat Kode 3.2).

```
1 {
2   "results": [
```

```

3     {
4         "is_valid": boolean
5     },
6     ...
7 ]
8 }

```

Kode 3.2: Format API response (bool-only)

Opsi false-only akan membuat API response hanya berisi data teks kalimat yang memiliki kesalahan sintaksis (lihat Kode 3.3).

```

1 {
2     "results": [
3         {
4             "sentence": string
5         },
6         ...
7     ]
8 }

```

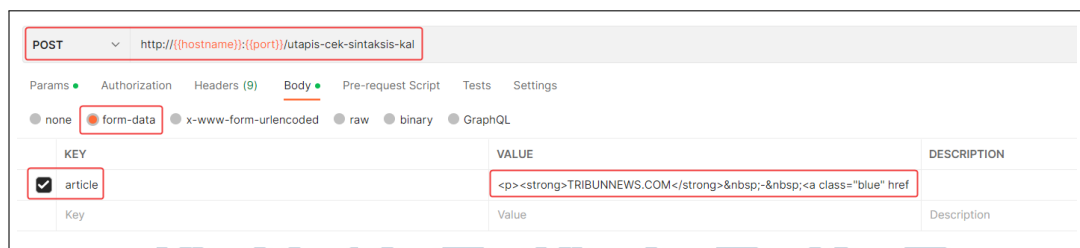
Kode 3.3: Format API response (false-only)

### 3.5 Contoh Cara Kerja Algoritma U-Tapis Pengecekan Sintaksis Kalimat

Berikut adalah langkah-langkah kerja API algoritma U-Tapis pengecekan sintaksis kalimat:

#### 1. Pengguna mengirimkan data berita ke API endpoint

Data berita ini bisa dikirimkan melalui *website* atau aplikasi lainnya. Untuk demonstrasi ini, data berita akan dikirimkan menggunakan aplikasi pengujian API bernama *Postman* (lihat Gambar 3.7).



Gambar 3.7. API request menggunakan Postman

Contoh *input* data berita yang dikirimkan dapat dilihat pada Kode 3.4



```

1 <p><strong>TRIBUNNEWS.COM</strong>&nbsp;-&nbsp;
2 <a class="blue" href="https://www.tribunnews.com/
3 tag/lembaga-perlindungan-saksi-dan-korban">
4 Lembaga Perlindungan Saksi dan Korban</a>&nbsp;(LPSK)
5 mengajukan rekomendasi permohonan keringanan hukuman
6 terhadap&nbsp;
7 <a class="blue" href="https://www.tribunnews.com/tag/
8 bharada-richard-eliezer">Bharada Richard
9 Eliezer</a>&nbsp;atau Bharada E dalam kasus
pembunuhan
10 Brigadir Nofriansyah
11 Yosua Hutabarat atau&nbsp;<a class="blue"
12 href="https://www.tribunnews.com/tag/brigadir-j">
Brigadir
13 J</a>.</p>\r\n<p>Pasalnya , Bharada Eliezer bersedia
14 menjadi Justice Collaborator (JC) dalam membuka kasus
ini .
15 </p>\r\n<p>Apalagi kasus ini
16 menyangkut kebohongan seorang perwira polri yang
17 seharusnya menjadi penegak hukum</p>

```

Kode 3.4: Contoh *input* data berita

## 2. Data berita melalui tahap preprocessing

Terdapat dua jenis algoritma *preprocessing* yang digunakan pada data berita tersebut, yaitu algoritma *preprocessing* untuk algoritma CRF dan CFG serta algoritma *preprocessing* untuk *API response*. Hasil *preprocessing* untuk algoritma CRF dan CFG dapat dilihat di Kode 3.5 sedangkan hasil *preprocessing* untuk *API response* dapat dilihat di Kode 3.6.

```

1 [
2 [
3 "tribunnews.com",
4 "-",
5 "lembaga",
6 "perlindungan",
7 "saksi",
8 "dan",
9 "korban",
10 "(",
11 "lpsk",
12 ")",
13 "mengajukan",

```

14 "rekomendasi",  
15 "permohonan",  
16 "keringanan",  
17 "hukuman",  
18 "terhadap",  
19 "bharada",  
20 "richard",  
21 "eliezer",  
22 "atau",  
23 "bharada",  
24 "e",  
25 "dalam",  
26 "kasus",  
27 "pembunuhan",  
28 "brigadir",  
29 "nofriansyah",  
30 "yosua",  
31 "hutabarat",  
32 "atau",  
33 "brigadir",  
34 "j",  
35 ".",  
36 ],  
37 [  
38 "pasalnya",  
39 ",",  
40 "bharada",  
41 "eliezer",  
42 "bersedia",  
43 "menjadi",  
44 "justice",  
45 "collaborator",  
46 "(",  
47 "jc",  
48 ")",  
49 "dalam",  
50 "membuka",  
51 "kasus",  
52 "ini",  
53 ".",  
54 ],  
55 [  
56 "apalagi",

```

57     "kasus" ,
58     "ini" ,
59     "menyangkut" ,
60     "kebohongan" ,
61     "seorang" ,
62     "perwira" ,
63     "polri" ,
64     "yang" ,
65     "seharusnya" ,
66     "menjadi" ,
67     "penegak" ,
68     "hukum" ,
69     "." ,
70 ],
71 ]

```

Kode 3.5: Hasil *preprocessing* untuk algoritma CRF dan CFG

```

1 [
2     "TRIBUNNEWS.COM-Lembaga Perlindungan Saksi dan Korban
3     (LPSK) mengajukan rekomendasi permohonan keringanan
4     hukuman terhadap Bharada Richard Eliezer atau
5     Bharada E dalam kasus pembunuhan Brigadir
6     Nofriansyah Yosua Hutabarat atau Brigadir J.",
7     "Pasalnya, Bharada Eliezer bersedia menjadi Justice
8     Collaborator (JC) dalam membuka kasus ini.",
9     "Apalagi kasus ini menyangkut kebohongan seorang
10    perwira polri yang seharusnya menjadi penegak
11    hukum.",
12 ]

```

Kode 3.6: Hasil *preprocessing* untuk API response

### 3. Data berita di-parsing dengan Conditional Random Field (CRF) dan di-tagging dengan Context Free Grammar (CFG).

Hasil *preprocessing* untuk algoritma CRF dan CFG tersebut kemudian diberikan ke algoritma *Conditional Random Field* (CRF) untuk proses pelabelan/*tagging* dan ke algoritma *Context Free Grammar* (CFG) untuk proses *parsing*. Hasil dari algoritma CRF dan CFG dapat dilihat pada Kode 3.7.

```

1 (Utapis Sintaksis - Default) New request received from
   127.0.0.1 at 2023-Jan-01 11:19:20
2 Found 3 sentences!

```

```

3 Sentence 1/3 = [('tribunnews.com', '<nomina>'), ('-', '<
td_tengah_kal>'), ('lembaga', '<nomina>'), ('perlindungan',
, '<nomina>'), ('saksi', '<nomina>'), ('dan', '<
konj_koor_tak_hingga>'), ('korban', '<nomina>'), ('(', '<
kurung_buka>'), ('lpsk', '<nomina>'), (')', '<kurung_tutup
>'), ('mengajukan', '<verba>'), ('rekomendasi', '<nomina>'),
), ('permohonan', '<nomina>'), ('keringanan', '<nomina>'),
('hukuman', '<nomina>'), ('terhadap',
4 '<preposisi>'), ('bharada', '<nomina>'), ('richard', '<nomina
>'), ('eliezer', '<nomina>'), ('atau', '<
konj_koor_tak_hingga>'), ('bharada', '<preposisi>'), ('e',
'<nomina>'), ('dalam', '<preposisi>'), ('kasus', '<nomina
>'), ('pembunuhan', '<nomina>'), ('brigadir', '<nomina>'),
('nofriansyah', '<nomina>'), ('yosua', '<numeralia>'), ('
hutabarat', '<nomina>'), ('atau', '<konj_koor_tak_hingga>'),
), ('brigadir', '<nomina>'), ('j', '<nomina>'), ('.', '<
td_akhir_kal>')]: True!
5 Sentence 2/3 = [('pasalnya', '<konj_atr_pertama>'), (',', '<
td_koma>'), ('bharada', '<preposisi>'), ('eliezer', '<
nomina>'), ('bersedia', '<nomina>'), ('menjadi', '<verba>'),
), ('justice', '<nomina>'), ('collaborator', '<nomina>'),
('(', '<kurung_buka>'), ('jc', '<nomina>'), (')', '<
kurung_tutup>'), ('dalam', '<preposisi>'), ('membuka', '<
nomina>'), ('kasus', '<nomina>'), ('ini', '<nomina>'), ('.
', '<td_akhir_kal>')]: True!
6 Sentence 3/3 = [('apalagi', '<preposisi>'), ('kasus', '<
nomina>'), ('ini', '<nomina>'), ('menyangkut', '<nomina>'),
), ('kebohongan', '<nomina>'), ('seorang', '<nomina>'), ('
perwira', '<nomina>'), ('polri', '<nomina>'), ('yang', '<
konj_sub_yang>'), ('seharusnya', '<adverbia>'), ('menjadi',
, '<verba>'), ('penegak', '<nomina>'), ('hukum', '<nomina
>')]: False!
7 (Utapis Sintaksis – Default) Request from 127.0.0.1 finished
at 2023-Jan-01 11:19:25 (Status: 200)!

```

Kode 3.7: Hasil *tagging* dengan CRF dan *parsing* dengan CFG

#### 4. API mengembalikan response terhadap data berita tersebut.

Setelah pemrosesan data dengan algoritma *Conditional Random Field* (CRF) dan algoritma *Context Free Grammar* (CFG) selesai dilakukan, API akan mengembalikan *response* ke pengguna. Hasil yang diberikan bergantung pada *query parameter* yang diberikan oleh pengguna.

*API response* tanpa *query parameter* dapat dilihat pada Kode 3.8.

```
1 {
2   "results": [
3     {
4       "is_valid": true ,
5       "sentence": "TRIBUNNEWS.COM-Lembaga Perlindungan
6         Saksi dan Korban (LPSK) mengajukan rekomendasi permohonan
7         keringanan hukuman terhadap Bharada Richard Eliezer atau
8         Bharada E dalam kasus pembunuhan Brigadir Nofriansyah
9         Yosua Hutabarat atau Brigadir J."
10    },
11   {
12     "is_valid": true ,
13     "sentence": "Pasalnya , Bharada Eliezer bersedia
14       menjadi Justice Collaborator (JC) dalam membuka kasus ini.
15     "
16   },
17   {
18     "is_valid": false ,
19     "sentence": "Apalagi kasus ini menyangkut
20       kebohongan seorang perwira polri yang seharusnya menjadi
21       penegak hukum"
22   }
23 ]
24 }
```

Kode 3.8: *API response* (default/tanpa *query parameter*)

*API response* dengan *query parameter* false-only dapat dilihat pada Kode 3.9.

```
1 {
2   "results": [
3     {
4       "sentence": "Apalagi kasus ini menyangkut
5         kebohongan seorang perwira polri yang seharusnya menjadi
6         penegak hukum"
7     }
8   ]
9 }
```

Kode 3.9: *API response* (false-only)

*API response* dengan *query parameter* bool-only dapat dilihat pada Kode 3.10.

```

1 {
2   "results": [
3     {
4       "is_valid": true
5     },
6     {
7       "is_valid": true
8     },
9     {
10      "is_valid": false
11    }
12  ]
13 }

```

Kode 3.10: API response (bool-only)

### 3.6 Hasil Evaluasi Algoritma CRF dan CFG

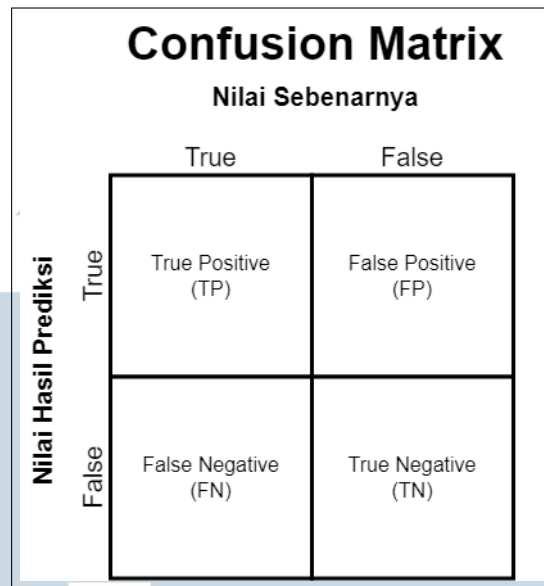
Untuk menguji akurasi dari algoritma *Conditional Random Field* (CRF) dan *Context Free Grammar* (CFG), digunakan lima artikel berita Tribun News yang belum pernah digunakan pada kedua algoritma tersebut sebelumnya. Data-data berita ini diproses secara manual dan akan dibandingkan dengan hasil prediksi dari algoritma CRF dan CFG untuk pengujian akurasi kedua algoritma tersebut.

Algoritma *Conditional Random Field* (CRF) akan dievaluasi dengan perhitungan akurasi sedangkan algoritma *Context Free Grammar* (CFG) akan dievaluasi dengan perhitungan *confusion matrix* yang kemudian akan digunakan untuk perhitungan *precision*, *recall*, *accuracy* dan *F1-Score*.

Akurasi merupakan perhitungan perbandingan antara total hasil prediksi yang benar dengan total keseluruhan prediksi yang dilakukan. Perhitungan akurasi dilakukan dengan membagi jumlah prediksi yang benar dengan total prediksi yang dilakukan (lihat Rumus 3.1) [18].

$$Akurasi = \frac{Jumlah\ prediksi\ yang\ benar}{Jumlah\ keseluruhan\ prediksi} \quad (3.1)$$

*Confusion matrix* merupakan teknik untuk mengukur performa dari sebuah algoritma dengan menghitung total *true positive*, *true negative*, *false positive*, dan *false negative* (lihat Gambar 3.8) [18].



Gambar 3.8. *Confusion Matrix*

Hasil perhitungan dari *confusion matrix* tersebut kemudian dapat digunakan untuk menghitung metrik lainnya, yaitu *precision*, *recall*, *accuracy*, dan *F1-Score*. *Precision* adalah rasio antara total nilai positif yang benar-benar bernilai positif dengan total nilai-nilai positif yang diberikan oleh algoritma (lihat Rumus 3.2). Nilai *precision* yang tinggi menandakan performa algoritma yang baik (nilai maksimal = 1,0; nilai minimal = 0,0) [18].

$$Precision = \frac{TP}{TP + FP} \quad (3.2)$$

*Recall* adalah rasio antara total nilai positif yang diberikan oleh algoritma dengan total nilai-nilai yang seharusnya bernilai positif (lihat Rumus 3.3). Nilai *recall* yang tinggi menandakan performa algoritma yang baik (nilai maksimal = 1,0; nilai minimal = 0,0) [18].

$$Recall = \frac{TP}{TP + FN} \quad (3.3)$$

*Accuracy* adalah rasio antara total prediksi-prediksi yang benar dengan total keseluruhan prediksi yang dilakukan. Saat menggunakan *confusion matrix*, *accuracy* dihitung dengan membagi total *true positive* dan *true negative* dengan total seluruh prediksi yang dilakukan (lihat Rumus 3.4). Nilai *accuracy* yang tinggi menandakan performa algoritma yang baik (nilai maksimal = 1,0; nilai minimal =

0,0) [18].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.4)$$

*F1-Score* merupakan teknik kalkulasi untuk menghitung performa suatu algoritma berdasarkan *precision* dan *recall* secara bersamaan. Teknik ini menggunakan *harmonic mean* untuk meningkatkan penalti (pengurangan nilai *F1-Score*) bila algoritma memiliki nilai *precision* dan/atau *recall* yang rendah (lihat Rumus 3.5). Nilai *F1-Score* yang tinggi menandakan performa algoritma yang baik (nilai maksimal = 1,0; nilai minimal = 0,0) [18].

$$F1-Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3.5)$$

Berikut adalah kelima berita Tribun News yang digunakan untuk evaluasi algoritma CRF dan CFG:

- Eksepsi Baiquni Ditolak, Hakim Perintahkan Teruskan Perkara Perintangan Penyidikan Kasus Brigadir J [19].
- Insiden Atap SD Runtuh Memakan Korban Jiwa, Ini Kata Sri Sultan HB X dan Disdik Gunungkidul [20].
- Subvarian Omicron XBB Menambah Kasus Aktif, Pemerintah Perpanjang PPKM Level 1 Seluruh Wilayah RI [21].
- Telkom Kembali Jadi Satu-Satunya Perusahaan Indonesia di Jajaran Forbes 2022 *World's Best Employer* [22].
- U-Tapis, Kecerdasan Buatan Tapis Kesalahan Bahasa Jurnalis [4].

Hasil evaluasi algoritma *Conditional Random Field* (CRF) per artikel berita dapat dilihat pada Tabel 3.3.

Tabel 3.3. Hasil evaluasi CRF per artikel

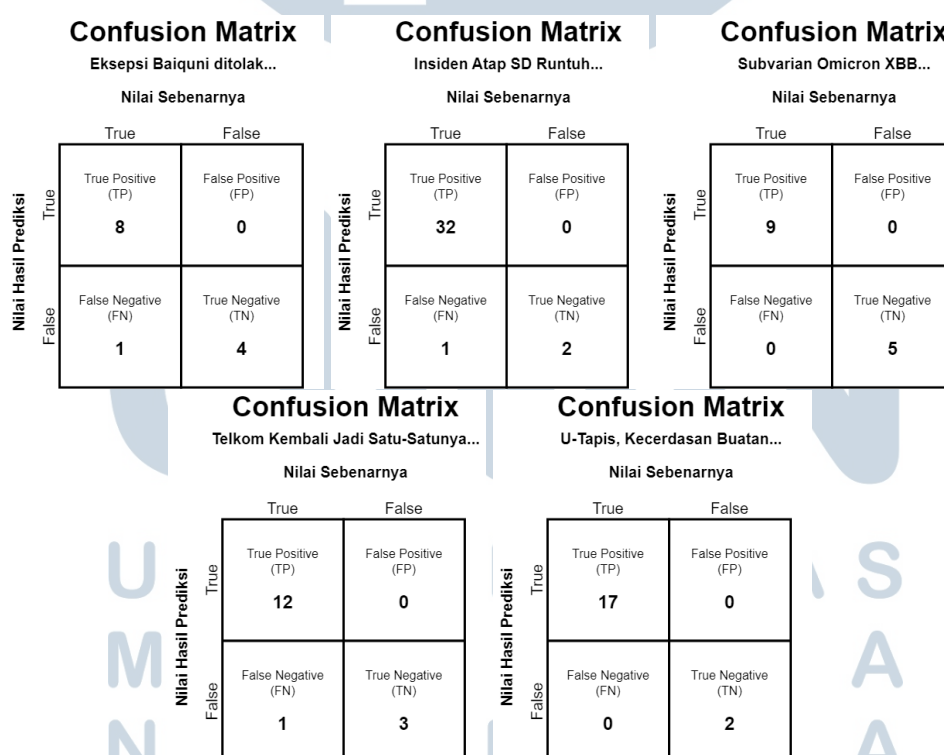
Berita	Akurasi
Eksepsi Baiquni Ditolak, Hakim Perintahkan Teruskan Perkara Perintangan Penyidikan Kasus Brigadir J.	0,9298892988929889
Lanjut pada halaman berikutnya	



Tabel 3.3 Hasil evaluasi CRF per artikel (lanjutan)

Berita	Akurasi
Insiden Atap SD Runtuh Memakan Korban Jiwa, Ini Kata Sri Sultan HB X dan Disdik Gunungkidul.	0,9275053304904051
Subvarian Omicron XBB Menambah Kasus Aktif, Pemerintah Perpanjang PPKM Level 1 Seluruh Wilayah RI.	0,9291666666666667
Telkom Kembali Jadi Satu-Satunya Perusahaan Indonesia di Jajaran Forbes 2022 World's Best Employer.	0,9036458333333334
U-Tapis, Kecerdasan Buatan Tapis Kesalahan Bahasa Jurnalis.	0,920844327176781

Hasil evaluasi algoritma *Context Free Grammar* (CFG) per artikel berita dapat dilihat pada Gambar 3.9 dan Tabel 3.4.



Gambar 3.9. Confusion Matrix dari kelima artikel berita

Tabel 3.4. Hasil evaluasi CFG per artikel

Berita	Precision	Recall	Accuracy	F1-Score
Eksepsi Baiquni Ditolak...	1,0	0,8888	0,9231	0,9412
Insiden Atap SD Runtuh...	1,0	0,9697	0,9714	0,9846
Subvarian Omicron XBB Menambah...	1,0	1,0	1,0	1,0
Telkom Kembali jadi Satu-Satunya...	1,0	0,9231	0,9375	0,96
U-Tapis, Kecerdasan Buatan...	1,0	1,0	1,0	1,0

Nilai akurasi yang didapatkan algoritma *Conditional Random Field* (CRF) saat artikel berita digabungkan menjadi satu evaluasi adalah 0,920844327176781 sedangkan hasil evaluasi algoritma *Context Free Grammar* (CFG) saat artikel-artikel berita tersebut digabungkan menjadi satu evaluasi dapat dilihat pada Gambar 3.10 dan Tabel 3.5.

UIN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

		Confusion Matrix	
		Gabungan semua artikel berita	
		Nilai Sebenarnya	
		True	False
Nilai Hasil Prediksi	True	True Positive (TP) <b>78</b>	False Positive (FP) <b>0</b>
	False	False Negative (FN) <b>3</b>	True Negative (TN) <b>16</b>

Gambar 3.10. Confusion Matrix gabungan kelima artikel berita

Tabel 3.5. Hasil evaluasi CFG gabungan kelima artikel berita

Precision	Recall	Accuracy	F1-Score
1,0	0,963	0,9691	0,9811

Berdasarkan hasil evaluasi yang telah dilakukan, algoritma *Conditional Random Field* (CRF) memiliki akurasi sekitar 90% hingga 93%, sedangkan algoritma *Context Free Grammar* (CFG) memiliki nilai *precision* 1,0; *recall* sekitar 0,88 hingga 1,0; *accuracy* sekitar 0,92 hingga 1,0; dan *F1-Score* sekitar 0,94 hingga 1,0. Hasil ini menunjukkan bahwa algoritma *Conditional Random Field* (CRF) dan *Context Free Grammar* (CFG) memiliki tingkat performa yang tinggi.

### 3.7 Hasil Uji Coba Algoritma U-Tapis Pengecekan Sintaksis Kalimat di Tribun News Jakarta, Surakarta, dan Makassar

Untuk menguji performa dari API (*Application Programming Interface*) algoritma U-Tapis pengecekan sintaksis kalimat, response yang diberikan oleh API akan diuji dari segi waktu penyelesaian dan akurasi prediksi yang diberikan.

Perhitungan waktu response akan dilakukan dari waktu *request* dikirimkan ke API hingga waktu *response* dikembalikan oleh API. Perhitungan akurasi prediksi yang diberikan API akan dilakukan dengan menggunakan *confusion matrix* yang akan digunakan untuk menghitung nilai *precision*, *recall*, *accuracy*, dan *F1-Score*.

Data hasil uji coba algoritma U-Tapis pengecekan sintaksis kalimat di Tribun News Jakarta, Surakarta, dan Makassar dapat dilihat pada Tabel 3.6, Tabel 3.7, dan Tabel 3.8.

Tabel 3.6. Hasil uji coba Tribun News Jakarta

Berita	Kalimat	Waktu Penyelesaian Algoritma (detik)	Hasil Prediksi	Hasil yang Diharapkan
1	Kejutan belum mereda di Piala Dunia 2022.	8,63	true	true
	Spanyol yang berstatus favorit juara, tumbang oleh tim underdog, Maroko di babak 16 besar Piala Dunia 2022, Kamis (6/12/2022) dengan skor 0-3.		true	true
	Secara statistik, skuad asuhan Luis Enrique unggul dalam segala hal, mulai dari penguasaan bola hingga jumlah operan.		false	true
	Hanya, La Furia Roja tak bisa memanfaatkan keunggulan itu untuk mencetak satu gol pun ke gawang Maroko		false	false

Tabel 3.7. Hasil uji coba Tribun News Surakarta

Berita	Kalimat	Waktu Penyelesaian Algoritma (detik)	Hasil Prediksi	Hasil yang Diharapkan
1	Presiden Joko Widodo meminta maaf kepada masyarakat yang merasa terganggu kenyamanannya selama proses pernikahan putra bungsunya, Kaesang Pangarep dengan Erina Gudono.	4,45	true	true
	Hal tersebut khususnya disampaikan presiden untuk warga Yogyakarta dan Solo.		true	true
	Kemudian, Jokowi menyampaikan rasa syukur setelah proses siraman putra bungsunya berlangsung lancar, pada Jumat (9/12/2022) pagi.		true	true
	Dirinya juga meminta doa restu kepada masyarakat agar rangkaian prosesi perkawinan Kaesang dan Erina lancar.		true	true
Lanjut pada halaman berikutnya				

Tabel 3.7 Hasil uji coba Tribun News Surakarta (lanjutan)

Berita	Kalimat	Waktu Penyelesaian Algoritma (detik)	Hasil Prediksi	Hasil yang Diharapkan
	Sebelumnya, Jokowi memimpin prosesi siraman Kaesang di kediaman keluarga di Sumber, Solo.		true	true
2	Momen mengharukan terjadi saat Brasil dikalahkan oleh Kroasia di babak perempat final piala dunia 2022 Qatar.	4,66	true	true
	Sang Bintang Timnas Brasil, Neymar jr tak kuasa menahan air matanya dan menumpahkannya ketika masih berada di lapangan.		true	true
	Uniknya, anak laki-laki dari seorang pemain Kroasia justru datang menghampiri Neymar.		true	true
	Pada video yang beredar di media sosial (medsos), tampak bocah laki-laki tersebut berusaha menghibur Neymar yang sedang bersedih.		true	true
Lanjut pada halaman berikutnya				

Tabel 3.7 Hasil uji coba Tribun News Surakarta (lanjutan)

Berita	Kalimat	Waktu Penyelesaian Algoritma (detik)	Hasil Prediksi	Hasil yang Diharapkan
	Usut punya usut, anak kecil itu adalah putra dari Ivan Perisic.		true	true
3	Putra sulung Gibran Rakabuming, Jan Ethes Srinarendra dan adik sepupunya, Sedah Mirah Nasution, mewakili Presiden Joko Widodo (Jokowi) sekeluarga untuk wawancara.	8,19	true	true
	Jan Ethes dan Sedah Mirah tampak tersenyum dan melambaikan tangan ke para awak media usai prosesi Midodareni di rumah Erina Gudono.		true	true
	Di belakang keduanya ada Presiden Jokowi dan Ibu Negara Iriana, Kaesang, Gibran, dan istrinya, Selvi Ananda, hingga Kahiyang Ayu dan Bobby Nasution.		true	true
	Sementara itu, cucu pertama Presiden Jokowi ini tampak tidak canggung di depan puluhan kamera dan mikrofon media.		true	true
Lanjut pada halaman berikutnya				

Tabel 3.7 Hasil uji coba Tribun News Surakarta (lanjutan)

Berita	Kalimat	Waktu Penyelesaian Algoritma (detik)	Hasil Prediksi	Hasil yang Diharapkan
	Sempat ditanya bertemu siapa saja di dalam, apakah bertemu dengan Erina calon istri Kaesang?		true	true
4	Keluarga Presiden Jokowi telah menjalani prosesi Midodareni di rumah Erina Gudono di Purwosari, Sinduadi, Sleman, Jumat (09/12/2022) malam.	3,34	false	true
	Presiden Jokowi mengatakan se usai Midodareni itu, ia dan keluarga besar datang ke Yogyakarta untuk midodareni dan melakukan serah-serahan, juga prosesi adat Tilik Nilik.		true	true
	Tilik Nilik itu hanya boleh dilakukan oleh para perempuan.		true	true
	Kaesang sang calon pengantin tak boleh bertemu Erina.		false	true
Lanjut pada halaman berikutnya				



Tabel 3.7 Hasil uji coba Tribun News Surakarta (lanjutan)

Berita	Kalimat	Waktu Penyelesaian Algoritma (detik)	Hasil Prediksi	Hasil yang Diharapkan
5	Tinggal selangkah lagi, Kaesang Pangarep mempersunting Erina Gudono, hari ini, Sabtu (10/12/2022).	6,44	false	false
	Akad nikah Kaesang Pangarep dan Erina Gudono dijadwalkan berlangsung pukul 12.00 WIB di Pendopo Agung Royal Ambarukmo, Yogyakarta.		true	true
	Jelang prosesi akad nikah kediaman Erina Gudono tampak sudah mulai sepi, tak ada aktivitas yang mencolok.		true	true
	Dari pantauan Tribunjogja.com, tampak tenda mulai dibongkar, hanya ada beberapa pekerja yang sedang berkemas-kemas.		true	true
	Beberapa kendaraan pengangkut telah berada di sekitar rumah Erina Gudono.		true	true

Tabel 3.8. Hasil uji coba Tribun News Makassar

Berita	Kalimat	Waktu Penyelesaian Algoritma (detik)	Hasil Prediksi	Hasil yang Diharapkan
1	Harga saham perusahaan teknologi ramai-ramai anjlok menyebabkan indeks harga saham gabungan (IHSG) terperosok di zona merah.	3,25	true	true
	Dikutip dari Kontan, IHSG turun 24,381 poin atau 0,34 persen ke 7.057,8.		false	true
	Harga saham perusahaan teknologi ramai-ramai anjlok menyebabkan indeks harga saham gabungan (IHSG) terperosok di zona merah.		true	true
2	Partai Persatuan Pembangunan (PPP) mendapat giliran menjadi tuan rumah pertemuan konsolidasi dan silaturahmi nasional Koalisi Indonesia Bersatu atau KIB.	4,81	true	true
Lanjut pada halaman berikutnya				

Tabel 3.8 Hasil uji coba Tribun News Makassar (lanjutan)

Berita	Kalimat	Waktu Penyelesaian Algoritma (detik)	Hasil Prediksi	Hasil yang Diharapkan
	Ini akan jadi silaturahmi nasional ketiga setelah dua agenda sebelumnya di Jakarta oleh Partai Golkar, dan PAN di Surabaya.		false	false
	Konsolidasi ini akan membahas kelanjutan kemenangan Pemilihan Presiden 2024.		true	true
3	Tepat hari ini Kaesang Pangarep akan melepas masa lajangnya menikahi Erina Gudono.	2,01	true	true
	Akad nikah akan digelar di Royal Amburakmo, Yogyakarta, Sabtu 10 Desember 2022.		true	true
	Prosesi Sungkeman dan Siraman pun sudah digelar di kediaman Presiden Jokowi di Solo Jumat kemarin.		true	true
4	Pada akhir tahun 2022, Forbes kembali melansir daftar orang terkaya di Indonesia, termasuk perbandingan kekayaan bos Indomaret vs Alfamart.	5,92	true	true

Lanjut pada halaman berikutnya

Tabel 3.8 Hasil uji coba Tribun News Makassar (lanjutan)

Berita	Kalimat	Waktu Penyelesaian Algoritma (detik)	Hasil Prediksi	Hasil yang Diharapkan
	Forbes mencatat kekayaan kolektif dari 50 orang terkaya di Indonesia mencapai 180 miliar dollar AS atau setara Rp 2.808 triliun.		true	true
	Nilai tersebut melesat 11 persen dibandingkan tahun 2021 sebesar 162 miliar dollar AS atau setara Rp 2.527 triliun.		true	true
5	Pemerintah Desa Marioriaja, Kecamatan Marioriwawo, menanggapi pemberitaan media online Indoglobe News, terkait kondisi panggung lapangan Annukannung.	5,78	true	false
	Dalam berita yang diterbitkan pada Sabtu (6/8/2020), dengan judul penataan Lapangan Desa Mario Riaja, "Menyimpang", disebutkan pembangunan panggung tersebut menyimpang dari perencanaan.		false	false
Lanjut pada halaman berikutnya				

Tabel 3.8 Hasil uji coba Tribun News Makassar (lanjutan)

Berita	Kalimat	Waktu Penyelesaian Algoritma (detik)	Hasil Prediksi	Hasil yang Diharapkan
	Wartawan yang mengaku telah melakukan investigasi di lapangan berlokasi di Dusun Madello itu, menemukan kondisi bangunan panggung sudah retak.		true	false

*Confusion matrix* dari API algoritma U-Tapis pengecekan sintaksis kalimat dapat dilihat pada Gambar 3.11.

		Nilai Sebenarnya	
		True	False
Nilai Hasil Prediksi	True	True Positive (TP) <b>33</b>	False Positive (FP) <b>2</b>
	False	False Negative (FN) <b>4</b>	True Negative (TN) <b>4</b>

Gambar 3.11. *Confusion Matrix* uji coba U-Tapis

- *True Positive (TP)*: 33
- *True Negative (TN)*: 4

- *False Positive (FP)*: 2
- *False Negative (FN)*: 4
- Total: 43

Hasil perhitungan performa dari API algoritma U-Tapis pengecekan sintaksis kalimat dapat dilihat pada Rumus 3.6, Rumus 3.7, Rumus 3.8, Rumus 3.9, dan Rumus 3.10.

$$Precision = \frac{TP}{(TP + FP)} = \frac{33}{(33 + 2)} = 0,942857142 \quad (3.6)$$

$$Recall = \frac{TP}{(TP + FN)} = \frac{33}{(33 + 4)} = 0,891891 \quad (3.7)$$

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} = \frac{37}{43} = 0,860465 \quad (3.8)$$

$$F1-Score = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * 0,942857142 * 0,891891}{0,942857142 + 0,891891} = 0,916666195 \quad (3.9)$$

$$Average Response Time = \frac{Total Response Time}{Total Request} = \frac{57,84}{11} = 5,225454 \text{ detik.} \quad (3.10)$$

### 3.8 Kendala dan Solusi yang Ditemukan

Selama proses pengerjaan kegiatan magang U-Tapis ini, terdapat beberapa kendala yang dihadapi. Berikut adalah uraian kendala-kendala yang dihadapi dan solusi-solusi untuk setiap kendala tersebut.

#### 3.8.1 Kendala

Berikut adalah rincian kendala-kendala yang dihadapi selama proses pengerjaan kegiatan magang U-Tapis ini.

### **A. Kendala 1: Korpus Kata untuk Pengecekan Sintaksis Kalimat Terlalu Banyak**

Pada awal mula pengerjaan proyek algoritma U-Tapis pengecekan sintaksis kalimat, algoritma *Context Free Grammar* (CFG) akan menggunakan kompilasi kata-kata (*corpus*) bahasa Indonesia yang telah dikumpulkan secara manual. Kompilasi kata-kata ini kemudian akan digunakan bersama dengan *rule-rule* algoritma *Context Free Grammar* (CFG) untuk menentukan kebenaran sintaksis suatu teks kalimat. Namun, terdapat terlalu banyak kata bahasa Indonesia yang harus dikompilasi, termasuk kata berimbuhan, bahasa gaul, bahasa dengan kesalahan tik (*typo*), *named entity recognition* (kata yang digunakan sebagai nama manusia, tempat, organisasi, dan lain-lain), dan sebagainya. Oleh karena itu, diperlukan alternatif lain agar proyek ini dapat diselesaikan sesuai jadwal (jangka waktu satu semester).

### **B. Kendala 2: VPS Milik UMN Belum Siap Saat Mendekati Jadwal Uji Coba di Kantor Tribun News Jakarta**

Saat pengembangan algoritma U-Tapis pengecekan sintaksis kalimat telah selesai dan siap dilakukan uji coba di Kantor Tribun News Jakarta, terdapat permasalahan dalam pengiriman perangkat keras yang akan digunakan untuk menjalankan *Virtual Private Server* (VPS) milik UMN. Hal ini menyebabkan *Virtual Private Server* (VPS) tidak dapat digunakan saat mendekati jadwal uji coba di Kantor Tribun News Jakarta. Oleh karena itu, diperlukan alternatif *deployment* lainnya untuk uji coba di Kantor Tribun News Jakarta.

### **C. Kendala 3: Penggunaan RAM yang Tinggi oleh Algoritma CFG**

Saat dilakukan pengujian pada algoritma pengecekan sintaksis kalimat yang telah selesai dibuat, algoritma ini memiliki permasalahan di bidang optimasi. Untuk kalimat yang mengandung jumlah kata-kata yang banyak (20 kata atau lebih), penggunaan RAM oleh algoritma CFG akan mencapai lebih dari 16 GB. Hal ini menyebabkan *Virtual Private Server* (VPS) milik Universitas Multimedia Nusantara (UMN) berhenti berfungsi (*crash*).

### 3.8.2 Solusi

Berikut adalah rincian solusi-solusi untuk setiap kendala yang dihadapi selama proses pengerjaan kegiatan magang U-Tapis ini.

#### A. Solusi dari Kendala 1: Korpus Kata untuk Pengecekan Sintaksis Kalimat Terlalu Banyak

Solusi untuk kendala ini, yaitu dengan menggunakan algoritma *Conditional Random Field* (CRF) untuk mengatasi kata-kata baru yang belum pernah dipelajari oleh algoritma. Algoritma ini mengaplikasikan *pattern recognition* sehingga dapat menggunakan *state/kondisi* dari data-data di dekatnya (*neighboring data*) untuk menentukan hasil prediksi label/tag. Dengan demikian, algoritma ini dapat melakukan *tagging* pada kata-kata yang pernah dipelajari sebelumnya kemudian melakukan *pattern recognition* untuk kata-kata yang belum pernah dipelajari sebelumnya.

#### B. Solusi dari Kendala 2: VPS Milik UMN Belum Siap Saat Mendekati Jadwal Uji Coba di Kantor Tribun News Jakarta

Untuk mengatasi kendala ini, *deployment* algoritma U-Tapis pengecekan sintaksis kalimat dilakukan pada platform *web deployment* milik pihak ketiga, yaitu *PythonAnywhere*. Platform ini digunakan sebagai solusi sementara untuk tahap uji coba di Tribun News Jakarta. Setelah *Virtual Private Server* (VPS) milik UMN siap, *deployment* dilakukan di *Virtual Private Server* (VPS) milik UMN.

#### C. Solusi dari Kendala 3: Penggunaan RAM yang Tinggi oleh Algoritma CFG

Untuk mengatasi kendala ini, dilakukan beberapa optimasi pada algoritma CFG. Berikut adalah optimasi-optimasi yang dilakukan:

- Membuang *rule-rule* di dalam algoritma *Context Free Grammar* (CFG) yang bersifat berulang (*redundant*) dan memastikan tidak ada *rule-rule* yang dapat menyebabkan rekursi tak hingga (*infinite recursion*).
- Mengubah teknik *parsing* dari *ChartParser* menjadi *SteppingChartParser*. *SteppingChartParser* merupakan turunan



dari `ChartParser` yang memungkinkan pengelolaan langkah kerja secara manual. Hal ini memungkinkan penghentian proses *parsing* saat ditemukan minimal satu *parse tree* untuk menghemat waktu dan penggunaan RAM.

- Mengubah teknik *parsing* dari *Top-Down Strategy* menjadi *Left-Corner Strategy*. *Left-Corner Strategy* dapat menggabungkan keunggulan dari *Top-Down Strategy* (teknik *parsing* yang berfokus pada *rule* secara keseluruhan) dan *Bottom-Up Strategy* (teknik *parsing* yang berfokus pada *input* yang diterima) dengan membaca *rule-rule* CFG dari pojok kiri-bawah dan perlahan-lahan berpindah ke kanan-atas. Teknik ini dapat mengurangi penelusuran *rule-rule* yang tidak sesuai dengan *input* yang diberikan sehingga dapat menghemat waktu dan penggunaan RAM.

Kendala ini juga dilaporkan ke dosen-dosen pembimbing proyek U-Tapis sehingga dapat dilakukan peningkatan jumlah RAM yang dimiliki oleh *Virtual Private Server* milik UMN menjadi 32 GB.

