

**IDENTIFIKASI KASUS DAN PREDIKSI
PUTUSAN HAKIM BERDASARKAN LAMA PEMIDANAAN
DENGAN KLASIFIKASI *SUPPORT VECTOR MACHINE* (SVM)**

(Studi Kasus : Kasus Tindak Pidana di Pengadilan Negeri Sleman)

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Program
Studi Statistika



Disusun Oleh:

Rizky Desi Ramadhani

16611129

**PROGRAM STUDI STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2019

HALAMAN PERSETUJUAN PEMBIMBING

TUGAS AKHIR

Judul : IDENTIFIKASI KASUS DAN PREDIKSI PUTUSAN
HAKIM BERDASARKAN LAMA PEMIDANAAN
DENGAN KLASIFIKASI *SUPPORT VECTOR MACHINE*
(SVM)

Nama Mahasiswa : Rizky Desi Ramadhani

Nomor Mahasiswa : 16611129

TUGAS AKHIR INI TELAH DIPERIKSA DAN DISETUJUI UNTUK DIUJIKAN

Yogyakarta, 3 April 2020

Pembimbing



(Ayundyah Kesumawati, S.Si., M.Si.)

HALAMAN PENGESAHAN
TUGAS AKHIR

IDENTIFIKASI KASUS DAN PREDIKSI PUTUSAN HAKIM DENGAN KLASIFIKASI
SUPPORT VECTOR MACHINE (SVM)

(Studi Kasus : Kasus Tindak Pidana di Pengadilan Negeri Sleman)

Nama Mahasiswa : Rizky Desi Ramadhani

NIM : 16611129

TUGAS AKHIR INI TELAH DIUJIKAN
PADA TANGGAL: 3 April 2020

Nama Penguji:

Tanda Tangan

1. Karimatul Ummah, S.H., M.Hum.



2. Arum Handini Primandari, S.Pd.Si., M.Sc.



3. Ayundyah Kesumawati, S.Si., M.Si.



Mengetahui,

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Islam Indonesia



Prof. Riyanto, S.Pd. M.Si., Ph.D.

KATA PENGANTAR



Assalamu'alaikum Wr.Wb

Alhamdulillah *alhamdulillah*, segala puji bagi Allah SWT yang telah memberikan rahmat dan hidayah-Nya berupa keimanan, kekuatan, kesabaran, kelancaran, dan keselamatan sehingga peneliti dapat menyelesaikan penelitian ini dengan baik. Shalawat serta salam senantiasa tercurah kepada Rasulullah SAW yang mengantarkan manusia dari zaman kegelapan menuju ke zaman yang terang benderang ini. Penyusunan penelitian ini dimaksudkan sebagai Tugas Akhir untuk memenuhi sebagian syarat-syarat guna mencapai gelar Sarjana Statistika di Universitas Islam Indonesia. Penelitian ini berjudul “Identifikasi Kasus dan Prediksi Putusan Hakim dengan Klasifikasi *Support Vector Machine*”

Keberhasilan pembuatan penelitian ini tentunya tidak terlepas tanpa dukungan dari berbagai pihak baik moril maupun materil. Oleh karena itu, peneliti ingin menyampaikan ucapan terima kasih sebesar-besarnya kepada semua pihak yang telah membantu dalam penyusunan tugas akhir ini terutama kepada:

1. Bapak Prof. Riyanto, S.pd., M.Si., Ph.D., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia, Yogyakarta.
2. Bapak Dr. Edy Widodo, S.Si., M.Si., selaku Ketua Jurusan Statistika beserta seluruh jajarannya.
3. Ibu Ayundyah Kesumawati, S.Si., M.Si., selaku Dosen Pembimbing Tugas Akhir yang telah memberi bimbingan selama penyusunan Laporan Tugas Akhir ini.
4. Bapak Ari Wibowo, S.H.I., S.H., M.H., selaku Dosen Fakultas Hukum yang telah memberi masukan dan saran dalam penyusunan Laporan Tugas Akhir.
5. Pihak Pengadilan Negeri Sleman yang telah memudahkan dalam pengambilan data.
6. Ayah, Ibu, Adik Dimas, Adik Azzam serta Keluarga Besar yang selalu sabar

dan selalu mencurahkan doa, dukungan, dan semangat untuk peneliti.

7. Teman-teman dekat yaitu Alfi, Anis, Kiki, dan Widia yang sudah menemani, memberikan semangat serta yang selalu memberikan bantuan dan saran dalam memulai hingga mengakhiri selama masa kuliah dan tugas akhir ini.
8. Teman-teman seperjuangan dan sebimbingan yaitu Cinmey, Dea, Ella, Fauziyah, Laras, dan Mita yang selalu kompak untuk bimbingan dan banyak memberikan bantuan dalam mengerjakan tugas akhir ini.
9. Sabila salsabila, sebagai sahabat yang telah mendengarkan keluh kesah, selalu menemani dan memberikan semangat serta doa untuk peneliti dalam segala hal.
10. Sahabat Dhias, Icha, dan Putri sebagai sahabat yang selalu siap sedia mendengarkan keluh kesah dan memberikan saran dalam segala hal.
11. Farel, sebagai teman yang selalu memberi saran dan jalan keluar mengenai objek penelitian.
12. Teman-teman Statistika angkatan 2016 yang telah membantu dan memberikan pengalaman suka dan duka selama masa kuliah.
13. Dan, semua pihak yang tidak bisa peneliti sebutkan satu per satu. Terima kasih.

Peneliti juga menyadari sepenuhnya bahwa tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, segala kritik dan saran yang sifatnya membangun selalu peneliti harapkan. Semoga tugas akhir ini dapat bermanfaat bagi peneliti khususnya dan bagi semua yang membutuhkan. Akhir kata, semoga Allah SWT selalu melimpahkan rahmat serta hidayah-Nya kepada kita semua. *Aamiin ya Robbal 'Alamiin*

Wassalamualaikum Wr.Wb

Yogyakarta, 3 April 2020

Penulis

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN PEMBIMBING	ii
HALAMAN PENGESAHAN	iii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB 2 TINJAUAN PUSTAKA.....	4
2.1 Penelitian Terdahulu	4
BAB 3 LANDASAN TEORI.....	10
3.1 Revolusi Industri 4.0.....	10
3.2 Hukum.....	10
3.3 Hukum Pidana	10
3.3.1 Tindak Pidana.....	11
3.3.2 Jenis-Jenis Pidana	11
3.3.3 Acara Pemeriksaan	12
3.3.4 Pertanggungjawaban Pidana	13
3.4 Pengadilan Negeri (PN)	13
3.5 Prosedur Persidangan Pengadilan	13
3.6 Hakim.....	15
3.7 Putusan Pengadilan	15
3.8 Analisis Deskriptif	15
3.9 Kecerdasan Buatan (<i>Artificial Intelligence</i>).....	16
3.10 Pembelajaran Mesin (<i>Machine Learning</i>).....	17
3.11 <i>Natural Language Processing</i>	18
3.12 <i>Text Mining</i>	20

3.12.1	<i>Text Processing</i>	20
3.12.2	<i>Text Transformation</i>	21
3.13	Pembobotan <i>Term Frequency-Inverse Document Frequency</i> ...	21
3.14	Data <i>Training</i> dan Data <i>Testing</i>	23
3.15	<i>Support Vector Machine (SVM)</i>	24
3.16	Proses <i>Tuning Data</i>	27
3.17	Confusion Matrix	28
BAB 4	METODOLOGI PENELITIAN	30
4.1	Populasi dan Sampel	30
4.2	Sumber dan Jenis Data.....	30
4.3	Variabel Penelitian.....	30
4.4	Metode Analisis Data.....	31
4.5	Tahapan Penelitian.....	31
BAB 5	HASIL DAN PEMBAHASAN	33
5.1	Analisis Deskriptif	33
5.2	<i>Preprocessing Data</i>	36
5.3	Pembobotan TF-IDF	41
5.4	Data <i>Training</i> dan Data <i>Testing</i>	44
5.5	Klasifikasi <i>Support Vector Machine (SVM)</i>	45
5.6	Prediksi <i>Class</i> atau Kategori Untuk Data Tuntutan Baru	50
BAB 6	PENUTUP	53
6.1	Kesimpulan.....	53
6.2	Saran.....	53

DAFTAR TABEL

Tabel 3.1. Macam-Macam Fungsi <i>Kernel</i> (Sumber : Nugroho, 2003)	27
Tabel 3.2. <i>Confusion Matrix</i>	28
Tabel 4.1 Tabel Variabel dan Definisi Operasional Variabel.....	30
Tabel 5.1. Data Awal Sebelum <i>Preprocessing</i>	36
Tabel 5.2. Hasil <i>Case Folding</i>	37
Tabel 5.3. Hasil <i>Remove Punctuation</i>	38
Tabel 5.4. Hasil <i>Stopword Removal</i>	40
Tabel 5.5. Hasil <i>Tokenizing</i>	41
Tabel 5.6 Sampel Hasil Pembobotan Kata TF-IDF	42
Tabel 5.7. Sampel Hasil Nilai TF	43
Tabel 5.8. Pembagian Data <i>Training</i> dan Data <i>Testing</i>	44
Tabel 5.9. <i>Confusion Matrix</i> Data <i>Testing Kernel Linear</i>	46
Tabel 5.10. <i>Confusion Matrix</i> Data <i>Testing Kernel RBF</i>	47
Tabel 5.11. <i>Confusion Matrix</i> Data <i>Testing Kernel Sigmoid</i>	48
Tabel 5.12. <i>Confusion Matrix</i> Data <i>Testing Kernel Polynomial</i>	49
Tabel 5.13. <i>Confusion Matrix</i> Data <i>Testing</i> tiap <i>Kernel</i>	50
Tabel 5.14. Hasil Prediksi <i>Class</i> untuk Data Tuntutan Baru.....	50

DAFTAR GAMBAR

Gambar 3.1. Konsep SVM (Sumber : Nugroho, 2003)	24
Gambar 3.2. Konsep Non Linear SVM (Sumber : Nugroho, 2003)	26
Gambar 3.3. Contoh <i>Pattern</i> DOE (Sumber : Staenlin, 2003)	28
Gambar 4.1. Tahapan Analisis Penelitian.....	31
Gambar 4.2. Tahapan Klasifikasi dan Prediksi Menggunakan SVM	32
Gambar 5.1. Jumlah Kasus Tindak Pidana Per Tahun	33
Gambar 5.2. Kasus Tindak Pidana Berdasarkan Publikasi.....	34
Gambar 5.3. Kasus Tindak Pidana Berdasarkan Jenis Acara Pidana.....	34
Gambar 5.4. Jumlah Klasifikasi tiap <i>Class</i>	35
Gambar 5.5. <i>Word Cloud</i> Variabel Tuntutan.....	35
Gambar 5.6. <i>Confusion Matrix</i> SVM Kernel Linear.....	46
Gambar 5.7. <i>Confusion Matrix</i> SVM Kernel RBF	47
Gambar 5.8. <i>Confusion Matrix</i> SVM Kernel Sigmoid.....	48
Gambar 5.9. <i>Confusion Matrix</i> SVM Kernel Polynomial	49

DAFTAR LAMPIRAN

Lampiran 1 Data Penelitian Semua Kasus Pelanggaran Pidana di PN Sleman...	61
Lampiran 2 <i>Script dan Output</i>	61

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Rizky Desi Ramadhani

NIM : 16611129

Tugas akhir dengan judul:

**IDENTIFIKASI KASUS DAN PREDIKSI PUTUSAN HAKIM BERDASARKAN
LAMA PEMIDANAAN DENGAN KLASIFIKASI *SUPPORT VECTOR MACHINE*
(SVM)**

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang sebelumnya pernah diajukan untuk memperoleh gelar sarjana di suatu Perguruan Tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 3 April 2020



Rizky Desi Ramadhani

**IDENTIFIKASI KASUS DAN PREDIKSI
PUTUSAN HAKIM DENGAN KLASIFIKASI
*SUPPORT VECTOR MACHINE (SVM)***

(Studi Kasus : Kasus Tindak Pidana di Pengadilan Negeri Sleman)

Rizky Desi Ramadhani

Program Studi Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Islam Indonesia

INTISARI

Sekarang ini, revolusi industri telah memasuki era revolusi industri 4.0, dimana perkembangan teknologi telah sampai pada tahap disruptif, yakni semua aspek lini mulai dilakukan otomatisasi. Salah satu pekerjaan yang memiliki algoritma tersebut adalah pekerjaan seorang hakim di bidang hukum. Pemanfaatan teknologi dalam hal ini metode SVM *kernel linear* dapat membantu mengklasifikasi hasil putusan suatu perkara pidana di Pengadilan Negeri Sleman, dimana *input* yang digunakan yakni berupa tuntutan di pengadilan negeri tersebut sehingga nantinya akan didapatkan suatu model prediksi yang bermanfaat dalam memprediksi putusan berdasarkan lama pembedaan suatu perkara pidana. Dengan menggunakan metode SVM *kernel linear*, didapatkan nilai akurasi untuk hasil klasifikasi dan prediksi putusan berdasarkan lama pembedaan sebesar 62%.

Kata Kunci: Revolusi Industri, Otomatisasi, NLP, Klasifikasi, Prediksi, SVM

Case Identification and Prediction Judges with Classification

Support Vector Machine (SVM)

(Case Study : Criminal Case in the Sleman District Court)

Rizky Desi Ramadhani

Department Statistics, Faculty of Mathematics and Natural Science

Islamic University of Indonesia

ABSTRACT

Today, the industrial revolution has entered the era of the industrial revolution 4.0, where technological developments have reached the disruptive stage, i.e. all aspects of the line have begun to be automated. One job that has this algorithm is the work of a judge in the field of law. Utilization of technology in this case the linear kernel SVM method can help classify the results of the decision of a criminal case in the Sleman District Court, where the input used is in the form of a claim in the district court so that a predictive model will be useful in predicting a decision based on the length of conviction of a criminal case . By using the linear kernel SVM method, an accuracy value for the results of classification and prediction of decisions based on the length of conviction is 62%.

Keywords: *Industrial Revolution, Automation, NLP, Classification, Prediction, SVM*

BAB 1 PENDAHULUAN

1.1 Latar Belakang Masalah

Revolusi industri merupakan perubahan secara besar-besaran diberbagai bidang serta memiliki dampak yang mendalam terhadap kondisi sosial, ekonomi, dan budaya di dunia. Sekarang ini, revolusi industri telah memasuki era revolusi industri 4.0. Dimana revolusi industri 4.0 secara fundamental mengakibatkan berubahnya cara manusia berpikir, hidup, dan berhubungan satu dengan yang lain (Banu Prasetyo, 2018). Era ini menggantikan seluruh sistem lama dengan cara-cara baru diberbagai aktivitas manusia, tidak hanya dalam bidang teknologi saja akan tetapi juga pada bidang yang lain seperti ekonomi, sosial, dan politik. Sehingga semua aspek lini mulai dilakukan otomatisasi. Otomatisasi pada era ini dikonvergensi dengan teknologi jaringan internet. Hal tersebut merupakan tren otomatisasi dan pertukaran data dalam teknologi manufaktur yang dimanfaatkan untuk mencapai efisiensi sebesar-besarnya sehingga dapat menghasilkan suatu model yang baru berbasis digital. Inovasi teknologi sangat dibutuhkan di era ini, karena menjadi peluang dalam berbisnis untuk menjadi tetap kompetitif di tengah meningkatnya biaya operasional dan pergeseran tuntutan ekonomi. Di Indonesia pun saat ini mulai menggarap konsep Revolusi Industri 4.0 bertajuk *Making Indonesia 4.0* melalui Kementerian Perindustrian.

Sejalan dengan hal tersebut, menurut Pugliano (2019), pekerjaan apa pun yang rutin dilakukan, bisa jadi dilakukan algoritma matematika dalam lima atau sepuluh tahun lagi. Salah satu pekerjaan yang memiliki algoritma tersebut adalah pekerjaan di bidang hukum karena pekerjaan pada bidang profesi ini bersifat rutin dan mengulang, contoh saja profesi hakim.

Menurut UU No 48 Tahun 2009 tentang kekuasaan kehakiman, hakim adalah hakim pada Mahkamah Agung dan hakim pada badan peradilan yang berada di bawahnya. Hakim merupakan salah satu penegak hukum yang memiliki tugas untuk menerima, memeriksa, dan mengadili serta menyelesaikan semua perkara yang diajukan ke Pengadilan.

Dalam proses persidangan suatu perkara pidana di Pengadilan Negeri (PN) Sleman pada tahun 2016-2020, diketahui bahwa rata-rata lama proses yang dibutuhkan seorang hakim untuk membuat putusan dari suatu perkara adalah 50 hari. Pemanfaatan teknologi dalam hal ini, dapat sangat membantu meringankan pekerjaan hakim untuk memutuskan sebuah perkara. Salah satu pemecahan masalah yang saat ini banyak dikembangkan adalah dengan menggunakan teknologi *artificial intelligence*. Salah satu kemampuan *artificial intelligence* terkait dengan pengolahan bahasa manusia khususnya pada tuntutan pengadilan yang digunakan dalam penelitian ini adalah metode *natural language processing* (NLP). NLP merupakan metode yang mengkaji interaksi antar manusia dengan komputer menggunakan bahasa manusia. Harapan dari penelitian ini adalah dapat membantu hakim dalam menentukan putusan sehingga penanganan perkara dapat diselesaikan dengan lebih cepat dan lebih efisien. Otomatisasi tersebut, semata-mata untuk membantu kinerja para hakim dalam identifikasi dan ekstraksi pola yang mengarah pada putusan suatu perkara.

Berdasarkan hasil uraian diatas, peneliti akan melakukan penelitian dengan judul : “Identifikasi Kasus dan Prediksi Putusan Hakim dengan Klasifikasi *Support Vector Machine* (SVM)”

1.2 Rumusan Masalah

Pada penelitian ini, rumusan masalah yang akan diangkat oleh peneliti adalah sebagai berikut :

1. Bagaimana gambaran umum mengenai kasus dan hasil putusan tindak pidana yang terdapat di PN Sleman?
2. Bagaimana klasifikasi putusan berdasarkan lama pembedaan dengan menggunakan metode *Support Vector Machine*?
3. Bagaimana prediksi putusan berdasarkan lama pembedaan dalam menjatuhkan putusan pidana pada kasus-kasus yang terdapat di PN Sleman?

1.3 Batasan Masalah

Permasalahan yang akan di bahas pada penelitian yang memiliki ruang lingkup yang luas, sehingga perlu adanya batasan masalah sebagai berikut :

1. Data pada penelitian ini yakni data sekunder berupa data rekapan perkara pidana di PN Sleman pada tahun 2016-2020.
2. *Software* yang digunakan peneliti yaitu *Python*.
3. Dalam *preprocessing data*, data angka tidak dihapuskan.
4. Dalam klasifikasi data, penelitian ini hanya menggunakan 2 *class* atau kategori saja yang didasarkan dari justifikasi peneliti setelah dikonsultasikan dengan ahli hukum.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian yang di lakukan sebagai berikut :

1. Mengetahui gambaran umum mengenai kasus tindak pidana yang terdapat di PN Sleman.
2. Mengetahui hasil klasifikasi putusan berdasarkan lama pemedanaan dengan menggunakan metode *Support Vector Machine*.
3. Mengetahui hasil prediksi dalam menjatuhkan putusan berdasarkan lama pemedanaan pada kasus-kasus yang terdapat di PN Sleman.

1.5 Manfaat Penelitian

Adapun manfaat penelitian adalah sebagai berikut:

1. Membantu serta memudahkan pihak hakim dalam mengidentifikasi kasus dan mengekstraksi pola yang mengarah pada prediksi putusan tertentu.
2. Mengurangi disparitas hakim dalam menentukan putusan perkara yang sama.
3. Memberikan pengetahuan baru kepada peneliti terhadap *artificial intelligence* khususnya pada metode *natural language processing* serta dalam penggunaan *Python*.

BAB 2 TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu merupakan salah satu hal penting guna mengetahui keterkaitan penelitian saat ini dengan penelitian sebelumnya. Setelah melakukan beberapa tinjauan pustaka terhadap penelitian terdahulu, terdapat beberapa penelitian yang memiliki keterkaitan yang sama baik dalam hal metode penelitian, objek penelitian, hingga subjek penelitian. Berikut merupakan ulasan mengenai penelitian-penelitian terdahulu yang dijadikan acuan peneliti.

Penelitian yang dilakukan oleh Simanjuntak tahun 2010 mengenai disparitas pidana terhadap pelaku tindak pidana pembunuhan di Wilayah Hukum Pengadilan Negeri Sleman. Tujuan penelitian ini adalah untuk mengetahui apakah disparitas pidana oleh hakim terhadap pelaku tindak pidana sudah berpedoman pada asas indivisualisasi pidana. Data yang digunakan adalah bahan hukum primer, sekunder, dan tersier. Proses penalaran dalam hal menarik kesimpulan, digunakan metode berfikir deduktif yakni melalui proses deduksi dari norma hukum positif yang berlaku, yakni peraturan perundang-undangan tentang hukum pidana dan hukum acara pidana serta peraturan perundang-undangan tentang kekuasaan pokok kehakiman.

Penelitian yang dilakukan oleh Aletras, Tsarapatsanis, Preoțiu-Pietro, dan Lampo tahun 2016 mengenai prediksi keputusan yudisial pada Pengadilan HAM Eropa dengan menggunakan perspektif *natural language processing*. Tujuan penelitian ini adalah untuk membangun sebuah model prediksi yang digunakan untuk mengidentifikasi suatu kasus di Pengadilan HAM Eropa dan mengekstraksi pola yang mengarah pada keputusan tertentu. Data yang digunakan terdiri dari kasus-kasus yang terkait dengan pasal 3, 6, dan 8 konvensi. Dengan menggunakan klasifikasi SVM *kernel linear* didapatkan akurasi model untuk memprediksi keputusan pengadilan sebesar 79%.

Penelitian yang dilakukan oleh Sulea, Zampieri, Vela, Genabith tahun 2017 mengenai prediksi area hukum dan putusan kasus Mahkamah Agung Perancis. Tujuan penelitian ini adalah menerapkan metode klasifikasi teks untuk

memprediksi area hukum dan keputusan kasus yang dihakimi oleh Mahkamah Agung Perancis. Data yang digunakan yakni koleksi putusan diakronis dari Pengadilan Tinggi Prancis. Dengan menggunakan SVM *kernel* linear didapatkan hasil skor 96% f1 dalam memprediksi putusan perkara, skor 90% f1 dalam memprediksi bidang hukum suatu kasus, dan skor 75,9% f1 dalam memperkirakan waktu ketika putusan dikeluarkan.

Penelitian yang dilakukan oleh Katz, Bommarito, Blackman tahun 2017 mengenai pendekatan umum untuk memprediksi perilaku Mahkamah Agung Amerika Serikat. Tujuan penelitian ini adalah untuk membangun model yang dirancang untuk memprediksi perilaku Mahkamah Agung Amerika Serikat dalam konteks umum. Data yang digunakan berasal dari *database* Mahkamah Agung Amerika Serikat. Dengan menggunakan *random forest*, didapatkan nilai akurasi sebesar 70,2% pada tingkat hasil kasus dan 71,9% pada tingkat suara keadilan.

Penelitian yang dilakukan oleh Bhilare, Parab, Soni, dan Tahkur pada tahun 2019 mengenai prediksi hasil perkara pengadilan dan analisis menggunakan *machine learning*. Tujuan penelitian ini adalah untuk memprediksi hasil perkara hukum dengan ketepatan yang maksimal. Data yang digunakan yakni deskripsi kasus dengan penilaian dalam korespondensi kolom. Dengan menggunakan beberapa metode, didapatkan metode yang memiliki tingkat keakuratan paling tinggi yakni metode SVM *kernel linear* dengan akurasi sebesar 78%.

Penelitian yang dilakukan oleh Gruginskie dan Vaccaro tahun 2019 mengenai prediksi waktu perkara gugatan. Tujuan penelitian ini adalah untuk menyesuaikan dan membandingkan model untuk prediksi panjang kasus keseluruhan dalam sistem Pengadilan di Brazil dengan fokus pada gugatan perdata di Pengadilan Federal. Data diperoleh TRF4 dengan rentang waktu satu tahun. Pada penelitian ini digunakan empat pendekatan utama yang digunakan untuk klasifikasi *lead time* yakni SVM, *naïve bayes* (NB), *random forest* (RF), dan *neural network* (NN). Hasil dari penelitian ini adalah pendekatan SVM dipilih karena keakuratan untuk memodelkan batas keputusan *non linier* yang kompleks, dan mengurangi kecenderungan *overfitting* lebih baik dibanding dengan metode lain dan memiliki nilai akurasi tertinggi sebesar 77,58%.

Penelitian yang dilakukan oleh Medvedeva, Vols, dan Wieling tahun 2019 mengenai penggunaan *machine learning* untuk memprediksi keputusan Pengadilan HAM Eropa. Tujuan penelitian ini adalah untuk menganalisis teks dari proses pengadilan untuk secara otomatis memprediksi keputusan pengadilan di masa depan. Data yang digunakan mengenai kasus-kasus pelanggaran yang berasal dari Pengadilan HAM Eropa secara terpisah. Dengan menggunakan metode SVM, didapatkan nilai akurasi sebesar 75%.

Penelitian yang dilakukan oleh Kusumawardani tahun 2019 mengenai hukum progresif dan perkembangan teknologi kecerdasan buatan. Tujuan penelitian ini adalah untuk melihat bagaimana hubungan hukum progresif dan perkembangan teknologi kecerdasan buatan pada sekarang ini. Pada dasarnya, sistem hukum yang ada dibangun di sekitar kemampuan manusia dan bertujuan untuk melindungi manusia dari penderitaan. Dengan munculnya teknologi kecerdasan buatan, maka hukum juga akan terpengaruh secara teknis dan teknologis. Dalam hukum progresif, teknologi harus dimaknai tidak semata-mata sebagai teknologi, melainkan teknologi yang dihasilkan harus mampu mengekspresikan nilai dan moral di dalamnya. Selain itu, pada akhirnya jika mendasarkan pada hukum progresif maka hukum yang diciptakan terkait teknologi kecerdasan buatan harus berbasis pada manusia dan kemanusiaan, yaitu mampu menolong manusia yang susah serta menderita, yang bertujuan mewujudkan keadilan yang membahagiakan bagi masyarakat.

Dari beberapa tinjauan pustaka yang dijadikan sebagai acuan dalam penelitian ini, **Tabel 2.1** merupakan tabel rangkuman perbandingan antara penelitian saat ini dan penelitian terdahulu.

Tabel 2.1. Perbandingan dengan Penelitian Sebelumnya

No	Penulis	Judul Penelitian	Metode	Persamaan	Perbedaan
1.	Marwan P Simanjutak (2010)	Disparitas Pidana Terhadap Pelaku Tindak Pidana Pembunuhan Di	Analisis Kualitatif	Sama-sama membahas studi kasus mengenai tindak	Metode yang digunakan sekarang adalah

No	Penulis	Judul Penelitian	Metode	Persamaan	Perbedaan
		Wilayah Hukum Pengadilan Negeri Sleman		pidana	metode NLP
2.	Nikolaos Aletras, Dimitrios Tsarapatsani, Daniel Preoțiuc-Pietro, dan Vasileios Lampo (2016)	<i>Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective</i>	<i>Natural Language Processing (NLP)</i>	Sama-sama menggunakan metode NLP dan SVM	Studi kasus yang digunakan sekarang adalah mengenai kasus tindak pidana Pengadilan Negeri
3.	Octavia-Maria Sulea, Marcos Zampieri, Mihaela Vela, Josef Van Genabith (2017)	<i>Predicting the Law Area and Decisions of French Supreme Court Cases</i>	<i>Natural Language Processing</i>	Sama-sama menggunakan metode NLP dan SVM	Studi kasus yang digunakan sekarang adalah mengenai kasus pelanggaran pidana Pengadilan Negeri
4.	Daniel Martin Katz, Michael J. Bommarito, Josh	<i>A General Approach for Predicting the Behavior of the</i>	<i>Natural Language Processing</i>	Sama-sama menggunakan metode NLP	Studi kasus yang digunakan sekarang

No	Penulis	Judul Penelitian	Metode	Persamaan	Perbedaan
	Blackman (2017)	<i>Supreme Court of the United State</i>			adalah mengenai kasus tindak pidana Pengadilan Negeri
5.	Prof. Priyanka Bhilare, Neha Parab, Namrata Soni, dan Bhakti Tahkur (2019)	<i>Predicting Outcome of Judicial Cases and Analysis using Machine Learning</i>	<i>Natural Language Processing</i>	Sama-sama menggunakan metode NLP dan SVM	Studi kasus yang digunakan sekarang adalah mengenai kasus tindak pidana Pengadilan Negeri
6.	Lúcia Adriana dos Santos Gruginskie, Guilherme Luís Roehé Vaccaro (2019)	<i>Lawsuit lead time prediction : Comparisan of Data Mining Techniques Based on Categorical Response Variable</i>	<i>Natural Language Processing</i>	Sama-sama menggunakan metode NLP dan SVM	Studi kasus yang digunakan sekarang adalah mengenai kasus tindak pidana Pengadilan Negeri
7.	Masha Medvedeva,	<i>Using Machine Learning to</i>	<i>Natural Language</i>	Sama-sama menggunakan	Studi kasus yang

No	Penulis	Judul Penelitian	Metode	Persamaan	Perbedaan
	Michel Vols, Martijn Wieling (2019)	<i>Predict Decisions of The European Court of Human Rights</i>	<i>Processing</i>	an metode NLP dan SVM	digunakan sekarang adalah mengenai kasus pelanggaran pidana Pengadilan Negeri
8.	Qur'ani Dewi Kusumawarda ni (2019)	Hukum Progresif dan Perkembangan Teknologi Kecerdasan Buatan	Analisis Kualitatif	Sama-sama membahas studi kasu mengenai hukum dan kecerdasan buatan	Metode yang digunakan sekarang adalah metode NLP

Penelitian yang dilakukan oleh Ramadhani tahun 2020 mengenai prediksi putusan hakim berdasarkan lama pembedaan pada setiap kasus pelanggaran pidana di Pengadilan Negeri Sleman. Tujuan dari penelitian ini adalah memprediksi putusan hakim berdasarkan lama pembedaan. Data yang digunakan adalah data tuntutan dan putusan berdasarkan lama pembedaan yang berasal dari Pengadilan Negeri Sleman.

BAB 3 LANDASAN TEORI

3.1 Revolusi Industri 4.0

Revolusi Industri merupakan keadaan dimana banyak aspek kehidupan yang terpengaruh oleh perubahan global. Proses produksi atau jasa yang awalnya memakan waktu lama, tenaga berat, biaya mahal menjadi lebih mudah, lebih cepat, dan lebih murah dalam prosesnya. Dengan adanya konsep revolusi industri, risiko kelangkaan tersebut dapat diturunkan atau bahkan dihilangkan. Sehingga tenaga, waktu, dan biaya yang dibutuhkan sebelumnya cukup besar dapat menjadi lebih ringan bahkan tidak ada dan dialihkan ke hal lain (Baenanda, 2019).

Angka empat pada istilah Industri 4.0 merujuk pada revolusi yang ke empat. Industri 4.0 merupakan fenomena yang unik jika dibandingkan dengan tiga revolusi industri yang mendahuluinya. Industri 4.0 adalah kombinasi dari sebuah sistem cerdas dan otomasi dalam industri. Pada era ini, pelaku industri menghubungkan komputer agar dapat berkomunikasi satu sama lain yang pada akhirnya dapat membuat keputusan tanpa keterlibatan manusia (Laksana, 2019).

3.2 Hukum

Menurut (Utrecht, 1966), hukum adalah himpunan petunjuk hidup berisi tentang perintah dan larangan, yang mengatur tata tertib kehidupan bermasyarakat dan harus ditaati oleh setiap individu masyarakat dan jika dilanggar dapat menimbulkan tindakan dari pihak pemerintah dari masyarakat itu.

3.3 Hukum Pidana

Menurut (Moeljanto, 1993), hukum pidana adalah bagian dari keseluruhan hukum yang berlaku di suatu negara yang mengadakan dasar-dasar dan mengatur ketentuan tentang perbuatan yang tidak boleh dilakukan, dilarang yang disertai ancaman pidana bagi barangsiapa yang melakukan. Menentukan kapan dan dalam hal apa kepada mereka yang melanggar larangan-larangan itu dapat dikenakan sanksi pidana sebagaimana yang telah diancamkan. Dan menentukan dengan cara bagaimana pengenaan pidana dapat dilaksanakan apabila orang telah melanggar larangan tersebut.

Intinya, hukum pidana mengatur tindakan apa yang tidak boleh dilakukan yang sudah ada pada ketentuan, jika tindakan tersebut dilakukan maka akan ada sanksi bagi orang yang melakukannya.

3.3.1 Tindak Pidana

Menurut Simons dalam buku *Asas-Asas Hukum Pidana*, tindak pidana adalah kelakuan yang diancam dengan pidana yang bersifat melawan hukum yang berhubungan dengan kesalahan serta dilakukan oleh orang yang mampu bertanggungjawab (Moeljanto, 1993).

3.3.2 Jenis-Jenis Pidana

Pada pasal 10 Kitab Undang-Undang Hukum Pidana (KUHP) di Indonesia menjelaskan bahwa pidana dibedakan menjadi dua.

1. Pidana pokok terdiri atas
 - a. pidana mati yaitu sanksi terberat, tertua, dan sering dikatakan paling kejam diantara semua jenis pidana yang ada. Penjatuhan pidana mati diancamkan dalam beberapa pasal tertentu didalam KUHP.
 - b. pidana penjara yaitu pidana yang berwujud perampasan kemerdekaan seseorang yang bertujuan untuk membina terdakwa agar dapat kembali menjadi anggota masyarakat yang baik dan berguna. Pidana penjara sendiri dibedakan menjadi dua yakni pidana penjara seumur hidup dan pidana sementara. Dimana pidana penjara sementara minimal satu tahun dan maksimal 20 tahun (Asikin, 2012).
 - c. pidana kurungan yaitu pidana yang sama seperti pidana penjara akan tetapi pidana ini lebih ringan dibandingkan pidana penjara. Dalam pasal 18 KUHP, diatur pidana kurungan minimal satu hari dan maksimal satu tahun.
 - d. pidana denda yaitu pidana berupa harta benda yang jumlah ancaman pidananya relatif ringan dan merupakan pokok pidana alternatif dari pidana kurungan dan pidana penjara. Dalam pasal 30 ayat 4 KUHP, diatur perhitungan untuk pengganti denda yakni putusan denda setengah rupiah atau kurang lamanya ditetapkan satu hari dan putusan denda yang lebih

dari setengah rupiah ditetapkan kurungan bagi tiap-tiap setengah rupiah dan kelebihanya tidak lebih dari suatu hari lamanya.

- e. pidana tutupan yaitu pidana yang baru dalam KUHP. Namun hingga saat ini, hanya pernah terjadi satu kali hakim yang menjatuhkan pidana tutupan yakni putusan putusan Mahkamah Tentara Agung RI pada tanggal 27 Mei 1948 dalam hal mengadili para pelaku kejahatan yang dikenal dengan sebutan peristiwa 3 Juli 1946.

2. Pidana tambahan terdiri atas

- a. pencabutan beberapa hak tertentu memiliki sifat yang sementara, kecuali jika terdakwa dijatuhi pidana penjara seumur hidup. Hukuman ini dasarnya ditujukan sebagai upaya mendegradasikan martabat seseorang. Dalam pasal 35 ayat 1 KUHP, hak-hak yang dicabut oleh hakim yakni hak menjabat segala jabatan, hak untuk masuk balatentara, hak memilih dan dipilih dalam pemilihan, hak untuk menjadi penasihat, kuasa wali atas anak sendiri, dan hak melakukan pekerjaan yang ditentukan.
- b. perampasan barang yang tertentu yaitu pidana yang dijatuhkan hakim untuk mencabut hak milik atas suatu barang dari pemiliknya.
- c. pengumuman putusan hakim yaitu digunakan sebagai usaha mencegah bagi orang-orang tertentu agar tidak melakukan tindak pidana yang sering dilakukan (R.Soesilo, 1991).

3.3.3 Acara Pemeriksaan

Terdapat tiga jenis acara pemeriksaan di sidang pengadilan (Badan Diklat Kejaksaan RI, 2019), yakni:

1. Acara Pemeriksaan Biasa

Menurut pasal 152-202 KUHP, acara pemeriksaan biasa adalah tindak pidana yang pembuktiannya mudah, penerapan hukumnya tidak mudah serta sifat melawan hukumnya tidak sederhana. Bentuk putusan dibuat tersendiri menurut ketentuan serta diucapkan dengan hadirnya terdakwa yakni dapat berupa putusan bebasm putusan lepas dari segala tuntutan hukum, putusan tuntutan tidak diterima, atau putusan pemidanaan.

2. Acara Pemeriksaan Cepat

Menurut pasal 205 ayat 1 KUHP, acara pemeriksaan cepat adalah tindak pidana yang diancamkan hukuman tidak lebih dari 3 bulan. Bentuk putusan terdiri dari keputusan berupa pidana denda dan atau keputusan berupa perampasan kemerdekaan.

3. Acara Pemeriksaan Singkat

Menurut pasal 203 ayat 1 KUHP, acara pemeriksaan singkat adalah tindak pidana yang pembuktian dan penerapan hukumnya mudah serta sifat melawan hukumnya sederhana. Bentuk putusan tidak dibuat secara khusus hanya dicatat dalam berita acara sidang serta diucapkan dengan hadirnya terdakwa.

3.3.4 Pertanggungjawaban Pidana

Pertanggungjawaban pidana adalah bentuk untuk menentukan apakah seorang terdakwa tersebut dipidana atau dibebaskan. Dalam pertanggungjawaban pidana beban pertanggungjawaban dibebankan kepada pelaku yang melanggar tindak pidana yang berkaitan dengan dasar dalam menjatuhkan sanksi pidana. Menurut (Huda, 2006) mengatakan bahwa seseorang memiliki pertanggungjawaban pidana apabila pelaku sudah melakukan perbuatan yang salah dan bertentangan dengan hukum. Hakikatnya, pertanggungjawaban ini adalah mekanisme yang diciptakan untuk reaksi atas pelanggaran atau kejahatan dari suatu perbuatan tertentu yang telah disepakati.

3.4 Pengadilan Negeri (PN)

Pengadilan Negeri adalah pengadilan yang sehari-hari memeriksa dan memutuskan perkara baik pidana maupun perdata yang merupakan pengadilan tingkat pertama dan berkedudukan di ibukota daerah kabupaten/kota dengan daerah hukum juga meliputi wilayah kabupaten/kota (Mahkamah Agung, 2016).

3.5 Prosedur Persidangan Pengadilan

Secara singkat, prosedur persidangan pidana yang telah diatur Undang-Undang No.8 Tahun 1981 Tentang Hukum Acara Pidana adalah sebagai berikut.

1. Sidang dinyatakan dibuka dan terbuka untuk umum (kecuali perkara tertentu dinyatakan tertutup untuk umum).
2. Jaksa Penuntut Umum (JPU) diperintahkan untuk menghadapkan terdakwa ke depan persidangan dalam keadaan bebas.
3. Terdakwa ditanyakan identitasnya dan ditanya apakah sudah menerima salinan surat tuntutan.
4. Terdakwa ditanya pula apakah dalam keadaan sehat dan bersedia untuk diperiksa di depan persidangan (kalau bersedia sidang dilanjutkan).
5. Terdakwa ditanyakan apakah akan didampingi oleh Penasihat Hukum (apabila didampingi apakah akan membawa sendiri, kalau tidak membawa sendiri akan ditunjuk Penasihat Hukum (PH) oleh Majelis Hakim dalam hal terdakwa diancam dengan pidana penjara lima tahun atau lebih/pasal 56 KUHAP ayat (1).
6. Dilanjutkan pembacaan surat tuntutan.
7. Atas pembacaan surat tuntutan tadi terdakwa ditanya akan mengajukan eksepsi atau tidak.
8. Dalam hal terdakwa/PH mengajukan eksepsi maka diberi kesempatan dan sidang ditunda.
9. Apabila ada eksepsi dilanjutkan tanggapan JPU atas eksepsi (replik);
10. Selanjutnya dibacakan putusan sela oleh Majelis Hakim.
11. Apabila eksepsi ditolak dilanjutkan pemeriksaan pokok perkara (pembuktian)
12. Pemeriksaan saksi-saksi yang diajukan oleh JPU (dimulai dari saksi korban).
13. Dilanjutkan saksi lainnya.
14. Apabila ada saksi yang meringankan diperiksa pula, saksi ahli *expert witness*
15. Pemeriksaan terhadap terdakwa.
16. Surat tuntutan pidana (*requisitor*) oleh penuntut umum;
17. Pembelaan (pledoi) oleh Penasehat hukum.
18. Replik atau tanggapan penuntut umum atas nota pembelaan penasihat hukum terdakwa.
19. Duplik atau tanggapan penasehat hukum terdakwa atas tanggapan penuntut umum.

20. Putusan oleh Majelis Hakim.

3.6 Hakim

Menurut UU No 48 Pasal 1 tentang kekuasaan kehakiman, hakim adalah hakim pada Mahkamah Agung dan hakim pada badan peradilan yang berada di bawahnya dalam lingkungan peradilan umum, lingkungan peradilan agama, lingkungan peradilan militer, lingkungan peradilan tata usaha negara, dan hakim pada pengadilan khusus yang berada dalam lingkungan tersebut. Pada pasal 1 butir 8 KUHP Hakim merupakan pejabat pengadilan negara yang memiliki wewenang oleh undang-undang untuk mengadili. Mengadili dalam hal ini adalah menerima, memeriksa, dan memutuskan perkara pidana berdasarkan asas bebas, jujur, dan tidak memihak pada sidang pengadilan dalam hal dan menurut cara yang diatur oleh undang-undang.

3.7 Putusan Pengadilan

Pada pasal 1 angka 11 KUHP, putusan pengadilan adalah suatu pernyataan dari seorang hakim dalam sidang pengadilan terbuka, pernyataan tersebut dapat berupa pemidanaan atau bebas atau lepas dari segala tuntutan hukum menurut cara-cara yang telah diatur oleh undang-undang..

Dalam mempertimbangkan putusan, hakim dapat dengan jelas menjelaskan mengapa seseorang yang menjadi terdakwa dijatuhi pidana kurungan atau penjara sekian tahun ataupun denda sekian rupiah, bahkan hingga pidana mati atau pidana seumur hidup. Tolak ukur dalam mempertimbangkan putusan tersebut salah satunya adalah pertimbangan keadaan memberatkan dan keadaan meringankan (Hananta, 2018).

3.8 Analisis Deskriptif

Menurut (Sugiyono, 2009), metode deskriptif adalah penelitian yang dilakukan untuk mengetahui keberadaan variabel mandiri, baik hanya pada satu variabel atau lebih (variabel yang berdiri sendiri) tanpa membuat perbandingan dan mencari hubungan variabel itu dengan variabel yang lain.

3.9 Kecerdasan Buatan (*Artificial Intelligence*)

Artificial intelligence adalah perangkat komputer yang dapat memahami lingkungannya dan dapat mengambil tindakan yang memaksimalkan peluang kesuksesan di lingkungan tersebut untuk beberapa tujuan (Russell, 2015). Sedangkan tujuan dari *Artificial intelligence* menurut McCarthy (1956) dalam (Dahria, 2008) adalah untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Cerdas, berarti memiliki pengetahuan ditambah pengalaman, penalaran (bagaimana membuat keputusan dan mengambil tindakan), dan moral yang baik. Dengan kata lain, *Artificial Intelligence* dapat membantu manusia dalam proses *problem solving*.

Artificial Intelligence dirancang menggunakan banyak algoritma. Algoritma tersebut membantu sistem dalam *Artificial Intelligence* menentukan respon yang diharapkan. Berikut adalah ruang lingkup dalam *Artificial Intelligence* yang digunakan kehidupan sehari-hari:

1. Sistem Pakar (*Expert System*)

Sistem pakar merupakan suatu sistem komputer yang menyamai atau meniru keahlian dalam menyelesaikan masalah dari seorang pakar.

2. Pengolahan Bahasa Alami (*Natural Language Processing*)

Kemampuan komputer dalam memahami bahasa manusia.

3. Pengenalan Ucapan (*Speech Recognition*)

Kemampuan manusia dalam berkomunikasi dengan komputer melalui suara.

4. Sistem Sensor dan Robotika

Sistem Sensor adalah suatu yang digunakan dalam mendeteksi adanya perubahan fisik atau kimia.

Robotika adalah salah satu cabang teknologi yang memadukan berbagai disiplin ilmu antara mekanik, elektronik, dan komputer.

5. *Computer Vision*

Kemampuan komputer dalam mengenali obyek yang diamati.

6. *Intelligent Computer Aided Instruction*

Kemampuan komputer dalam mengajar manusia dengan membentuk teknik pengajaran sesuai dengan pola.

7. *Game Playing*

Permainan anantara manusia melawan mesin yang memiliki intelektual untuk berpikir.

Pada penelitian ini, *artificial intelligence* yang digunakan adalah pengolahan bahasa manusia yakni kemampuan komputer dalam memahami bahasa manusia. Bahasa manusia yang dimaksud sendiri adalah tuntutan seorang hakim dalam memberikan hasil putusan berdasarkan lama pembedanaan di suatu perkara pidana.

3.10 Pembelajaran Mesin (*Machine Learning*)

Menurut Tanaka dan Okutami (2014) dalam (Ahmad, 2017) *Machine Learning* atau pembelajaran mesin merupakan pendekatan dalam *Artificial Intelligence* yang banyak digunakan untuk menggantikan atau menirukan perilaku manusia untuk menyelesaikan masalah atau melakukan otomatisasi. Sesuai namanya, ML mencoba menirukan bagaimana proses manusia atau makhluk cerdas belajar dan menggeneralisasi. Banyak hal yang dipelajari dari *machine learning*, tetapi pada dasarnya terdapat 4 hal pokok yang dipelajari yakni:

1. *Supervised Learning*

Tipe *machine learning* yang mempelajari fungsi pemetaan *input* ke *output*. Hasilnya adalah untuk memperkirakan fungsi dari pemetaanya. Sehingga memiliki *input* baru dan dapat melakukan prediksi *output* dari *input* tersebut. Contohnya *Support Vector Machine*, *Naïve Bayes Classifier*, dan lain sebagainya.

2. *Unsupervised Learning*

Tipe *machine learning* yang memiliki data *input* akan tetapi tidak memiliki variabel *output* yang berhubungan. Hasilnya adalah untuk memodelkan struktur dasar atau distribusi data yang bertujuan untuk mempelajari data lebih jauh lagi. Contohnya *K-Means*, *Self-Organizing Maps*, dan lain sebagainya.

3. *Semi-supervised Learning*

Tipe *machine learning* yang memiliki data *input* dalam jumlah besar dan beberapa dari data tersebut yang dilabeli.

4. *Reinforcement Learning*

Tipe *machine learning* yang dapat menentukan tingkah laku ideal terhadap konteks yang spesifik secara otomatis yang bertujuan untuk memaksimalkan performanya. Contohnya *Q-Learning*, *SARSA*, *ASC*, dan lain sebagainya.

3.11 *Natural Language Processing*

Natural Language Processing (NLP) merupakan salah satu cabang ilmu AI yang berfokus pada pengolahan bahasa natural. Bahasa natural adalah bahasa yang secara umum digunakan oleh manusia dalam berkomunikasi satu sama lain. Bahasa yang diterima oleh komputer butuh untuk diproses dan dipahami terlebih dahulu supaya maksud dari *user* bisa dipahami dengan baik oleh komputer (Suhartono, 2013).

Menurut (D.L. Poole, 2010), terdapat 3 aspek utama pada teori pemahaman mengenai *natural language* :

1. *Syntax*

Menjelaskan bentuk dari bahasa. *Syntax* biasa dispesifikasikan oleh sebuah *grammar*. *Natural language* jauh lebih daripada *formal language* yang digunakan untuk logika kecerdasan buatan dan program komputer.

2. *Semantics*

Menjelaskan arti dari kalimat dalam satu bahasa. Meskipun teori *semantics* secara umum sudah ada, ketika membangun sistem *natural language understanding* untuk aplikasi tertentu, akan digunakan representasi yang paling sederhana.

3. *Pragmatics*

Menjelaskan bagaimana pernyataan yang ada berhubungan dengan dunia. Untuk memahami bahasa, agen harus mempertimbangan lebih dari hanya sekedar kalimat. Agen harus melihat lebih ke dalam konteks kalimat, keadaan dunia, tujuan dari *speaker* dan *listener*, konvensi khusus, dan sejenisnya.

Selain teori pemahaman diatas, terdapat bidang pengetahuan lain yang berhubungan dengan *natural language* yakni:

1. Fonetik dan Fonologi

Pengetahuan yang digunakan dalam mendeteksi suara untuk menjadi sebuah kata yang dikenali.

2. Morfologi

Pengetahuan mengenai pembentukan kata dari kata dasar

3. *Discourse Knowledge*

Menjelaskan pengenalan suatu kata yang terbaca sebelumnya yang kemudian akan berpengaruh pada arti kata berikutnya.

4. *World Knowledge*

Menjelaskan arti khusus dari suatu kata dalam suatu percakapan dengan konteks tertentu.

Pada *field natural language processing*, dijelaskan beberapa area utama penelitian oleh (J. Pustejovsky, 2012) diantaranya :

1. *Question Answering Systems (QAS)*

Kemampuan komputer untuk menjawab pertanyaan yang diberikan oleh *user*. Daripada memasukkan *keyword* ke dalam *browser* pencarian, dengan QAS, *user* bisa langsung bertanya dalam bahasa natural yang digunakannya, baik itu Inggris, Mandarin, ataupun Indonesia.

2. *Summarization*

Pembuatan ringkasan dari sekumpulan konten dokumen atau email. Dengan menggunakan aplikasi ini, *user* bisa dibantu untuk mengkonversikan dokumen teks yang besar ke dalam bentuk *slide* presentasi.

3. *Machine Translation*

Produk yang dihasilkan adalah aplikasi yang dapat memahami bahasa manusia dan menterjemahkannya ke dalam bahasa lain. Termasuk di dalamnya adalah *google translate* yang apabila dicermati semakin membaik dalam penterjemahan bahasa.

4. *Speech Recognition*

Field ini merupakan cabang ilmu *natural language processing* yang cukup sulit. Proses pembangunan model untuk digunakan telpon/komputer dalam mengenali bahasa yang diucapkan sudah banyak dikerjakan. Bahasa yang sering digunakan adalah berupa pertanyaan dan perintah.

5. *Document Classification*

Sedangkan aplikasi ini adalah merupakan area penelitian *natural language processing* yang paling sukses. Pekerjaan yang dilakukan aplikasi ini adalah menentukan dimana tempat terbaik dokumen yang baru dimasukan ke dalam sistem. Hal ini sangat berguna pada aplikasi *spam filtering*, *news article classification*, dan *movie review*.

3.12 Text Mining

Menurut (R. Feldman, 2007) dalam bukunya yang berjudul *The Text Mining Handbook*, *text mining* adalah suatu proses yang menggali informasi dalam koleksi teks yang besar dan secara otomatis mengidentifikasi pola dan hubungan yang menarik dalam data tekstual . Dalam menyelesaikan suatu masalah, *text mining* mengembangkan banyak teknik dari bidang lain, seperti *Data mining*, *Information Retrieval*, Statistik dan Matematik, *Machine Learning*, *Linguistic*, *Natural Language Processing*, dan *Visualization*. Kegiatan penelitian dalam *text mining* diantaranya ekstraksi dan penyimpanan teks, *text preprocessing*, pengumpulan data statistik dan indexing serta analisa konten.

Sedangkan tujuan dari *text mining* adalah mendapatkan suatu informasi berguna dari sekumpulan dokumen pada koleksi dokumen yang besar. Data yang digunakan dalam *text mining* adalah kumpulan teks yang memiliki format *unstructured data*, atau minimal *semistructured*.

Proses *text mining* dibagi menjadi 3 tahap utama , yaitu *text preprocessing*, *text transformation* atau *feature generation*, *pattern discovery* (Even, 2002)

3.12.1 Text Processing

Tahap ini adalah tahapan awal pada *Text Mining*. *Text Preprocessing* dilakukan untuk menghilangkan bagian atau teks yang tidak diperlukan sehingga

mendapatkan data yang berkualitas untuk dieksekusi atau agar proses mining lebih akurat (S. Sanjaya, 2015). Pada penelitian ini, *text preprocessing* yang digunakan yakni:

1. *Case folding*

Mengubah semua huruf dalam dokumen menjadi huruf kecil.

2. *Remove Punctuation*

Menghapus tanda baca dalam dokumen

3. *Tokenization*

Mengubah sekumpulan teks menjadi kata-kata

3.12.2 Text Transformation

Text transformation atau pembentukan atribut mengacu pada proses untuk mendapatkan representasi dokumen yang diharapkan. Pendekatan representasi dokumen yang lazim digunakan oleh model “*bag of words*” dan model ruang vektor (*vector space model*). Transformasi teks sekaligus juga melakukan pengubahan kata-kata ke bentuk dasarnya dan pengurangan dimensi kata di dalam dokumen (Rusydia, 2016). Didalam *feature generation*, terdapat *feature selection* yaitu pemilihan fitur atau tahap lanjutan dari pengurangan dimensi pada proses *text transformation*. *Feature selection* yang digunakan pada penelitian ini yakni:

1. *Stopwords Removal*

Menghilangkan kata-kata yang bukan merupakan ciri (kata unik) dari suatu dokumen.

2. *Stemming dan Lemmatization*

Mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akar atau dasarnya

3.13 Pembobotan Term Frequency-Inverse Document Frequency

Metode *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah cara pemberian bobot dalam hubungan suatu kata (*term*) terhadap dokumen. Selain itu juga digunakan sebagai metode pembandingan terhadap metode pembobotan baru. Karena pada setiap kata memiliki tingkat kepentingan berbeda,

maka setiap kata tersebut diberikan sebuah indikator yang diberi nama *term weight*. Menurut Mandala dalam (Fattah, 2016) pembobotan kata dipengaruhi oleh hal-hal berikut ini:

1. *Term Frequency* (TF) adalah faktor yang menentukan bobot kata pada suatu dokumen berdasarkan jumlah kemunculannya dalam dokumen tersebut. Nilai jumlah kemunculan suatu kata (*term frequency*) diperhitungkan dalam pemberian bobot terhadap suatu kata. Semakin besar jumlah kemunculan suatu kata dalam dokumen, semakin besar pula bobotnya dalam dokumen atau akan memberikan nilai kesesuaian yang semakin besar.
2. *Inverse Document Frequency* (IDF) adalah pengurangan dominasi *term* yang sering muncul di berbagai dokumen. Hal ini diperlukan karena *term* yang banyak muncul di berbagai dokumen, dapat dianggap sebagai *term* umum (*common term*) sehingga tidak penting nilainya. Sebaliknya faktor kejarangmunculan kata (*term scarcity*) dalam koleksi dokumen harus diperhatikan dalam pemberian bobot.

Pada *term frequency*, terdapat beberapa jenis formula yang dapat digunakan yakni :

1. *Term Frequency* Biner adalah nilai *term frequency* yang memperhatikan apakah suatu kata ada atau tidak dalam dokumen, jika ada bernilai satu dan jika tidak bernilai nol.
2. *Term Frequency* Murni adalah nilai *term frequency* yang diberikan berdasarkan jumlah kemunculan suatu kata di dokumen. Misalkan muncul lima kali kata pada dokumen maka kata tersebut akan bernilai lima.
3. *Term Frequency* Logaritmik adalah nilai *term frequency* yang menghindari dominasi dokumen yang mengandung sedikit kata dalam *query*, namun mempunyai frekuensi tinggi.

$$TF = 1 + \log (TF) \quad (3.1)$$

4. *Term Frequency* Normalisasi adalah nilai *term frequency* menggunakan perbandingan antara frekuensi sebuah kata dengan jumlah keseluruhan kata pada dokumen.

$$TF = 0.5 + 0.5 \times \left(\frac{TF}{\max TF} \right) \quad (3.2)$$

Inverse Document Frequency dihitung dengan menggunakan formula sebagai berikut :

$$IDF_j = \ln \left(\frac{D}{DF_j} \right) \quad (3.3)$$

Keterangan :

D : jumlah semua dokumen yang ada dalam koleksi

DF_j : jumlah dokumen yang mengandung *term* (j)

Rumus pada metode TF dan IDF adalah gabungan dari formula perhitungan *row TF* dan formula IDF . Jenis formula *term frequency* yang digunakan untuk perhitungan TF pada penelitian ini adalah TF murni sehingga ketika digunakan dalam setiap kata pada setiap dokumen akan dihitung bobotnya dengan rumus:

$$W_{ij} = TF_{ij} * IDF_j \quad (3.4)$$

$$W_{ij} = TF_{ij} * \ln \left(\frac{D}{DF_j} \right) \quad (3.5)$$

Keterangan:

i : dokumen ke- d

j : kata ke- t dari kata kunci

W_{ij} : bobot *term* (j) terhadap dokumen (i)

TF_{ij} : jumlah kemunculan *term* (j) dalam dokumen (i)

Dari **Persamaan 3.4**, didapatkan hasil bahwa berapapun besarnya nilai TF_{ij} apabila nilai $D = DF_j$ maka didapatkan hasil nol pada perhitungan IDF. Untuk menanggulangi hal tersebut, maka dapat ditambahkan nilai 1 pada sisi IDF, sehingga perhitungan bobotnya dirumuskan menjadi sebagai berikut :

$$W_{ij} = TF_{ij} * \left(\ln \left(\frac{D}{DF_j} \right) + 1 \right) \quad (3.6)$$

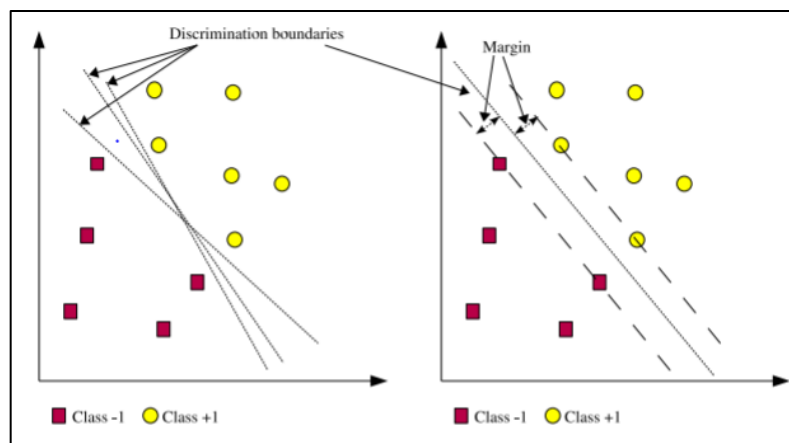
3.14 Data Training dan Data Testing

Didalam membuat model *machine learning*, agar model yang diperoleh memiliki kemampuan generalisasi yang baik dalam melakukan klasifikasi data,

maka data dibagi menjadi data *train* dan data *test*. Data *training* digunakan dalam algoritma klasifikasi untuk membentuk suatu model *classifier* yang cocok. Sedangkan data *testing* digunakan untuk mengukur sejauh mana *classifier* berhadil melakukan klasifikasi dengan benar. Sehingga data yang ada pada *testing* sebaiknya tidak boleh ada pada data *training*.

Proporsi antara data *training* dan data *testing* memanglah tidak mengikat atau sesuai keinginan. Namun agar variasi dalam model tidak terlalu besar, disarankan data *training* lebih besar daripada data *testing*. Menurut Paratu dalam (Thalib, 2018) biasanya 2/3 dari total data dijadikan data *training* sedangkan sisanya dijadikan data *testing*. Selain itu, ada pula penelitian yang menghasilkan keakuratan model klasifikasi optimum dengan proporsi 80:20 dan 90:10 untuk data training dan data testing.

3.15 Support Vector Machine (SVM)



Gambar 3.1. Konsep SVM (Sumber : Nugroho, 2003)

Pertama kali *Support Vector Machine* (SVM) diperkenalkan oleh Boser, Guyon, Vapnik tahun 1992. Konsep dari SVM sendiri adalah usaha untuk menemukan *hyperlane* terbaik pada *input space*. *Hyperplane* adalah garis batas pemisah data antar-kelas. Dimana untuk menemukan *hyperlane* terbaik adalah dengan mengukur margin *hyperlane* tersebut dan mencari titik maksimalnya. Margin adalah jarak antara *hyperplane* dengan data terdekat pada masing-masing kelas. *Pattern* yang terdekat pada masing-masing kelas inilah yang disebut dengan *support vector* (Yunliang, 2010). Pada **Gambar 3.1** terlihat bahwa terdapat

beberapa *pattern* yang merupakan anggota dari 2 buah *class* yakni -1 dan +1 yang telah disimbolkan dengan warna kotak merah untuk *class* -1 dan lingkaran kuning untuk *class* +1.

Data yang ada dinotasikan sebagai $\vec{x}_i \in R^d$ dan *class* dinotasikan sebagai $y_i \in \{-1, +1\}$ untuk $i = 1, 2, \dots, l$, dimana l adalah banyaknya data yang ada. Diasumsikan bahwa kedua kelas tersebut dapat terpisah secara sempurna oleh *hyperlane* berdimensi d yang didefinisikan sebagai berikut

$$\vec{w} \cdot \vec{x} + b = 0 \quad (3.7)$$

Margin yang optimal akan ditemukan ketika nilai jarak antara *hyperlane* dan titik terdekatnya yaitu $\frac{1}{\|\vec{w}\|}$. Hal tersebut dapat dirumuskan sebagai *quadratic programming* yakni mencari titik minimal yang terlihat pada **Persamaan 3.8** dengan memperhatikan *constraint* pada **Persamaan 3.9**.

$$\min_{\vec{w}} \tau(\vec{w}) = \frac{1}{2} \|\vec{w}\|^2 \quad (3.8)$$

$$y_i(\vec{w}, b, \alpha) - 1 \geq 0, \quad \forall_i \quad (3.9)$$

Optimisasi tersebut dapat diselesaikan dengan Metode *Langrange Multipliers*.

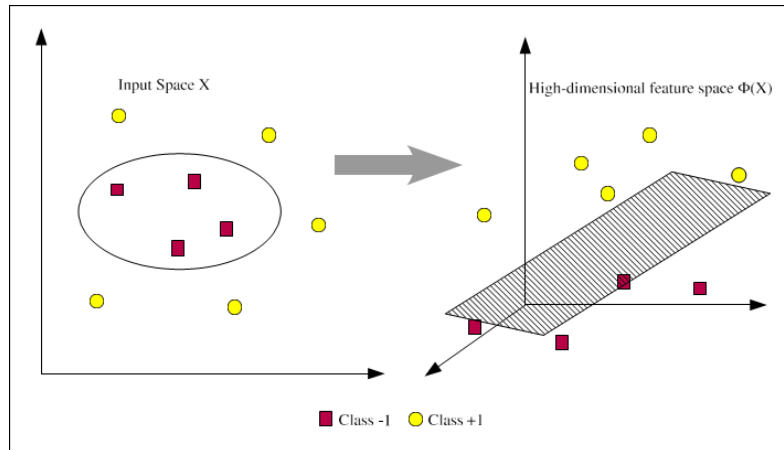
$$L(\vec{w}, b, \alpha) = \frac{1}{2} \|\vec{w}\|^2 - \sum_{i=1}^l \alpha_i (y_i(\vec{x}_i \cdot \vec{w} + b) - 1) \quad (3.10)$$

Akan tetapi jika kedua kelas tersebut tidak terpisah secara sempurna, maka untuk mengatasinya SVM dirumuskan ulang dengan teknik *soft margin*. Dengan menambahkan parameter C untuk mengontrol *trade off* antar margin dan error dari klasifikasi. Maka **Persamaan 3.8** diubah menjadi

$$\min_{\vec{w}} \tau(\vec{w}, \varepsilon) = \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^l \varepsilon_i \quad (3.11)$$

SVM merupakan satu varian dari *linear machine* sehingga hanya dapat dipakai untuk menyelesaikan masalah yang bersifat *linearly separable*. Namun implementasinya, jarang diperoleh masalah data yang bersifat *linear*. Namun semakin berkembangnya SVM, banyak kasus-kasus SVM yang bersifat *non linear*. Cara mengatasi kasus tersebut adalah dengan memasukan konsep *kernel trick* pada ruang kerja berdimensi tinggi. Dalam *non linear SVM*, data \vec{x}_i dipetakan

fungsi $\Phi(\vec{x})$ ke ruang vektor yang berdimensi lebih tinggi sehingga terbentuk ruang vektor baru dimana *hyperlane* tersebut dapat dikonstruksikan. Diilustrasikan sebagai berikut:



Gambar 3.2. Konsep Non Linear SVM (Sumber : Nugroho, 2003)

Pada **Gambar 3.2** terlihat pada gambar disebelah kiri bahwa pada *class* -1 dan +1 berada pada *input space* berdimensi dua yang tidak dapat dipisahkan secara linear. Kemudian pada **Gambar 3.2** disebelah kanan terlihat bahwa fungsi Φ memetakan tiap data pada *input space* tersebut ke ruang vektor baru yang berdimensi tinggi , dalam hal ini dapat dilihat bahwa dari dimensi 2 menjadi dimensi 3 sehingga kedua *class* dapat dipisahkan secara linear oleh sebuah *hyperlane* yang dinotasikan sebagai berikut:

$$\Phi : R^d \rightarrow R^q \quad ; \quad d < q \quad (3.12)$$

Kemudian dilakukan proses *training* yang sama sebagaimana pada *linear SVM*. Proses optimisasi pada fase ini memerlukan perhitungan antara dua buah *example dot product* pada ruang vektor baru. *Dot product* kedua buah vektor t dinotasikan sebagai $\Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)$. Nilai dari *Dot product* kedua buah vektor tersebut dapat dihitung secara tak langsung tanpa mengetahui fungsi transformasi Φ (Nugroho, 2003). Teknik komputasi ini disebut *kernel tricks* yang dirumuskan

$$K(\vec{x}_i, \vec{x}_j) = \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j) \quad (3.13)$$

Hasil klasifikasi dari data \vec{x}_i diperoleh persamaan berikut:

$$f(\Phi(\vec{x})) = \vec{w} \cdot \Phi(\vec{x}) + b \quad (3.14)$$

$$= \sum_{i=1}^n \alpha_i y_i K(\vec{x}, \vec{x}_i) + b \quad (3.15)$$

Keterangan

x_i = data *input* baris ke-i

x_j = data *input* baris ke-j

y_i = *class output* baris ke-i

b = bias

α_i = *support vector*

w = pembobot

q = dimensi ke -q

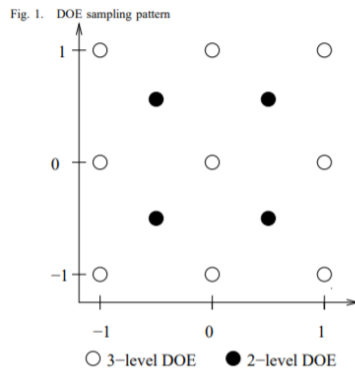
Terdapat berbagai jenis fungsi yang dapat dipakai sebagai *kernel* K seperti pada **Tabel 3.1**

Tabel 3.1. Macam-Macam Fungsi *Kernel* (Sumber : Nugroho, 2003)

Fungsi <i>Kernel</i>	Definisi
<i>Linear</i>	$K(\vec{x}_i, \vec{x}_j) = \vec{x}_i \cdot \vec{x}_j$
<i>Polynomial</i>	$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^p$
<i>Radial Basic Function (RBF)</i>	$K(\vec{x}_i, \vec{x}_j) = \exp\left(-\frac{\ \vec{x}_i - \vec{x}_j\ ^2}{2\sigma^2}\right)$
<i>Sigmoid</i>	$K(\vec{x}_i, \vec{x}_j) = \tanh(\alpha \vec{x}_i \cdot \vec{x}_j + \beta)$

3.16 Proses *Tuning* Data

Dalam metode *support vector machine*, terdapat dua atau lebih parameter yang digunakan pada proses *training* dan parameter tersebut berpengaruh terhadap hasil klasifikasi. Untuk mendapatkan nilai parameter yang optimal digunakanlah *tuning*. Proses *tuning* adalah proses yang digunakan untuk mengoptimalkan sistem proses serta meminimalisasikan nilai *error* antar variabel. Dengan menggunakan sebuah algoritma yang berdasar pada prinsip *design of experiment* (DOE), dapat dengan mudah mengidentifikasi parameter yang optimal atau paling tidak dapat mengidentifikasi parameter yang mendekati optimal.



Gambar 3.3. Contoh *Pattern* DOE (Sumber : Staenlin, 2003)

Dari **Gambar 3.3** menunjukkan pola untuk mencari nilai dua parameter, dimana itu merupakan kombinasi dari n parameter. Diketahui pada titik berwarna putih adalah titik yang menunjukkan tiga dimensi $\{-1, 0, +1\}$ dari rancangan percobaan n parameter. Kemudian titik hitam adalah titik yang menunjukkan dua dimensi $\{-0.5, 0.5\}$. Dari kedua titik tersebut menghasilkan 13 iterasi titik pada setiap iterasi dalam dua parameter yang kemudian digunakan mesin untuk mengevaluasi semua sampel pada tiap iterasi dan dipilih titik terbaik (Staelin, 2003).

3.17 Confusion Matrix

Menurut Han dan Kamber (2011:365) dalam (Said, 2016) *Confusion matrix* adalah alat yang berguna untuk menganalisis seberapa baik *classifier* mengenali *tuple* dari kelas yang berbeda. *True positive* (TP) dan *true negative* (TN) memberikan informasi ketika klasifikasi benar sedangkan apabila *false* (FP) *positive* dan *false negative* (FN) memberikan informasi ketika klasifikasi salah. Ilustrasi *confusion matrix* sebagai berikut:

Tabel 3.2. *Confusion Matrix*

		<i>Predicted</i>		Total
		<i>Yes</i>	<i>No</i>	
<i>Actual</i>	<i>Yes</i>	TP (<i>Correct result</i>)	TN (<i>Unexpected result</i>)	P
	<i>No</i>	FP (<i>Missing result</i>)	FN (<i>Corect absence of result</i>)	N
Total		P'	N'	P+N

Berdasarkan *confusion matrix*, dapat ditentukan nilai *precision*, *recall*, *accuracy*, dan *specificity*. *Precision* adalah tingkat ketepatan sistem dalam menjawab informasi yang diminta oleh *users*. *Recall* adalah tingkat keberhasilan sistem didalam menemukan kembali sebuah informasi. *Accuracy* adalah tingkat kedekatan antara nilai actual dengan nilai prediksi. Secara umum *precision*, *recall* dan *accuracy* dirumuskan sebagai berikut:

$$Precision = \frac{TP}{TP+FP} \quad (3.16)$$

$$Recall = \frac{TP}{TP+FN} \quad (3.17)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.18)$$

BAB 4 METODOLOGI PENELITIAN

4.1 Populasi dan Sampel

Populasi yang digunakan yakni kasus-kasus tindak pidana yang terdapat di Pengadilan Negeri Sleman. Sedangkan sampel penelitian yang dibutuhkan yakni kasus-kasus tindak pidana di Pengadilan Negeri Sleman pada tahun 2016-2020 yakni berjumlah 2247 kasus.

4.2 Sumber dan Jenis Data

Sumber data berasal dari *website* resmi sistem informasi penelusuran perkara di Pengadilan Negeri Sleman yakni <http://pn-sleman.go.id/sipp/>. Sedangkan jenis data yang digunakan dalam penelitian ini adalah data sekunder. Data berupa data rekapan perkara pidana di Pengadilan Negeri Sleman pada Januari 2016 hingga Januari 2020 berupa tuntutan pada pidana biasa dan putusan berdasarkan lama pemidanaan dengan status perkara minutasi.

4.3 Variabel Penelitian

Variabel yang digunakan pada penelitian ini ditampilkan dalam **Tabel 4.1**.

Tabel 4.1 Tabel Variabel dan Definisi Operasional Variabel

Variabel	Definisi Operasional Variabel
1) Surat Tuntutan (tuntutan)	Surat yang membuat pembuktian dari surat dakwaan berdasarkan alat bukti yang telah terungkap di persidangan . Selain itu, juga merupakan kesimpulan penuntut umum tentang kesalahan terdakwa yang disertai dengan tuntutan pidana.
2) Lama Pemidanaan (putusan)	Penahanan kemerdekaan seseorang karena melakukan tindak pidana.

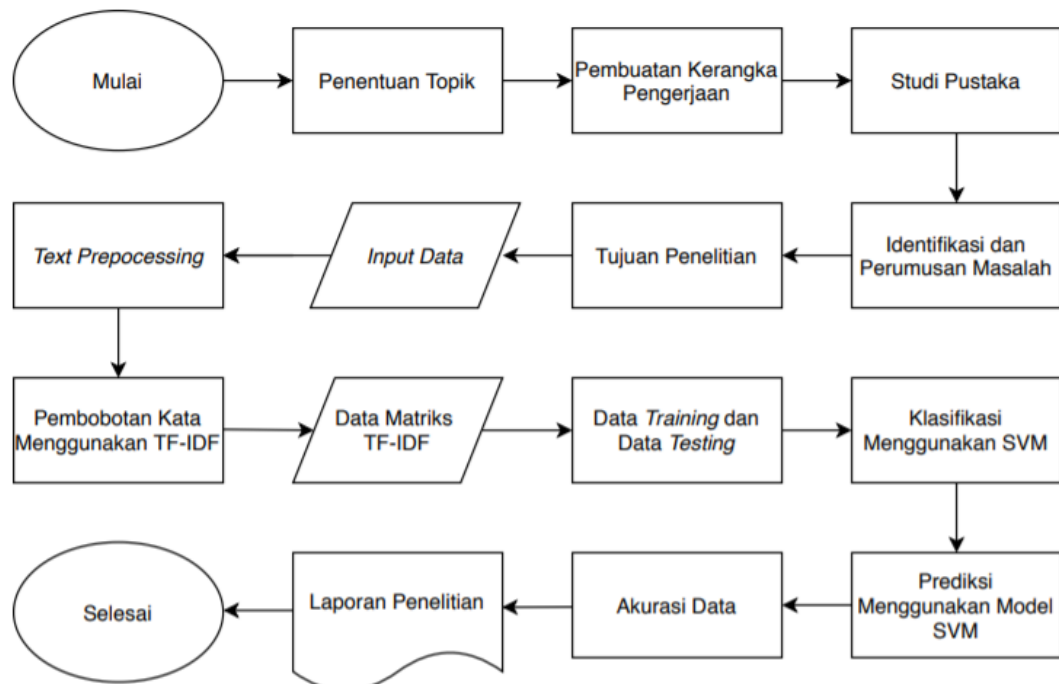
Kelas/kategori/class yang ditentukan peneliti berdasarkan maksimum lama pemidanaan pada tiap jenis pidana. Apabila lama pemidanaan ≥ 1 tahun maka akan masuk pada class lebih dari satu tahun, jika <1 tahun maka akan masuk dalam class kurang dari satu tahun.

4.4 Metode Analisis Data

Software yang digunakan pada penulisan ini adalah *Microsoft Excel 2013*, dan *Anaconda3 5.3.1*. Terdapat metode analisis data yang digunakan dalam penelitian ini yaitu :

1. *Text Mining* dalam hal ini *preprocessing* data yang digunakan adalah *case folding*, *remove punctuation*, *removal stopwords*, dan *tokenization*. Kemudian untuk pembobotan kata menggunakan TF-IDF.
2. Metode SVM digunakan untuk klasifikasi dan prediksi lama pemidanaan. SVM yang digunakan adalah SVM *linear*, *RBF*, *sigmoid*, dan *polynomial* menggunakan parameter $C = 0.1, 1, 10, 100, 1000$ dan parameter $\gamma = 1, 0.1, 0.01, 0.001, 0.0001$.

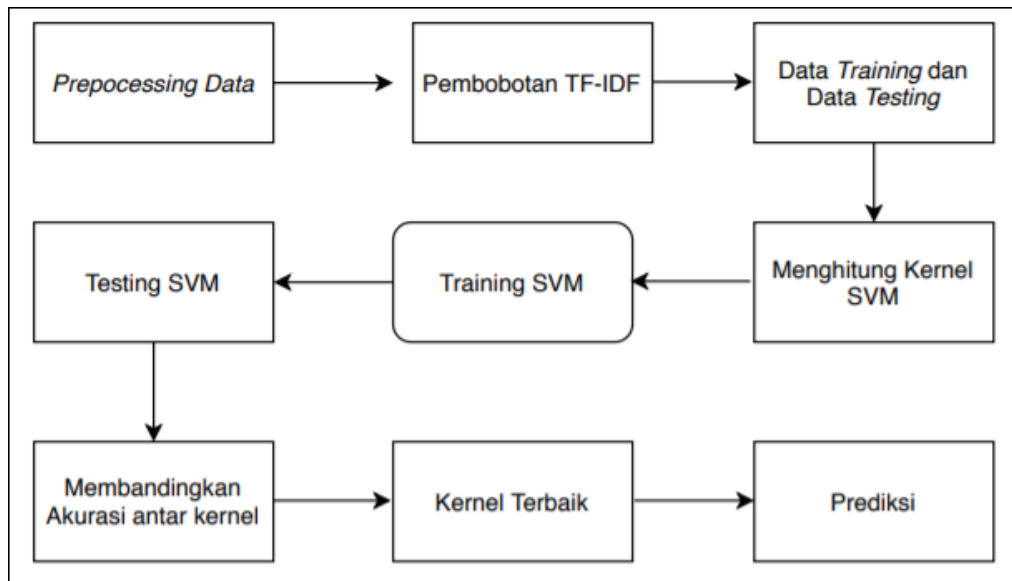
4.5 Tahapan Penelitian



Gambar 4.1. Tahapan Analisis Penelitian

4.5.1. Tahapan Klasifikasi dan Prediksi Menggunakan SVM

Pada penelitian ini, klasifikasi SVM yang digunakan adalah *kernel linear*, *polynomial*, *sigmoid*, dan RBF. Perbandingan diperlukan untuk mengetahui tingkat akurasi kinerja mesin dalam melakukan klasifikasi dari keempat *kernel* tersebut. Sehingga akan ditemukan *kernel* terbaik. Adapun tahapannya sebagai berikut:



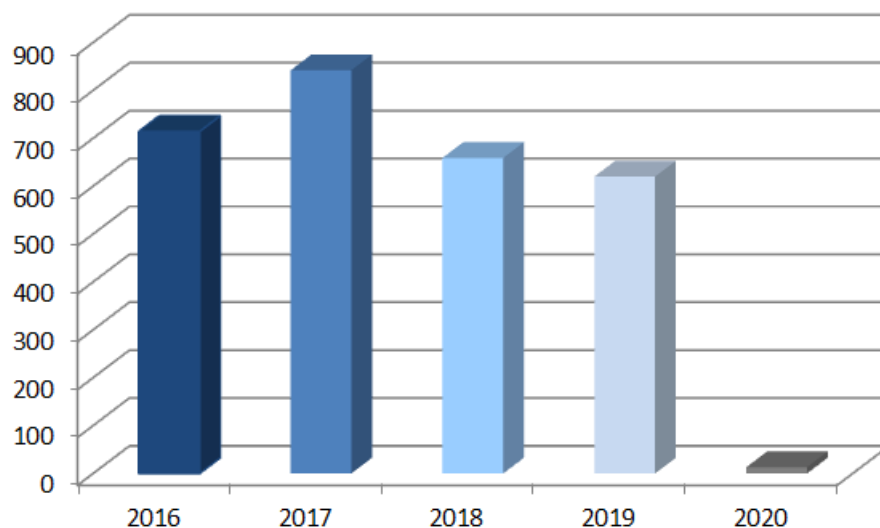
Gambar 4.2. Tahapan Klasifikasi dan Prediksi Menggunakan SVM

BAB 5 HASIL DAN PEMBAHASAN

Pada bab ini berisi mengenai hasil deskriptif dari data yang telah diperoleh sebagai studi pendahuluan. Kemudian sebelum memasuki tahap prediksi, dilakukan klasifikasi terlebih dahulu untuk mengetahui jenis *kernel* apa yang terbaik dan mengetahui seberapa akurat mesin mengklasifikasi data terhadap class atau kategori yang telah dibuat.

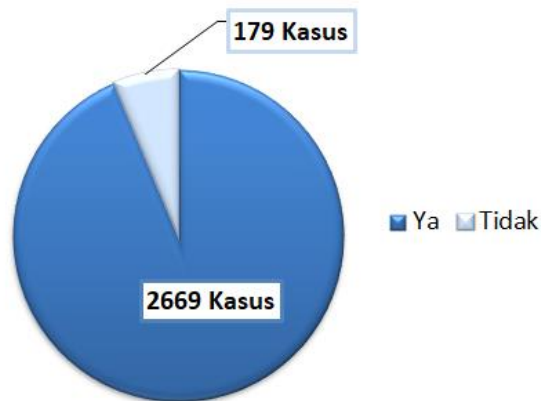
5.1 Analisis Deskriptif

Pada tahap awal studi pendahuluan ini, analisis deskriptif digunakan untuk mengetahui gambaran secara umum dari data yang telah diperoleh mengenai kasus-kasus tindak pidana di Pengadilan Negeri Sleman pada tahun 2016-2020.



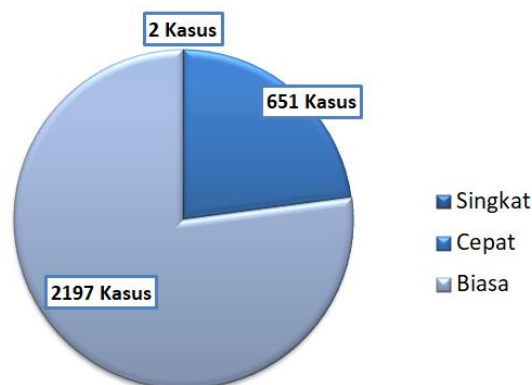
Gambar 5.1. Jumlah Kasus Tindak Pidana Per Tahun

Pada **Gambar 5.1** dijelaskan mengenai jumlah kasus tindak pidana pada 5 tahun terakhir yakni dari Januari 2016 hingga Januari 2020. Diketahui pada tahun 2016 menuju tahun 2017 mengalami peningkatan. Namun pada tahun 2017 menuju tahun 2018 serta tahun 2018 menuju tahun 2019 mengalami penurunan yang lumayan signifikan. Sedangkan tahun 2020 belum dapat di jelaskan karena data yang digunakan hanya data 1 bulan saja. Artinya, tindak kejahatan dan pelanggaran yang ada di Sleman dapat diketahui sudah berkurang pada 4 tahun terakhir.



Gambar 5.2. Kasus Tindak Pidana Berdasarkan Publikasi

Selain itu, pada penelitian ini menggunakan kasus tindak pidana yang dipublikasikan. Dapat dilihat pada **Gambar 5.2** diketahui bahwa kasus yang dipublikasikan sekitar 94% dari total kasus pidana dan kasus tidak di publikasikan sekitar 6% dari total kasus pidana yang berarti data disamarkan. Sehingga dapat diketahui kasus yang dapat dipublikasi dan digunakan oleh peneliti hanya 2669 kasus pidana atau sekitar 94% dari total kasus pidana yang ada di PN Sleman Januari 2016-Januari 2020.



Gambar 5.3. Kasus Tindak Pidana Berdasarkan Jenis Acara Pidana

Pada **Gambar 5.3** juga dapat diketahui bahwa kasus tindak pidana berdasarkan jenis acara pidana dibagi menjadi 3 yakni acara singkat, cepat, dan biasa. Dari 3 jenis acara pidana tersebut dapat diketahui bahwa jenis acara pidana yang memiliki kasus terbanyak adalah jenis acara biasa. Artinya banyak kasus yang bentuk akhir dari kasus tersebut adalah putusan. Dan pada penelitian ini, peneliti hanya menggunakan jenis acara pidana biasa, sehingga data yang digunakan sebesar 2197 kasus.

kemudian kata ‘terbukti’, ‘bersalah’, ‘perjudian’, ‘kertoarjo’, dan lain sebagainya. Dan dari kata tersebut dapat diketahui pula bahwa dari semua kasus tindak pidana banyak yang terbukti bersalah dan kasus yang sering muncul yakni kasus perjudian. Adanya cukup banyak nama muncul dalam *wordcloud* karena dalam tiap dokumen penyebutan nama disebut berulang-ulang.

Setelah mengetahui gambaran umum dari kasus tindak pidana yang ada di PN Sleman pada Januari 2016 hingga Januari 2020. Selanjutnya untuk masuk ke tahap klasifikasi dan prediksi data, dilakukan lah *preprocessing data*.

5.2 *Preprocessing Data*

Tahap pertama sebelum klasifikasi dan prediksi data dilakukan *preprocessing data* terlebih dahulu. Hal tersebut bertujuan untuk membersihkan dan menghilangkan teks yang tidak perlu pada data. Berikut adalah contoh data yang digunakan dalam penelitian ini.

Tabel 5.1. Data Awal Sebelum *Preprocessing*

Tuntutan	Class/Kategori
Menuntut Supaya Majelis Hakim Pengadilan Negeri Sleman yang memeriksa dan mengadili perkara ini, memutuskan : Menyatakan Terdakwa NUR ROHMAN EKO PRASETYO telah terbukti secara sah dan meyakinkan bersalah melakukan tindak pidana “Pencurian dengan kekerasan” sebagaimana di maksud dalam Pasal 365 ayat (2) ke-1,2 KUHP menjatuhkan pidana terhadap Terdakwa NUR ROHMAN EKO PRASETYO dengan pidana penjara selama 1 (satu) tahun dikurangi selama terdakwa berada dalam tahanan, dengan perintah agar terdakwa tetap di tahan; Menyatakan barang bukti : 1 (satu) unit sepeda motor Honda Vario tahun 2014 wrn putih Nopol.AB-6323- RN 1 (satu) unit sepeda motor Honda Beat wrn Hitam tahun 2011 Nopol.AB-6671-JU berikut STNK) 1 (satu) buah Hp	Kurang dari 1 tahun

merk SAMSUNG wrn putih seri J1 Dipergunakan dalam perkara lain a.n Terdakwa DAMAI Als DAMEN Dkk Menetapkan agar terdakwa dibebani membayar biaya perkara sebesar Rp. 2.000,- (dua ribu rupiah);	
---	--

Terdapat beberapa tahapan *preprocessing data* yang digunakan pada penelitian ini, seperti *case folding*, *remove punctuation*, *stopword removal*, dan *tokenizing*.

5.2.1. Case Folding

Case folding adalah tahap dalam mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf yang akan diubah, yakni dari alphabet “a” hingga “z”. Tujuan dari *case folding* ini adalah agar tidak ada kata yang mempunyai arti sama walaupun terdeteksi memiliki penulisan yang berbeda. Hasil dari *case folding* sebagai berikut

Tabel 5.2. Hasil *Case Folding*

Tuntutan Sebelum	Tuntutan Sesudah
<p>Menuntut <u>S</u>upaya <u>M</u>ajelis <u>H</u>akim <u>P</u>engadilan <u>N</u>egeri <u>S</u>leman yang memeriksa dan mengadili perkara ini, memutuskan : <u>M</u>enyatakan Terdakwa <u>NUR ROHMAN EKO PRASETYO</u> telah terbukti secara sah dan meyakinkan bersalah melakukan tindak pidana “<u>P</u>encurian dengan kekerasan” sebagaimana di maksud dalam <u>P</u>asal 365 ayat (2) ke-1,2 <u>KUHP</u> menjatuhkan pidana terhadap <u>T</u>erdakwa <u>NUR ROHMAN EKO</u> <u>PRASETYO</u> dengan pidana penjara</p>	<p>menuntut <u>s</u>upaya <u>m</u>ajelis <u>h</u>akim <u>p</u>engadilan <u>n</u>egeri <u>s</u>leman yang memeriksa dan mengadili perkara ini, memutuskan : <u>m</u>enyatakan terdakwa <u>nur rohman eko prasetyo</u> telah terbukti secara sah dan meyakinkan bersalah melakukan tindak pidana “<u>p</u>encurian dengan kekerasan” sebagaimana di maksud dalam <u>p</u>asal 365 ayat (2) ke- 1,2 <u>kuhp</u> menjatuhkan pidana terhadap <u>t</u>erdakwa <u>nur rohman eko</u> <u>prasetyo</u> dengan pidana penjara selama 1 (satu) tahun dikurangi selama</p>

<p>selama 1 (satu) tahun dikurangi selama terdakwa berada dalam tahanan, dengan perintah agar terdakwa tetap di tahan; <u>M</u>enyatakan barang bukti : 1 (satu) unit sepeda motor <u>H</u>onda <u>V</u>ario tahun 2014 wrn putih <u>N</u>opol.<u>AB</u>-6323-<u>RN</u> 1 (satu) unit sepeda motor <u>H</u>onda <u>B</u>eat wrn <u>H</u>itam tahun 2011 <u>N</u>opol.<u>AB</u>-6671-JU berikut <u>STNK</u>) 1 (satu) buah <u>H</u>p merk <u>SAMSUNG</u> wrn putih seri <u>J1</u> <u>D</u>ipergunakan dalam perkara lain a.n <u>T</u>erdakwa <u>DAMAI ALS DAMEN Dkk</u> <u>M</u>enetapkan agar terdakwa dibebani membayar biaya perkara sebesar Rp. 2.000,- (dua ribu rupiah);</p>	<p>terdakwa berada dalam tahanan, dengan perintah agar terdakwa tetap di tahan; <u>m</u>enyatakan barang bukti : 1 (satu) unit sepeda motor <u>h</u>onda <u>v</u>ario tahun 2014 wrn putih <u>n</u>opol.<u>ab</u>-6323-<u>rn</u> 1 (satu) unit sepeda motor <u>h</u>onda <u>b</u>eat wrn <u>h</u>itam tahun 2011 <u>n</u>opol.<u>ab</u>-6671-ju berikut <u>stnk</u>) 1 (satu) buah <u>h</u>p merk <u>samsung</u> wrn putih seri <u>j1</u> <u>d</u>ipergunakan dalam perkara lain a.n <u>t</u>erdakwa <u>damai als damen dkk</u> <u>m</u>enetapkan agar terdakwa dibebani membayar biaya perkara sebesar rp. 2.000,- (dua ribu rupiah);</p>
---	--

5.2.2. Remove Punctuation

Remove punctuation adalah tahap menghapus tanda baca yang ada dalam dokumen. Seperti pada **Tabel 5.3** dapat terlihat saat tuntutan sebelum terdapat tanda baca maka setelah diproses saat tuntutan sesudah tanda baca telah terhapus.

Tabel 5.3. Hasil *Remove Punctuation*

Tuntutan Sebelum	Tuntutan Sesudah
<p>menuntut supaya majelis hakim pengadilan negeri sleman yang memeriksa dan mengadili perkara ini, memutuskan : menyatakan terdakwa nur rohman eko prasetyo telah terbukti secara sah dan meyakinkan bersalah melakukan tindak pidana “pencurian</p>	<p>menuntut supaya majelis hakim pengadilan negeri sleman yang memeriksa dan mengadili perkara ini memutuskan menyatakan terdakwa nur rohman eko prasetyo telah terbukti secara sah dan meyakinkan bersalah melakukan tindak pidana pencurian</p>

<p>dengan kekerasan” sebagaimana di maksud dalam pasal 365 ayat (2) ke-1.2 kuhp menjatuhkan pidana terhadap terdakwa nur rohman eko prasetyo dengan pidana penjara selama 1 (satu) tahun dikurangi selama terdakwa berada dalam tahanan, dengan perintah agar terdakwa tetap di tahan; menyatakan barang bukti : 1 (satu) unit sepeda motor honda vario tahun 2014 wrn putih nopol_ab_6323_ rn 1 (satu) unit sepeda motor honda beat wrn hitam tahun 2011 nopol_ab_6671_ju berikut stnk) 1 (satu) buah hp merk samsung wrn putih seri j1 dipergunakan dalam perkara lain a_n terdakwa damai als damen dkk menetapkan agar terdakwa dibebani membayar biaya perkara sebesar rp. 2.000,- (dua ribu rupiah);</p>	<p>dengan kekerasan sebagaimana di maksud dalam pasal 365 ayat 2 ke 1 2 kuhp menjatuhkan pidana terhadap terdakwa nur rohman eko prasetyo dengan pidana penjara selama 1 satu tahun dikurangi selama terdakwa berada dalam tahanan dengan perintah agar terdakwa tetap di tahan menyatakan barang bukti 1 satu unit sepeda motor honda vario tahun 2014 wrn putih nopol ab 6323 rn 1 satu unit sepeda motor honda beat wrn hitam tahun 2011 nopol ab 6671 ju berikut stnk 1 satu buah hp merk samsung wrn putih seri j1 dipergunakan dalam perkara lain a n terdakwa damai als damen dkk menetapkan agar terdakwa dibebani membayar biaya perkara sebesar rp 2 000 dua ribu rupiah</p>
---	--

5.2.3. Stopword Removal

Stopword removal adalah tahap dimana kata-kata yang bukan merupakan ciri suatu dokumen akan dihilangkan. Kata-kata yang dihilangkan sudah memiliki kamus, jika ada kata-kata yang ingin dihilangkan selain yang ada pada kamus, maka peneliti dapat menambahkan kata tersebut di kamus. Dan pada penelitian ini, peneliti menambahkan *removal word* yakni seperti kata 'nya', 'kali', 'sih', 'yg', 'pengadilan', 'negeri', 'majelis', 'hakim', 'pidana', 'als', 'tindak', 'terdakwa', 'memeriksa', 'mengadili', 'memutuskan', 'menuntut', 'perkara', 'hari', 'bulan', 'tahun', 'jam', 'tanggal', 'alias', 'bahwa', 'kesatu', 'kedua', 'pertama', 'ketiga', 'wib', 'pp', 'sleman', 'yogyakarta', 'huruf', 'kabupaten', 'baiknya', 'berkali', 'bin',

'kurangnya', 'mata', 'olah', 'sekurang', 'setidak', 'tama', 'tidaknya'. Sehingga hasil dari *stopword removal* dapat terlihat pada **Tabel 5.4**

Tabel 5.4. Hasil *Stopword Removal*

Tuntutan Sebelum	Tuntutan Sesudah
<p><u>menuntut supaya majelis hakim pengadilan negeri sleman yang memeriksa dan mengadili perkara ini memutuskan menyatakan terdakwa</u> nur rohman eko prasetyo <u>telah</u> terbukti <u>secara</u> sah <u>dan meyakinkan</u> bersalah <u>melakukan tindak pidana</u> pencurian <u>dengan</u> kekerasan <u>sebagaimana di</u> maksud <u>dalam pasal</u> 365 <u>ayat</u> 2 <u>ke</u> 1 2 kuhp menjatuhkan <u>pidana terhadap terdakwa</u> nur rohman eko prasetyo <u>dengan pidana</u> penjara <u>selama</u> 1 <u>satu</u> tahun dikurangi <u>selama terdakwa berada dalam</u> tahanan <u>dengan</u> perintah <u>agar terdakwa tetap di tahan menyatakan</u> barang bukti 1 <u>satu</u> unit sepeda motor honda vario tahun 2014 wrn putih nopol ab 6323 rn 1 satu unit sepeda motor honda beat wrn hitam tahun 2011 nopol ab 6671 ju <u>berikut</u> stnk 1 <u>satu</u> buah hp merk samsung wrn putih seri j1 <u>dipergunakan dalam perkara lain</u> a n <u>terdakwa</u> damai <u>als</u> damen dkk menetapkan <u>agar terdakwa</u> dibebani membayar biaya <u>perkara sebesar</u> rp 2 000 dua ribu rupiah</p>	<p>nur rohman eko prasetyo terbukti sah bersalah pencurian kekerasan maksud 365 2 ke12 kuhp menjatuhkan nur rohman eko prasetyo penjara 1 tahundikurangi tahanan perintah tahan barang bukti 1 unit sepeda motor honda variotahun 2014 wrn putih nopol ab6323 rn 1 unit sepeda motor honda beat wrn hitam 2011 nopol ab6671 ju stnk 1 buah hp merk samsung wrn putih seri j1 an damai damen dkk menetapkan dibebani membayar biaya rp 2000 ribu rupiah</p>

5.2.4. Tokenizing

Tokenizing adalah tahap mengubah sekumpulan teks menjadi kata-kata atau potongan kata tunggal yang dapat dilihat pada **Tabel 5.5**. Acuan pemisah dari setiap *token* biasanya adalah spasi dan tanda baca. *Tokenizing* dilakukan untuk memperoleh potongan kata atau *token* yang akan menjadi entitas yang memiliki nilai dalam penyusunan matriks dokumen untuk proses berikutnya.

Tabel 5.5. Hasil *Tokenizing*

Tuntutan Sebelum	Tuntutan Sesudah
nur rohman eko prasetyo terbukti sah bersalah pencurian kekerasan maksud 365 2 ke12 kuhp menjatuhkan nur rohman eko prasetyo penjara 1 tahundikurangi tahanan perintah tahan barang bukti 1 unit sepeda motor honda vario tahun 2014 wrn putih nopolab6323 rn 1 unit sepeda motor honda beat wrn hitam 2011 nopolab6671 ju stnk 1 buah hp merk samsung wrn putih seri j1 an damai damen dkk menetapkan dibebani membayar biaya rp 2000 ribu rupiah	['nur', 'rohman', 'eko', 'prasetyo', 'terbukti', 'sah', 'bersalah', 'pencurian', 'kekerasan', 'maksud', '365', '2', 'ke12', 'kuhp', 'menjatuhkan', 'nur', 'rohman', 'eko', 'prasetyo', 'penjara', '1', 'tahun', 'dikurangi', 'tahanan', 'perintah', 'tahan', 'barang', 'bukti', '1', 'unit', 'sepeda', 'motor', 'honda', 'vario', 'tahun', '2014', 'wrn', 'putih', 'nopolab6323', 'rn', '1', 'unit', 'sepeda', 'motor', 'honda', 'beat', 'wrn', 'hitam', '2011', 'nopolab6671ju', 'stnk', '1', 'buah', 'hp', 'merk', 'samsung', 'wrn', 'putih', 'seri', 'j1', 'an', 'damai', 'damen', 'dkk', 'menetapkan', 'dibebani', 'membayar', 'biaya', 'rp', '2000', 'ribu', 'rupiah']

5.3 Pembobotan TF-IDF

Setelah melakukan *preprocessing data*, data yang akan dianalisis pada proses berikutnya adalah *word embedding* yakni melakukan konversi sebuah teks menjadi angka, metode yang digunakan adalah metode pembobotan TF-IDF. Metode pembobotan TF-IDF dapat dipakai untuk mengubah data kata menjadi

data numerik. Komponen TF-IDF terdiri dari bobot TF dan bobot IDF, dimana nilai pembobotan TF-IDF berasal dari hasil perkalian antara bobot TF dan IDF. Dasar dari pembagian kelas dari TF-IDF tersebut tergantung dari dokumen tiap kata tersebut. Misalkan jika dokumen X ada pada kelas kurang dari satu tahun, maka kata-kata yang ada di dokumen X akan masuk pada kelas kurang dari satu tahun. Pada **Tabel 5.6** adalah sampel kata yang digunakan untuk memperlihatkan hasil pembobotan kata TF-IDF.

Tabel 5.6 Sampel Hasil Pembobotan Kata TF-IDF

<i>Class</i>	abadi	aman	ambil	...	zat
:	:	:	:	...	:
Kurang dari satu tahun	0	0	8.717351	...	0
Kurang dari satu tahun	0	0	0	...	0
:	:	:	:	...	:
Kurang dari satu tahun	6.925592		0	...	0
Kurang dari satu tahun	0	0	0	...	0
Kurang dari satu tahun	0	0	0	...	0
:	:	:	:	...	:
Kurang dari satu tahun	0	7.618739	0	...	0
Kurang dari satu tahun	0	0	0	...	0
Kurang dari satu tahun	0	0	0		0
:	:	:	:	...	:
Kurang dari satu tahun	0	0	0	...	8.717351
:	:	:	:	...	:
Lebih dari satu tahun	0	0	0	...	0

Berdasarkan **Tabel 5.6** adalah sampel hasil pembobotan TF-IDF pada *software*, jika terdapat nilai nol (0) artinya tidak mengandung kata pada dokumen sampel yang dipilih. Untuk menghitung nilai TF-IDF secara manual, langkah pertama yakni menghitung nilai TF.

Tabel 5.7. Sampel Hasil Nilai TF

<i>Class</i>	abadi	aman	Ambil	...	zaman
:	:	:	:	...	:
Kurang dari satu tahun	0	0	1	...	0
Kurang dari satu tahun	0	0	0	...	0
:	:	:	:	...	:
Kurang dari satu tahun	1	0	0	...	0
Kurang dari satu tahun	0	0	0	...	0
Kurang dari satu tahun	0	0	0	...	0
:	:	:	:	...	:
Kurang dari satu tahun	0	1	0	...	0
Kurang dari satu tahun	0	0	0	...	0
Kurang dari satu tahun	0	0	0	...	0
:	:	:	:	...	:
Kurang dari satu tahun	0	0	0	...	1
:	:	:	:	...	:
Lebih dari satu tahun	0	0	0	...	0

Untuk contoh menghitung manual TF-IDF, peneliti mengambil kata “abadi” pada dokumen 245 yang masuk ke kategori kurang dari satu tahun. **Tabel 5.7** adalah hasil perhitungan sampel nilai TF. Diketahui frekuensi kata “abadi” pada dokumen 245 terdapat 1 kemunculan. Kemudian setelah menghitung nilai TF, dilanjut dengan menghitung nilai DF. Nilai DF dari kata “abadi” dari semua dokumen adalah sebanyak 6. Perhitungan manual IDF menggunakan **Persamaan 3.6**

$$W_{ij} = TF_{ij} * \left(\ln \left(\frac{D}{DF_j} \right) + 1 \right) = 1 * \left(\ln \left(\frac{2247}{6} \right) + 1 \right) = 6,925592$$

Sehingga didapatkan nilai TF-IDF untuk kata “abadi” pada dokumen 245 sebesar 6.925592. Dan dapat disimpulkan bahwa, perhitungan dengan *software* dan pengerjaan manual memiliki nilai bobot yang sama.

Pada proses *word embedding* ini, akan muncul sebuah kamus dari seluruh kata yang *unique*. Dan *unique word* yang digunakan dalam penelitian ini adalah sebesar 21789 kata pada 2247 data atau dokumen.

5.4 Data Training dan Data Testing

Setelah dilakukan TF-IDF, kemudian melakukan klasifikasi menggunakan SVM. Untuk melakukan klasifikasi menggunakan SVM, terlebih dahulu menentukan data *training* dan data *testing*. Dalam penelitian ini, SVM memisahkan *feature* atau variabel yang berupa kata yang telah dibobot dengan TF-IDF dan jumlah *feature* yang digunakan penelitian ini adalah sebanyak 21813 *feature* yang nantinya akan dibagi sesuai dengan proporsi data *training* dan *testing*. Data *training* ini digunakan saat proses klasifikasi untuk melatih mesin mempelajari pola data, dan data *testing* digunakan untuk mengetes mesin mengenai hasil pola data. Data yang digunakan peneliti memiliki dua katogeri atau *class* yakni kurang dari satu tahun dan lebih dari satu tahun. Pada penelitian saat ini, menggunakan pembagian data *training* dan data *testing* sebesar 80:20. Berikut adalah perhitungan mengenai data *training* dan data *testing*.

$$\text{Data Training} = \frac{80}{100} \times 2247 = 1797$$

$$\text{Data Testing} = \frac{20}{100} \times 2247 = 450$$

Pembagian data *training* dan data *testing* pada masing-masing *class* dapat dilihat pada

Tabel 5.8. Pembagian Data *Training* dan Data *Testing*

<i>Class</i>	<i>Data Training</i>	<i>Data Testing</i>	Total
Lebih dari satu tahun	771	183	954
Kurang dari satu tahun	1026	267	1293
Total	1797	450	2247

Pada **Tabel 5.8** dapat diketahui bahwa data *training* yang digunakan sebagai model awal sebesar 80% dari total data 2247 dokumen. Sehingga diperoleh jumlah data *training* sebanyak 1797 dokumen untuk data *training*. Dengan

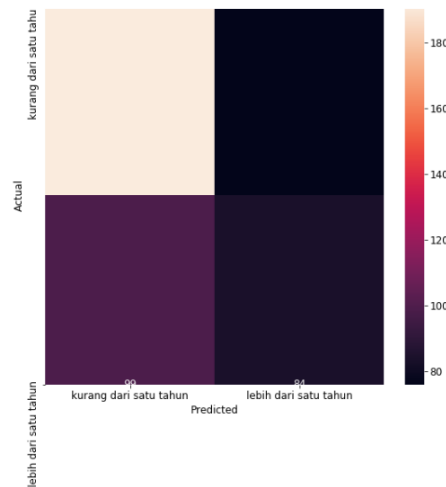
pembagian data untuk masing-masing *class* yakni pada *class* lebih dari satu sebanyak 771 dokumen dan *class* kurang dari satu sebanyak 1026 dokumen. Sedangkan untuk data *testing* yang digunakan sebagai model awal sebesar 20% dari total data 2247 dokumen. Sehingga diperoleh jumlah data *testing* sebanyak 450 dokumen. Dengan pembagian data untuk masing-masing *class* yakni pada *class* lebih dari satu sebanyak 183 dokumen dan *class* kurang dari satu sebanyak 267 dokumen.

5.5 Klasifikasi Support Vector Machine (SVM)

Setelah membagi data menjadi data *training* dan data *testing*, selanjutnya adalah proses klasifikasi menggunakan SVM. SVM bekerja untuk menemukan fungsi pemisah (*hyperlane*) antar klasifikasi. Dengan menggunakan data *training*, mesin akan terlatih menjadi alat analisis untuk mempelajari pola data berdasarkan ciri-ciri yang ada pada tiap data. Setelah melakukan *training* data, kemudian mesin akan menguji data dengan menggunakan data *testing*, sehingga dapat mengetahui tingkat akurasi dalam memprediksi *class* atau kategori pada data baru yang berbeda dengan data *training*. Pada penelitian ini, peneliti menggunakan 4 *kernel* yaitu *linear*, *RBF*, *Sigmoid*, dan *Polynomial* serta beberapa pilihan parameter $C = 0.1, 1, 10, 100, 1000$ dan parameter $\gamma = 1, 0.1, 0.01, 0.001, 0.0001$. Untuk mendapatkan parameter terbaik dari beberapa pilihan parameter, peneliti menggunakan proses *tuning* yang kemudian dicari nilai akurasi yang terbaik dari 4 *kernel* tersebut.

Confusion matrix atau tabulasi silang digunakan untuk mengukur kebaikan klasifikasi yang berisi mengenai data asli (*row*) dan data prediksi (*column*). Nilai akurasi pada *confusion matrix* yakni apabila angka secara tepat berada pada posisi diagonal. Untuk mencari nilai akurasi dapat diketahui dengan cara membagi jumlah prediksi benar dengan jumlah seluruh prediksi. Sedangkan nilai *error* pada *confusion matrix* yakni apabila angka secara tidak tepat berada dalam *diagonal* tersebut. Untuk mencari nilai *error* dapat diketahui dengan membagi jumlah prediksi salah dengan jumlah seluruh prediksi. Dibawah ini adalah hasil dari 4 *kernel* dengan nilai parameter yang telah di *tuning*.

Pertama yakni klasifikasi SVM menggunakan *Kernel Linear*



Gambar 5.6. *Confusion Matrix SVM Kernel Linear*

Berikut pada **Tabel 5.9.** adalah tabel kontigensi hasil dari *confusion matrix Kernel Linear* yang dapat dilihat dari **Gambar 5.6**

Tabel 5.9. *Confusion Matrix Data Testing Kernel Linear*

		<i>Predicted</i>		Total
		KS	LS	
<i>Actual</i>	KS	192	97	289
	LS	75	86	161
Total		267	183	450

Keterangan :

KS : Kurang dari satu tahun ; LS : Lebih dari satu tahun

Berdasarkan **Tabel 5.9** dapat dihitung nilai *error* dan akurasi sebagai berikut:

$$\text{Akurasi} = \frac{\sum(\text{prediksi benar})}{\sum(\text{seluruh prediksi})}$$

$$\text{Error} = \frac{\sum(\text{prediksi salah})}{\sum(\text{seluruh prediksi})}$$

$$\text{Akurasi} = \frac{192 + 86}{450}$$

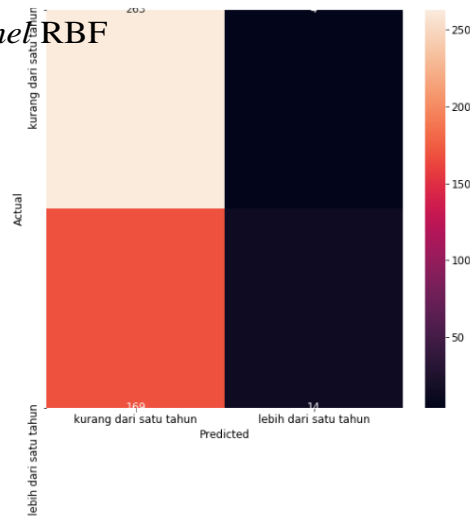
$$\text{Error} = \frac{97 + 75}{450}$$

$$\text{Akurasi} = 0,617$$

$$\text{Error} = 0,383$$

Jadi , dapat diketahui bahwa dengan menggunakan klasifikasi SVM *kernel linear* dengan parameter $C = 0,1$. Didapatkan hasil akurasi sebesar

61.7% dan *error* sebesar 38.3%. Kemudian untuk klasifikasi SVM menggunakan *kernel* RBF



Gambar 5.7. *Confusion Matrix* SVM Kernel RBF

Berikut pada **Tabel 5.10.** adalah tabel kontigensi hasil dari *confusion matrix* Kernel Linear yang dapat dilihat dari **Gambar 5.7**

Tabel 5.10. *Confusion Matrix* Data Testing Kernel RBF

		<i>Predicted</i>		Total
		KS	LS	
<i>Actual</i>	KS	263	169	432
	LS	4	14	18
Total		167	183	450

Keterangan :

KS : Kurang dari satu tahun

LS : Lebih dari satu tahun

Berdasarkan **Tabel 5.10** dapat dihitung nilai *error* dan akurasi sebagai berikut:

$$\text{Akurasi} = \frac{\sum(\text{prediksi benar})}{\sum(\text{seluruh prediksi})}$$

$$\text{Error} = \frac{\sum(\text{prediksi salah})}{\sum(\text{seluruh prediksi})}$$

$$\text{Akurasi} = \frac{263 + 14}{450}$$

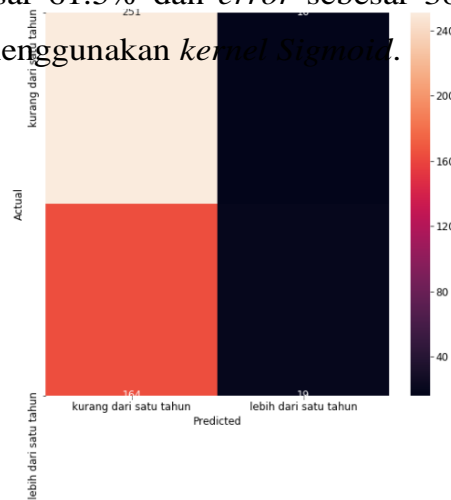
$$\text{Error} = \frac{169 + 4}{450}$$

$$\text{Akurasi} = 0,615$$

$$\text{Error} = 0,385$$

Jadi , dapat diketahui bahwa dengan menggunakan klasifikasi SVM *kernel* RBF dengan parameter C = 1 dan gamma = 0.0001. Didapatkan

hasil akurasi sebesar 61.5% dan *error* sebesar 38.5%. Kemudian untuk klasifikasi SVM menggunakan *kernel Sigmoid*.



Gambar 5.8. *Confusion Matrix SVM Kernel Sigmoid*

Berikut pada **Tabel 5.11.** adalah tabel kontigensi hasil dari *confusion matrix Kernel Sigmoid* yang dapat dilihat dari **Gambar 5.8**

Tabel 5.11. *Confusion Matrix Data Testing Kernel Sigmoid*

		<i>Predicted</i>		Total
		KS	LS	
<i>Actual</i>	KS	251	164	415
	LS	16	19	35
Total		267	183	450

Keterangan :

KS : Kurang dari satu tahun

LS : Lebih dari satu tahun

Berdasarkan **Tabel 5.11** dapat dihitung nilai *error* dan akurasi sebagai berikut:

$$\text{Akurasi} = \frac{\sum(\text{prediksi benar})}{\sum(\text{seluruh prediksi})}$$

$$\text{Error} = \frac{\sum(\text{prediksi salah})}{\sum(\text{seluruh prediksi})}$$

$$\text{Akurasi} = \frac{251 + 19}{450}$$

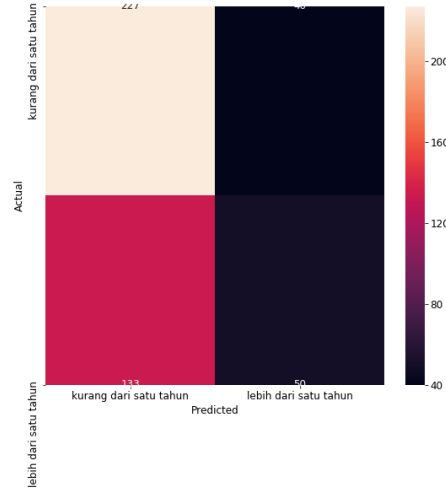
$$\text{Error} = \frac{164 + 16}{450}$$

$$\text{Akurasi} = 0,6$$

$$\text{Error} = 0,4$$

Jadi , dapat diketahui bahwa dengan menggunakan klasifikasi SVM *kernel sigmoid* dengan parameter C = 1 dan gamma = 0.0001. Didapatkan

hasil akurasi sebesar 60% dan *error* sebesar 40%. Yang terakhir adalah klasifikasi SVM menggunakan *kernel polynomial*.



Gambar 5.9. *Confusion Matrix SVM Kernel Polynomial*

Berikut pada **Tabel 5.12.** adalah tabel kontigensi hasil dari *confusion matrix kernel polynomial* yang dapat dilihat dari **Gambar 5.9**

Tabel 5.12. *Confusion Matrix Data Testing Kernel Polynomial*

		<i>Predicted</i>		Total
		KS	LS	
<i>Actual</i>	KS	228	134	362
	LS	39	49	88
Total		267	183	450

Keterangan :

KS : Kurang dari satu tahun

LS : Lebih dari satu tahun

Berdasarkan **Tabel 5.12** dapat dihitung nilai *error* dan akurasi sebagai berikut:

$$\text{Akurasi} = \frac{\sum(\text{prediksi benar})}{\sum(\text{seluruh prediksi})}$$

$$\text{Error} = \frac{\sum(\text{prediksi salah})}{\sum(\text{seluruh prediksi})}$$

$$\text{Akurasi} = \frac{228 + 49}{450}$$

$$\text{Error} = \frac{134 + 39}{450}$$

$$\text{Akurasi} = 0,615$$

$$\text{Error} = 0,385$$

Jadi , dapat diketahui bahwa dengan menggunakan klasifikasi SVM *kernel sigmoid* dengan parameter $C = 0.1$ dan $\gamma = 1$. Didapatkan hasil akurasi sebesar 61.5% dan *error* sebesar 38.5%.

Sehingga dapat disimpulkan bahwa dari hasil analisis SVM menggunakan 4 *kernel* yakni *kernel linear*, *RBF*, *Sigmoid*, dan *Polynomial* dapat dirangkum dalam **Tabel 5.13**

Tabel 5.13. *Confusion Matrix* Data Testing tiap *Kernel*

<i>Kernel</i>	Nilai Akurasi
<i>Linear</i>	61.7%
<i>RBF</i>	61.5%
<i>Sigmoid</i>	60%
<i>Polynomial</i>	61.5%

Dari **Tabel 5.12** didapatkan nilai akurasi dari 4 *kernel* tersebut. Dapat diketahui nilai akurasi terbesar pada *kernel linear* dengan nilai parameter $C=0.1$. Sehingga untuk proses selanjutnya peneliti menggunakan model tersebut.

5.6 Prediksi *Class* atau Kategori Untuk Data Tuntutan Baru

Setelah dilakukan *training* data, didapatkan model klasifikasi SVM dengan menggunakan *kernel linear* dengan nilai parameter $C=0.1$. Model didapatkan dari proses *tunning* data yang kemudian model dievaluasi dengan menggunakan data tuntutan yang baru yang bukan dari data *training* melainkan data baru berbeda-beda yang peneliti coba. Maksud dari berbeda-beda adalah peneliti menggunakan lebih dari satu dokumen tuntutan baru dan mencoba memecah beberapa kata untuk menguji seberapa akurat model dalam memprediksi tiap bagian dalam dokumen.

Tabel 5.14. Hasil Prediksi *Class* untuk Data Tuntutan Baru

Tuntutan	Prediksi	Kesesuaian
Menuntut supaya majelis hakim pengadilan negeri sleman yang memeriksa dan mengadili perkara ini memutuskan : menyatakan terdakwa	Kurang dari satu tahun	Tidak Sesuai

sabrino tralulu secara sah dan meyakinkan telah terbukti bersalah melakukan tindak pidana pasal 36 ayat 3 UU RI no 7 tahun 2011 tentang mata uang		
menyatakan terdakwa alibobo bersalah melakukan tindak penggelapan	Kurang dari satu tahun	Sesuai
Menyatakan terdakwa agus telah bersalah melakukan tindak pidana yang diatur dalam pasal 351 ayat 2 KUHP	Kurang dari satu tahun	Tidak Sesuai
menuntut supaya majelis hakim pengadilan negeri sleman memeriksa dan mengadili perkara ini memutuskan : menyatakan terdakwa barbie bersalah melakukan tindak pidana tanpa hak menyimpan senjata api, sebagaimana diatur dalam pasal 1 ayat 1 UU Darurat no 12 Tahun 1951 tentang senjata api dan bahan peledak sebagaimana dakwaan Penuntut Umum	Kurang dari satu tahun	Sesuai
menuntut Menyatakan Terdakwa agus telah terbukti secara sah dan meyakinkan bersalah melakukan tindak pidana dengan terang-terangan dan dengan tenaga bersama menggunakan kekerasan terhadap orang yang mengakibatkan luka-luka sebagaimana di maksud dalam Pasal 170 ayat (1) jo. Ayat (2) ke-1 KUHP . Menjatuhkan pidana terhadap Terdakwa agus dengan pidana penjara masing-masing selama 8 bulan dengan perintah agar Para Terdakwa tetap ditahan . Dirampas untuk dimusnahkan Menetapkan agar Para terdakwa dibebani membayar biaya perkara masing-	Lebih dari satu tahun	Sesuai

masing sebesar Rp. 2.000,- (dua ribu rupiah); Terhadap tuntutan pidana tersebut, atas pertanyaan		
--	--	--

Pada **Tabel 5.14** adalah hasil dari prediksi *class* atau kategori yang dilakukan oleh mesin menggunakan algoritma SVM dengan *kernel linear*. Dapat terlihat bahwa dari ke 5 pernyataan tuntutan baru yang sesuai ada 3 pernyataan dan yang tidak sesuai ada 2 pernyataan.

Dimana penentuan hasil klasifikasi sesuai dan tidak sesuai dari pernyataan tuntutan baru dilihat sumber-sumber hukum di Indonesia yang telah peneliti jabarkan pada **Tabel 5.15**. Cara baca pada **Tabel 5.15** , misalkan untuk percobaan data tuntutan baru pada pasal 36 ayat 3 UU RI No 7 Tahun 2011 memberikan hasil prediksi putusan berdasarkan lama pembedaan kurang dari satu tahun . Hal tersebut kurang sesuai karena seharusnya putusan berdasarkan lama pembedaan adalah lebih dari satu tahun , begitu seterusnya.

Tabel 5.15. Hasil Prediksi *Class* untuk Data Tuntutan Baru

Pasal	Pasal (jo)	Ayat	Butir	Huruf	Sumber	No	Tahun	Tentang	Hasil
36		3			UU RI	7	2011	Mata uang	Lebih dari satu tahun
351		2			KUHP				Lebih dari satu tahun
170		1			KUHP				Lebih dari satu tahun
	170	2	1		KUHP				
1		1			UU Darurat	12	1951	Senjata Api	Kurang dari satu tahun

BAB 6 PENUTUP

6.1 Kesimpulan

Berdasarkan pembahasan pada bab 3, maka diperoleh beberapa kesimpulan dari hasil analisis yang telah dilakukan yaitu:

1. Umumnya, kondisi kasus-kasus pelanggaran pidana di Pengadilan Negeri Sleman Tahun Januari 2016- Januari 2020 adalah sebagai berikut:
 - a. Kasus-kasus pelanggaran pidana di Pengadilan Negeri Sleman terbanyak ada pada tahun 2017 dan paling sedikit ada pada tahun 2020 karena diambil hanya dalam 1 bulan saja. Serta pada 4 tahun terakhir mengalami penurunan yang lumayan signifikan.
 - b. Banyak kasus-kasus pelanggaran pidana di Pengadilan Negeri Sleman di publikasikan. Sekitar 94% dari total kasus tindak pidana yang ada dapat dilihat masyarakat luas.
 - c. Sedangkan untuk kasus-kasus pelanggaran pidana berdasarkan jenis acara pidana paling banyak ada pada jenis acara biasa.
2. Berdasarkan hasil klasifikasi data yang dianalisis menggunakan SVM dengan 4 *kernel* yakni *kernel linear*, RBF, *sigmoid*, dan *polynomial*. Didapatkan hasil yang memiliki *kernel* terbaik adalah *kernel linear* dengan menggunakan parameter $C=0,1$. Sehingga diperoleh nilai akurasi pada *kernel* tersebut sebesar 61.7% dan nilai *error* sebesar 38.3%.
3. Untuk prediksi dengan model yang telah dibuat, didapatkan hasil dari 5 tuntutan baru yang hasilnya sesuai terdapat 3 tuntutan dan yang tidak sesuai terdapat 2 tuntutan.

6.2 Saran

Saran yang dapat diberikan peneliti dari hasil penelitian ini:

1. Untuk penelitian berikutnya, disarankan untuk lebih teliti dalam *input* data, karena *input* data dilakukan secara manual. Sehingga masih ada kesalahan-kesalahan dalam penulisan yang masih masuk ke dalam data variabel tuntutan.
2. Karena dalam *text preprocessing* penelitian ini tidak menghapuskan data angka, disarankan untuk penelitian berikutnya melakukan *input* data angka yang dibutuhkan saja.
3. Menambah data yang dijadikan acuan dalam penelitian, agar mesin semakin terlatih untuk memprediksi atau mengklasifikasi data baru.
4. Menambah wawasan dan memperbanyak referensi mengenai hukum pidana.
5. Untuk menentukan hasil kesesuaian putusan hakim, harus lebih mempelajari sumber-sumber hukum yang ada di Indonesia.
6. Untuk penelitian selanjutnya, dapat di cari langkah untuk penyesuaian dalam menambahkan variabel diluar hasil perhitungan TF-IDF
7. Membuat *chatbot* untuk pihak Pengadilan Negeri, supaya mereka lebih mudah dan mengerti dalam melihat hasil dari putusan yang telah dibuat.

DAFTAR PUSTAKA

Literatur dan Website

- Ahmad, A. (2017). Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning. *Jurnal Teknologi Indonesia*.
- Asikin, Z. (2012). *Pengantar Tata Hukum Indonesia*. Jakarta: PT. Raja Grafindo Persada.
- Badan Diklat Kejaksaan RI. (2019). *Modul Hukum Acara Pidana*. Jakarta.
- Baenanda, L. (2019, Mei 2). *Sejarah dan Perkembangan Revolusi Industri*. Retrieved November 28, 2019, from <https://binus.ac.id/knowledge/2019/05/sejarah-dan-perkembangan-revolusi-industri/>
- Banu Prasetyo, d. (2018). Revolusi Industri 4.0 dan Tantangan Perubahan Sosial. *Strategi Pembangunan Nasional Menghadapi Revolusi Industri 4.0*, 22.
- Christianini, N. d. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge: Cambridge University Press.
- D.L. Poole, A. M. (2010). *Artificial intelligence Foundations of Computational Agents*. New York: Cambridge University Press.
- Dahria, M. (2008). Kecerdasan Buatan. *Jurnal SAINTIKOM VOL.5*, 185.
- Daniel Martin Katz, d. (2017). A General Approach for Predicting the Behavior of the Supreme Court of the United States. *Plos One*.
- Even, Z. (2002). *Introduction to Text Mining*. Retrieved Februari 11, 2020, from <http://www.docstoc.com/docs/25443990/Introduction-to-Text-Mining>
- Fattah, R. (2016). Twitter Text Mining untuk Informasi Gempa Bumi Menggunakan TF-IDF di Indonesia. *Tugas Akhir Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim*. Malang.
- Gruginskie LAdS, V. G. (2018). Lawsuit lead time prediction: Comparison of data mining techniques based on categorical response variable. *Jurnal PLoS ONE 13(6): e0198122*.

- H. Eddy Djunaedi Karnasudirja, S. M. (1983). *Beberapa Pedoman Pemidanaan Dan Pengamatan Narapidana*. Jakarta.
- Hadisuprpto, H. (1976). *Pengantar Hukum Indonesia*. Yogyakarta: Liberty Yogyakarta.
- Hananta, D. (2018). Pertimbangan Keadaan Meringankan dan Memberatkan dalam Penjatuhan Pidana. *Jurnal Hukum dan Peradilan Vol.7 No.1*, 87-100.
- Huda, C. (2006). *Dari Tindak Pidana Tanpa Kesalahan Menuju Kepada Tiada Pertanggungjawaban Pidana Tanpa Kesalahan*. Jakarta: Kencana.
- J. Pustejovsky, A. (2012). *Natural Language Annotation for Machine Learning*. Beijing: O'Reilly.
- J. Yunliang, S. ., (2010). The Classification for E-government Document Based on SVM. In *Web Information Systems and Mining (WISM). International Conference*, (pp. 257-260).
- Johnson, R. A., & Bhattacharyya, G. K. (2010). *Statistics Principles & Methods*. USA: John Wiley & Sons.
- Karnasudirja, H. (1983). *Beberapa Pedoman Pemidanaan Dan Pengamatan Narapidana*. Jakarta.
- Kusumawardani, Q. D. (2019). Hukum Progresif dan Perkembangan Teknologi Kecerdasan Buatan. *VeJ Volume 5 Nomor 1*, 166.
- Laksana, N. C. (2019, Febuari 18). *Apa itu Industri 4.0 dan Bagaimana Indonesia Menyongsongnya*. Retrieved Januari 24, 2020, from <https://www.tek.id/tek/apa-itu-industri-4-0-dan-bagaimana-indonesia-menyongsongnya-b1Xbl9d4L>
- Li, X. (2013). Comparison and Analysis between Holt Exponential Smoothing and Brown Exponential Smoothing Used for Freight Turnover Forecast. *Third International Conference on Intelligent System Design and Engineering Applications* (pp. 453-456). IEEE.
- Lucia Adriana dos Santos Gruginskie, d. (2018). Lawsuit Lead Time Prediction: Comparison of Data Mining Techniques Based on Categorical Response Variable. *Plos One*.

- Mahkamah Agung. (2016). *Profil Mahkamah Agung Republik Indonesia*. Jakarta: SEKMA-RI.
- Marcos Zampieri, d. (2017). Predicting the Law Area and Decisions of French Supreme Court Cases. *Proceedings of Recent Advances in Natural Language Processing*, (pp. 716-722). Bulgaria.
- Masha Medvedeva, d. (2019). Using Machine Learning to predict decisions of the European Court of Human Rights. *Artificial Intelligence and Law*.
- McCarthy, J. (1956). *Artificial Intelligence*.
- Moeljanto. (1993). *Asas-Asas Hukum Pidana*. Jakarta: PT. Rineka Cipta.
- Nugroho, A. S. (2003). *Pengantar Support Vector Machine*.
- Philipus, d. (2011). *Pengantar Hukum Administrasi Indonesia*. Yogyakarta: Gadjah Mada University Press.
- Prof Priyanka Bhilare, d. (2019). Predicting Outcome of Judicial Cases and Analysis Using Machine Learning. *International Research Journal of Engineering and Technology (IRJET)*, 326-330.
- R. Feldman, J. S. (2007). *The Text Mining Handbook: Advanced to Analysing Unstructured Data*. Cambridge: Cambridge University Press.
- R. Soesilo. (1991). *KUHP serta Komentar-Komentar Lengkap Pasal Demi Pasal*. Bogor: Politeia.
- Rainer Drath, A. H. (2014). Industrie 4.0: Hit or Hype? [Industry Forum]. *IEEE Industrial Electronics Magazine*, 56-58.
- RI, M. A. (2016). *Profil Mahkamah Agung Republik Indonesia*. Jakarta: SEKMA-RI.
- Russell, S. (2015). *Artificial Intelligence: A Modern Approach*. Pearson.
- Rusydiana, A. S. (2016, September 1). *Empat Tahap Proses Text Mining*. Retrieved Februari 11, 2020, from <http://textmining-center.blogspot.com/2016/09/4-tahap-proses-text-mining.html>
- S. Sanjaya, E. A. (2015). Pengelompokan Dokumen Menggunakan Winnowing Fingerprint dengan Metode K-Nearest Neighbour. *Jurnal CoreIT*, 50-56.

- Said, B. (2016). Klasifikasi dan Analisis Sentimen Data SMS Center Bupati Pamekasan Menggunakan Naive Bayes dengan Mad Smoothing. *Jurnal Link Vol.25*, 3.
- Santosa, B. (2007). *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu.
- Sari, D. P. (2010). Implementasi Tugas Hakim Pengawas dan Pengamat Pengawasan dan Pengamatan Terhadap Narapidana. *Dinamika Hukum Vol 10 No 2*.
- Simanjuntak, M. P. (2010). Disparitas Pidana Terhadap Pelaku Tindak Pidana Pembunuhan di Wilayah Hukum Pengadilan Negeri Sleman. *Tugas Akhir Fakultas Hukum Universitas Atma Jaya Yogyakarta*.
- Staelin, C. (2003). Parameter Selection for Support Vector Machine. *HP Laboratories Israel*.
- Sugiyono. (2009). *Metode Penelitian Kuantitatif, Kualitatif dan R&D*. Bandung: Alfabeta.
- Suhartono, D. (2013, Juni 22). *Natural Language Processing*. Retrieved Febuari 10, 2020, from <https://socs.binus.ac.id/2013/06/22/natural-language-processing/>
- Thalib, A. K. (2018). Analisis Klasifikasi Hoax pada Unstructured Data Text di Situs Portal Berita Detik.com dan Turnbackhoax.id. *Tugas Akhir Jurusan Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Islam Indonesia*.
- Utrecht, E. (1966). *Pengantar dalam Hukum Indonesia*. Penerbit Universitas.
- Walpole, R. E., Myers, R. H., Myers, S. L., & Ye, K. (2011). *Probability & Statistics for Engineers & Scientists 9th Ed*. USA: Pearson.
- Yunliang, Q. . (2010). The Classification for E-government Document Based on SVM. In Web Information Systems and Mining (WISM). *International Conference*, (pp. 257-260).

Perundang-Undangan

Pasal 1 Angka 11 Kitab Undang-Undang Hukum Pidana (KUHP).

Pasal 10 KUHP.

Pasal 12 KUHP.

Pasal 18 KUHP.

Pasal 30 Ayat 4 KUHP.

Pasal 35 Ayat 1 KUHP.

Pasal 39 KUHP.

Pasal 152-202 KUHAP.

Pasal 203 Ayat 1 KUHAP.

Pasal 205 Ayat 1 KUHAP.

Undang-Undang No.8 Tahun 1981 tentang Hukum Acara Pidana.

Pasal 1 butir 8 Kitab Undang-Undang Hukum Acara Pidana(KUHAP).

LAMPIRAN

Lampiran 1 Data Penelitian Semua Kasus Pelanggaran Pidana di PN Sleman

No	Nomor Pe	Tgl Registrasi	Klasifikasi	Penuntut	Terdakwa	Status Per	Lama Pros	Jenis Perke	Dakwaan	Pihak Dipu	Penuntuta	Putusan	Putusan	Klasifikasi	Lama Kurungan
1	1	17/Pid.S/20	28-May-19	Lain-lain	Erlin Yulias Muhamma	Minutasi	1	Singkat	Pertama P Ya		Menuntut Supaya Ma	Mengadili	Menyalahg	2 tahun	
2	2	1/Pid.Sus/	14-Jun-16	Lain-lain	Yogie Rahi Dahlan bin	Minutasi	1	Singkat	Bahwa Ter Ya		Menuntut Supaya Ma	Mengadili	Orang asin	Subsida	
3	9	134/Pid.C/	06-Dec-19	Pelanggar	Pambudi P Ria Fitriyal	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Melakukan	Subsida	
4	10	135/Pid.C/	06-Dec-19	Pelanggar	Pambudi P Rina Purwi	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Melakukan	Subsida	
5	11	136/Pid.C/	06-Dec-19	Pelanggar	Pambudi P Haryono	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Melakukan	Subsida	
6	12	137/Pid.C/	06-Dec-19	Pelanggar	Pambudi P Turyani	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Melakukan	Subsida	
7	13	138/Pid.C/	06-Dec-19	Pelanggar	Supriyanto Dwi Nurwi	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Mengecer	Subsida	
8	14	129/Pid.C/	22-Nov-19	Pelanggar	Suwarda Tri Diono	Minutasi	1	Cepat	Bahwa pac Ya			Mengadili	Menyatakan	Terdakwa	
9	15	130/Pid.C/	22-Nov-19	Pelanggar	Haryata S Bondan Ce	Minutasi	1	Cepat	Dalam Per Ya			Mengadili	Menyatakan	Terdakwa	
10	16	131/Pid.C/	22-Nov-19	Pelanggar	Suharyana Rusman Ye	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Menyatakan	Terdakwa	
11	17	132/Pid.C/	22-Nov-19	Pelanggar	Ardhina Re Rusman Ye	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Menyatakan	Terdakwa	
12	18	133/Pid.C/	22-Nov-19	Pelanggar	Florentia S Yuliana Kri	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Menyatakan	Terdakwa	
13	19	124/Pid.C/	15-Nov-19	Pelanggar	Pambudi P Stefanus A	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Menyimpa	1 bulan	
14	20	125/Pid.C/	15-Nov-19	Pelanggar	Pambudi P Damar Ag	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Menyimpa	3 hari	
15	21	126/Pid.C/	15-Nov-19	Pelanggar	Pambudi P Agustinus	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Menyimpa	1 bulan	
16	22	127/Pid.C/	15-Nov-19	Pelanggar	Triyanto Eddy Wijar	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Menyimpa	3 hari	
17	23	128/Pid.C/	15-Nov-19	Pelanggar	Triyanto St. Purnom	Minutasi	1	Cepat	Pada hari i Ya						
18	24	121/Pid.C/	08-Nov-19	Pelanggar	Agus Santc Marchella	Minutasi	1	Cepat	Pelanggar Ya			Mengadili	Mendirikai	3 hari	
19	25	122/Pid.C/	08-Nov-19	Pelanggar	Agus Santc Indriyani	Minutasi	1	Cepat	Pelanggar Ya			Mengadili	Mendirikai	3 hari	
20	26	123/Pid.C/	08-Nov-19	Pelanggar	Agus Santc Arisah Nur	Minutasi	1	Cepat	Pelanggar Ya			Mengadili	Mendirikai	3 hari	
21	27	117/Pid.C/	25-Oct-19	Pelanggar	Pambudi P Aryani	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Pengusaha	3 hari	
22	28	118/Pid.C/	25-Oct-19	Pelanggar	Pambudi P Indriyani	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Pengusaha	3 hari	
23	29	119/Pid.C/	25-Oct-19	Pelanggar	Pambudi P Septi Wira	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Pengusaha	3 hari	
24	30	120/Pid.C/	25-Oct-19	Pelanggar	Ansori Khoirun Ni	Minutasi	1	Cepat	Dalam per Ya			Mengadili	Pasal 5 Jo	5 bulan	
25	31	114/Pid.C/	04-Oct-19	Pelanggar	Pratiwanti Danang St	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Pasal 37 A	2 bulan	
26	32	115/Pid.C/	04-Oct-19	Pelanggar	Pratiwanti Edi Supom	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Pasal 37 A	2 bulan	
27	33	116/Pid.C/	04-Oct-19	Pelanggar	Pratiwanti Sulisty H	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Pasal 37 A	1 bulan 15 hari	
28	34	112/Pid.C/	20-Sep-19	Pelanggar	Alvin Aji K Eko Hapri	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Menyimpa	1 bulan	
29	35	113/Pid.C/	20-Sep-19	Pelanggar	Pratiwanti Suharyanti	Minutasi	1	Cepat	Pada hari i Ya			Mengadili	Mendirikai	1 bulan	
30	36	111/Pid.C/	13-Sep-19	Pelanggar	Radhitya J Petar Celic	Minutasi	1	Cepat	Melanggar Ya			Mengadili	Tidak mele	7 hari	
31	37	100/Pid.C/	06-Sep-19	Pelanggar	Ardhina R Multi Sula	Minutasi	1	Cepat	Melanggar Ya			Mengadili	Mengadili	1 bulan	

Untuk data lengkap, dapat membuka link dibawah ini: http://tiny.cc/tugas_akhir

Lampiran 2 Script dan Output

```
!apt-get install pandoc
```

```
'apt-get' is not recognized as an internal or external command,  
operable program or batch file.
```

```
import pandas as pd  
data = pd.read_excel('datatuntutan.xlsx')  
data.head()
```

	tuntutan	klasifikasi	putusan
0	Menuntut Menyatakan mereka terdakwa 1. WALIJO,...	Tanpa Hak Dengan Sengaja Memberi Kesempatan Ke...	kurang dari satu tahun
1	Menuntut Menyatakan mereka terdakwa 1. WALIJO,...	Tanpa Hak Dengan Sengaja Memberi Kesempatan Ke...	kurang dari satu tahun
2	Menuntut Supaya Majelis Hakim Pengadilan Neger...	Tanpa Hak Dengan Sengaja Memberi Kesempatan Ke...	kurang dari satu tahun
3	Menuntut Supaya Majelis Hakim Pengadilan Neger...	Tanpa Hak Dengan Sengaja Memberi Kesempatan Ke...	kurang dari satu tahun
4	Menuntut Menyatakan terdakwa Rahmat Hidayat Al...	Kekerasan terhadap orang yang menyebabkan oran...	lebih dari satu tahun

```
#melihat jumlah baris dan kolom
data.shape
```

```
(2248, 3)
```

```
col = ['putusan', 'tuntutan']
data = data[col]
data.columns
```

```
Index(['putusan', 'tuntutan'], dtype='object')
```

```
from io import StringIO
col = ['putusan', 'tuntutan']
data = data[col]
data = data[pd.notnull(data['tuntutan'])]

data.columns = ['putusan', 'tuntutan']
data['putusan_id'] = data['putusan'].factorize()[0]
putusan_id_data = data[['putusan', 'putusan_id']].drop_duplicates().sort_values('putusan_id')
putusan_to_id = dict(putusan_id_data.values)
id_to_putusan = dict(putusan_id_data[['putusan_id', 'putusan']].values)
data.head()
```

	putusan	tuntutan	putusan_id
0	kurang dari satu tahun	Menuntut Menyatakan mereka terdakwa 1. WALIJO,...	0
1	kurang dari satu tahun	Menuntut Menyatakan mereka terdakwa 1. WALIJO,...	0
2	kurang dari satu tahun	Menuntut Supaya Majelis Hakim Pengadilan Neger...	0
3	kurang dari satu tahun	Menuntut Supaya Majelis Hakim Pengadilan Neger...	0
4	lebih dari satu tahun	Menuntut Menyatakan terdakwa Rahmat Hidayat Al...	1

```
data['putusan'].value_counts()
```

```
kurang dari satu tahun    1293
lebih dari satu tahun     954
Name: putusan, dtype: int64
```

```
print("Number of null in label: {}".format(data['putusan'].isnull().sum()))
print("Number of null in text: {}".format(data['tuntutan'].isnull().sum()))
```

```
Number of null in label: 0
Number of null in text: 0
```

```
#Case Folding
data['tuntutan'] = [entry.lower() for entry in data['tuntutan']]
data['tuntutan'].head()
```

```
0    menuntut menyatakan mereka terdakwa 1. walijo,...
1    menuntut menyatakan mereka terdakwa 1. walijo,...
2    menuntut supaya majelis hakim pengadilan neger...
3    menuntut supaya majelis hakim pengadilan neger...
4    menuntut menyatakan terdakwa rahmat hidayat al...
Name: tuntutan, dtype: object
```

```
# Removing Punctuation
import string
string.punctuation

#Remove Punctuation
def remove_punct(text):
    text_nopunct = "".join([char for char in text if char not in string.puncti
    return text_nopunct

data['tuntutan'] = data['tuntutan'].apply(lambda x: remove_punct(x))
data['tuntutan'].head()
```

```
0    menuntut menyatakan mereka terdakwa 1 walijo b...
1    menuntut menyatakan mereka terdakwa 1 walijo b...
2    menuntut supaya majelis hakim pengadilan neger...
3    menuntut supaya majelis hakim pengadilan neger...
4    menuntut menyatakan terdakwa rahmat hidayat al...
Name: tuntutan, dtype: object
```

```
#Link https://github.com/har07/PySastrawi
!pip install Sastrawi
```

```
Requirement already satisfied: Sastrawi in c:\users\lenovo\appdata\local\pr
ograms\python\python36\lib\site-packages (1.0.1)
```

```
WARNING: You are using pip version 20.0.1; however, version 20.0.2 is avail
able.
```

```
You should consider upgrading via the 'c:\users\lenovo\appdata\local\progra
ms\python\python36\python.exe -m pip install --upgrade pip' command.
```

```
#stopword sastrawi di link 'https://devtrik.com/python/stopword-removal-bah
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverF

factory = StopWordRemoverFactory()
stopwords = factory.get_stop_words()
print(stopwords)
```

```
['yang', 'untuk', 'pada', 'ke', 'para', 'namun', 'menurut', 'antara', 'di
a', 'dua', 'ia', 'seperti', 'jika', 'jika', 'sehingga', 'kembali', 'dan',
'tidak', 'ini', 'karena', 'kepada', 'oleh', 'saat', 'harus', 'sementara',
'setelah', 'belum', 'kami', 'sekitar', 'bagi', 'serta', 'di', 'dari', 'tela
h', 'sebagai', 'masih', 'hal', 'ketika', 'adalah', 'itu', 'dalam', 'bisa',
'bahwa', 'atau', 'hanya', 'kita', 'dengan', 'akan', 'juga', 'ada', 'merek
a', 'sudah', 'saya', 'terhadap', 'secara', 'agar', 'lain', 'anda', 'begit
u', 'mengapa', 'kenapa', 'yaitu', 'yakni', 'daripada', 'itulah', 'lagi', 'm
aka', 'tentang', 'demi', 'dimana', 'kemana', 'pula', 'sambil', 'sebelum',
'sesudah', 'supaya', 'guna', 'kah', 'pun', 'sampai', 'sedangkan', 'selagi',
'sementara', 'tetapi', 'apakah', 'kecuali', 'sebab', 'selain', 'seolah', 's
eraya', 'seterusnya', 'tanpa', 'agak', 'boleh', 'dapat', 'dsb', 'dst', 'dl
l', 'dahulu', 'dulunya', 'anu', 'demikian', 'tapi', 'ingin', 'juga', 'ngga
k', 'mari', 'nantinya', 'melainkan', 'oh', 'ok', 'seharusnya', 'sebetulnya',
'setiap', 'setidaknya', 'sesuatu', 'pasti', 'saja', 'toh', 'ya', 'walau',
'tolong', 'tentu', 'amat', 'apalagi', 'bagaimanapun']
```

```
# import StopWordRemoverFactory class
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverF

factory = StopWordRemoverFactory()
stopword = factory.get_stop_words()

# Kalimat
data['tuntutan'] = data['tuntutan'].apply(lambda x: " ".join(x for x in x.sp
data['tuntutan'].head()
```

```
0    menuntut menyatakan terdakwa 1 walijo bersama ...
1    menuntut menyatakan terdakwa 1 walijo bersama ...
2    menuntut majelis hakim pengadilan negeri slema...
3    menuntut majelis hakim pengadilan negeri slema...
4    menuntut menyatakan terdakwa rahmat hidayat al...
Name: tuntutan, dtype: object
```

```
import requests
def stopwords():
    r = requests.get("https://raw.githubusercontent.com/masdevid/ID-Stopword
    data = []
    for x in r.split("\n"):
        data.append(x)
    return data
```

```
stopwords()
```

```
['ada',  
'adalah',  
'adanya',  
'adapun',  
'agak',  
'agaknya',  
'agar',  
'akan',  
'akankah',  
'akhir',  
'akhiri',  
'akhirnya',  
'aku',  
'akulah',  
'amat',  
'amatlah',  
'anda',  
'andalah',  
'antar',  
'antara'
```

```
# Import Stopword Factory class  
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory  
  
#Create factory  
factory = StopWordRemoverFactory()  
more_stopword = ['nya', 'kali', 'sih', 'yg', 'pengadilan', 'negeri', 'majelis', 'hal']  
stopwordplus = factory.get_stop_words()+stopwords()+more_stopword  
  
data['tuntutan'] = data['tuntutan'].apply(lambda x: " ".join(x for x in x.split() if x not in stopwordplus))  
data['tuntutan']
```

```
0      1 walijo 2 ferdyanto terbukti bersalah perjudi...  
1      1 walijo 2 ferdyanto terbukti bersalah perjudi...  
2      1 widiyono kertoarjo 2 saptana harjo suwito te...  
3      1 widiyono kertoarjo 2 saptana harjo suwito te...  
4      rahmat hidayat kencut bahrudin terbukti sah be...  
      ...  
2243   agus pri utomo agus wahyudi terbukti sah bersa...  
2244   anak batam iabdul aziz aziz agus maryono ii tr...  
2245   1pardiyo 2 didik susilo bersalah bersamasama i...  
2246   1 andrias dwi kurniawan sim sim bersalah "deng...  
2247   i agus david riyanto david dragon ii eko nurca...  
Name: tuntutan, Length: 2247, dtype: object
```

stopwordplus

```
['yang',  
'untuk',  
'pada',  
'ke',  
'para',  
'namun',  
'menurut',  
'antara',  
'dia',  
'dua',  
'ia',  
'seperti',  
'jika',  
'jika',  
'sehingga',  
'kembali',  
'dan',  
'tidak',  
'ini',  
'karena'
```

```
import re  
# Function to Tokenize words  
def tokenize(text):  
    tokens = re.split('\W+', text) #W+ means that either a word character (A  
    return tokens  
  
data['tuntutan'] = data['tuntutan'].apply(lambda x: tokenize(x.lower()))  
#We convert to lower as Python is case-sensitive.  
data.head()
```

	putusan	tuntutan	putusan_id
0	kurang dari satu tahun	[1, walijo, 2, ferdyanto, terbukti, bersalah, ...	0
1	kurang dari satu tahun	[1, walijo, 2, ferdyanto, terbukti, bersalah, ...	0
2	kurang dari satu tahun	[1, widiyono, kertoarjo, 2, saptana, harjo, su...	0
3	kurang dari satu tahun	[1, widiyono, kertoarjo, 2, saptana, harjo, su...	0
4	lebih dari satu tahun	[rahmat, hidayat, kencut, bahrudin, terbukti, ...	1

```
from xlsxwriter.utility import xl_rowcol_to_cell  
saveresult = pd.ExcelWriter('tuntutanpreprocessing.xlsx', engine='xlsxwriter')  
data.to_excel(saveresult, index=False, sheet_name='report')  
saveresult.save()
```

```
data.to_csv("tuntutanpreprocessing.csv", sep=',')
import pandas as pd
data = pd.read_csv('tuntutanpreprocessing.csv')
data.head()
```

Unnamed: 0	putusan	tuntutan	putusan_id
0	0 kurang dari satu tahun	['1', 'waliyo', '2', 'ferdyanto', 'terbukti', ...	0
1	1 kurang dari satu tahun	['1', 'waliyo', '2', 'ferdyanto', 'terbukti', ...	0
2	2 kurang dari satu tahun	['1', 'widiyono', 'kertoarjo', '2', 'saptana', ...	0
3	3 kurang dari satu tahun	['1', 'widiyono', 'kertoarjo', '2', 'saptana', ...	0
4	4 lebih dari satu tahun	['rahmat', 'hidayat', 'kencut', 'bahrudin', 't...	1

TF-IDF

```
from sklearn.feature_extraction.text import CountVectorizer
count_vectorizer = CountVectorizer(encoding='latin-1', ngram_range=(1, 1), t
countvec = count_vectorizer.fit_transform(data.tuntutan).toarray()
countvec
```

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

```
countvec2 = pd.DataFrame(countvec)
countvec2
```

	0	1	2	3	4	5	6	7	8	9	...	21779	21780	21781	21782	21783	21784	21785	
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
...
2242	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
2243	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
2244	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
2245	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
2246	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0

2247 rows × 21789 columns

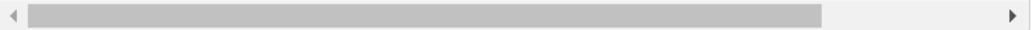

```
kata_kata = count_vectorizer.get_feature_names()
countvec3 = pd.DataFrame(countvec, columns=kata_kata)
countvec3
```

	00	000	0000	000000	000000000	000004	000005	000006	000014	0001skdirbprrimi20
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
...
2242	0	0	0	0	0	0	0	0	0	0
2243	0	0	0	0	0	0	0	0	0	0
2244	0	0	0	0	0	0	0	0	0	0
2245	0	0	0	0	0	0	0	0	0	0
2246	0	0	0	0	0	0	0	0	0	0

2247 rows × 21789 columns



```
from sklearn.feature_extraction.text import TfidfTransformer
transformer = TfidfTransformer(norm=None, use_idf=True, smooth_idf=False, sublinear_tf=False)
tfidf = transformer.fit_transform(countvec)
tfidf
```



<2247x21789 sparse matrix of type '<class 'numpy.float64''>
with 164784 stored elements in Compressed Sparse Row format>

```
tfidf1 = tfidf.toarray()
tfidf1
```

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

```
tfidf2 = pd.DataFrame(tfidf1)
tfidf2
```

	0	1	2	3	4	5	6	7	8	9	...	21779	21780	21781	21782	21783	2
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
...	
2242	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
2243	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
2244	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
2245	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
2246	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	

2247 rows × 21789 columns



```
kata_kata2 = count_vectorizer.get_feature_names()
df1 = pd.DataFrame(tfidf1, columns=kata_kata2)
df1
```

	00	000	0000	000000	000000000	000004	000005	000006	000014	0001skdirbprrimi2
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
...	
2242	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2243	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2244	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2245	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2246	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

2247 rows × 21789 columns



```
result = pd.concat([data['putusan_id'],df1], axis=1)
result
```

	putusan_id	00	000	0000	000000	000000000	000004	000005	000006	000014	...	z
0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
1	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
2	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
3	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
4	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
...
2242	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
2243	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
2244	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
2245	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
2246	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	

2247 rows × 21790 columns



```
y = result['putusan_id']
y
```

```
0      0
1      0
2      0
3      0
4      1
..
2242   0
2243   0
2244   0
2245   1
2246   1
```

Name: putusan_id, Length: 2247, dtype: int64

```
##### KLASIFIKASI SVM #####
```

```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    df1, y , test_size=0.2, random_state=0
)
```

```

from sklearn.metrics import confusion_matrix
from sklearn.svm import SVC

def train_svm(X_train, y_train):
    """
    Create and train the Support Vector Machine.
    """
    svm = SVC(C=0.1, kernel='linear')
    svm.fit(X_train, y_train)
    return svm
# Create and train the Support Vector Machine

svm = train_svm(X_train, y_train)

# Make an array of predictions on the test set
pred = svm.predict(X_test)

# Output the hit-rate and the confusion matrix for each model
print(svm.score(X_test, y_test))
print(confusion_matrix(pred, y_test))

```

```

0.6177777777777778
[[192  97]
 [ 75  86]]

```

```

from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(norm=None, encoding='latin-1', ngram_range=(1, 1), stop_words=None)

features = tfidf.fit_transform(data.tuntutan).toarray()
labels = data.putusan_id
features.shape

```

```
pd.DataFrame(features)
```

	0	1	2	3	4	5	6	7	8	9	...	21803	21804	21805	21806	21807	21808
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
...
2242	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
2243	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
2244	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
2245	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
2246	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0

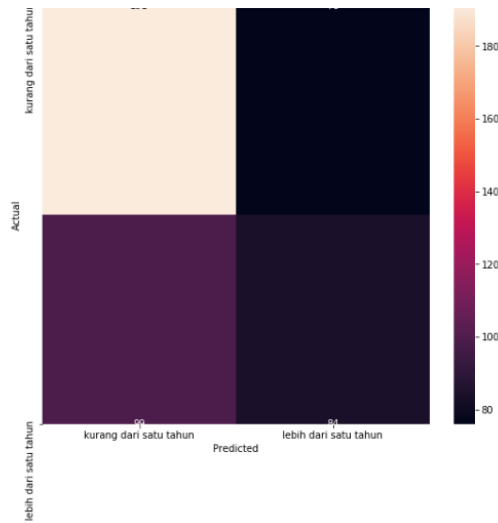
2247 rows × 21813 columns



```
tfidf.get_feature_names()
```

```
['00',  
'000',  
'0000',  
'000000',  
'00000000',  
'000004',  
'000005',  
'000006',  
'000014',  
'0001skdirbprrimi2011',  
'00027',  
'000284',  
'0003',  
'00030',  
'000303',  
'000305',  
'000307',  
'000308',  
'000311',  
'000330']
```

```
from sklearn.model_selection import train_test_split  
from sklearn.svm import LinearSVC  
import matplotlib.pyplot as plt  
from sklearn.metrics import confusion_matrix  
import seaborn as sns  
  
model = SVC(C=0.1, kernel='linear')  
X_train, X_test, y_train, y_test, indices_train, indices_test = train_test_split(X, y, test_size=0.2, random_state=42)  
model.fit(X_train, y_train)  
y_pred = model.predict(X_test)  
  
conf_mat = confusion_matrix(y_test, y_pred)  
fig, ax = plt.subplots(figsize=(9,8))  
sns.heatmap(conf_mat, annot=True, fmt='d',  
            xticklabels=putusan_id_data.putusan.values, yticklabels=putusan_id_data.putusan.values,  
            plt.ylabel('Actual')  
            plt.xlabel('Predicted'))
```



```

from IPython.display import display

for predicted in putusan_id_data.putusan_id:
    for actual in putusan_id_data.putusan_id:
        if predicted != actual and conf_mat[actual, predicted] >= 20:
            print("{} predicted as {}: {} examples.".format(id_to_putusan[actual], id_to_putusan[predicted], conf_mat[actual, predicted]))
            display(data.loc[indices_test[(y_test == actual) & (y_pred == predicted)]])
            print('')

```

'lebih dari satu tahun' predicted as 'kurang dari satu tahun' : 99 examples.

	putusan	tuntutan
596	lebih dari satu tahun	['tawar', 'siswanto', 'alm', 'somodiranbersala...
1440	lebih dari satu tahun	['sumaryanto', 'peyok', 'sutiman', 'ervan', 'd...
995	lebih dari satu tahun	['bakti', 'aji', 'kapindo', 'subarman', 'terbu...
1307	lebih dari satu tahun	['eko', 'wijayanto', 'andin', 'alm', 'sunardi'...
930	lebih dari satu tahun	['krisna', 'efendi', 'alm', 'subandi', 'terbuk...
...
2134	lebih dari satu tahun	['junaidi', 'ardiansyah', 'bersalah', 'pencuri...
1518	lebih dari satu tahun	['tin', 'kartini', 'tin', 'binti', 'jawahir', ...
1934	lebih dari satu tahun	['andreas', 'arif', 'purnomo', 'puryono', 'ter...
1510	lebih dari satu tahun	['i', 'emil', 'rachmand', 'renald', 'christian...
453	lebih dari satu tahun	['suwito', 'toto', 'rejo', 'suwito', 'terbukti...

99 rows x 2 columns

'kurang dari satu tahun' predicted as 'lebih dari satu tahun' : 76 examples.

	putusan	tuntutan
2060	kurang dari satu tahun	['yohanes', 'dwi', 'antoro', 'alm', 'petrus', ...
461	kurang dari satu tahun	['alex', 'suyono', 'karsidi', 'bersalah', 'pen...
2091	kurang dari satu tahun	['yulianto', 'nonong', 'mujiyo', 'terbukti', '...
10	kurang dari satu tahun	['nunung', 'tri', 'susanto', 'bersalah', 'seng...
39	kurang dari satu tahun	['adrianus', 'candra', 'sujatmiko', 'amd', 'te...
...
2043	kurang dari satu tahun	['eko', 'noviyanto', 'tilis', 'suratmin', 'ber...
838	kurang dari satu tahun	['unggul', 'desantara', 'atiardi', 'sardijono'...
1968	kurang dari satu tahun	['wisnu', 'dwipurwo', 'jiwandono', 'gundol', '...
713	kurang dari satu tahun	['hartati', 'syanti', 'binti', 'syamsu', 'alam...
9	kurang dari satu tahun	['agus', 'sumanta', 'antok', 'bersalah', 'seng...

76 rows × 2 columns

```
##### PREDIKSI DATA #####
```

```
from sklearn.model_selection import train_test_split
from sklearn.svm import LinearSVC
from sklearn import svm
import sklearn
from sklearn.svm import SVC

model = LinearSVC(C=0.1)
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_s
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

```
model.fit(features, labels)
```

```
LinearSVC(C=0.1, class_weight=None, dual=True, fit_intercept=True,
intercept_scaling=1, loss='squared_hinge', max_iter=1000,
multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
verbose=0)
```

```
texts = ["1) Menuntut supaya majelis hakim pengadilan negeri sleman yang mem
"2) Menyatakan terdakwa alibobo bersalah melakukan tindak penggelapan
"3) Menyatakan terdakwa agus telah bersalah melakuka tindak pidana y
"4) Menuntut Supaya majelis hakim pengadilan negeri sleman memeriksa
"5)menuntut Menyatakan Terdakwa agus telah terbukti secara sah dan i
text_features = tfidf.transform(texts)
predictions = model.predict(text_features)
for text, predicted in zip(texts, predictions):
    print("{} {}".format(text))
    print(" - Prediksi: {}".format(id_to_putusan[predicted]))
    print("")
```

"1) Menuntut supaya majelis hakim pengadilan negeri sleman yang memeriksa dan mengadili perkara ini memutuskan : menyatakan terdakwa sabrino tralulu secara sah dan meyakinkan telah terbukti bersalah melakukan tindak pidana pasal 36 ayat 3 UU RI no 7 tahun 2011 tentang mata uang"

- Prediksi: 'kurang dari satu tahun'

"2) Menyatakan terdakwa alibobo bersalah melakukan tindak penggelapan"

- Prediksi: 'kurang dari satu tahun'

"3) Menyatakan terdakwa agus telah bersalah melakukan tindak pidana yang diatur dalam pasal 351 ayat 2 KUHP"

- Prediksi: 'kurang dari satu tahun'

"4) Menuntut Supaya majelis hakim pengadilan negeri sleman memeriksa dan mengadili perkara ini memutuskan : menyatakan terdakwa barbie bersalah melakukan tindak pidana tanpa hak menyimpan senjata api, sebagaimana diatur dalam pasal 1 ayat 1 UU Darurat no 12 Tahun 1951 tentang senjata api dan bahan peledak sebagaimana dakwaan Penuntut Umum"

- Prediksi: 'kurang dari satu tahun'

"5)menuntut Menyatakan Terdakwa agus telah terbukti secara sah dan meyakinkan bersalah melakukan tindak pidana dengan terang-terangan dan dengan tenaga bersama menggunakan kekerasan terhadap orang yang mengakibatkan luka-luka sebagaimana di maksud dalam Pasal 170 ayat (1) jo. Ayat (2) ke-1 KUHP . Menjatuhkan pidana terhadap Terdakwa agus dengan pidana penjara masing-masing selama 8 bulan dengan perintah agar Para Terdakwa tetap ditahan . Dirampas untuk dimusnahkan Menetapkan agar Para terdakwa dibebani membayar biaya perkara masing-masing sebesar Rp. 2.000,- (dua ribu rupiah); Terhadap tuntutan pidana tersebut, atas pertanyaan "

- Prediksi: 'lebih dari satu tahun'

```
from sklearn import metrics
print(metrics.classification_report(y_test, y_pred,
                                   target_names=data['putusan'].unique()))
```

	precision	recall	f1-score	support
kurang dari satu tahun	0.66	0.73	0.69	267
lebih dari satu tahun	0.54	0.46	0.49	183
accuracy			0.62	450
macro avg	0.60	0.59	0.59	450
weighted avg	0.61	0.62	0.61	450

Untuk *script* dan *output* yang lengkap, dapat membuka link dibawah ini:

http://tiny.cc/tugas_akhir