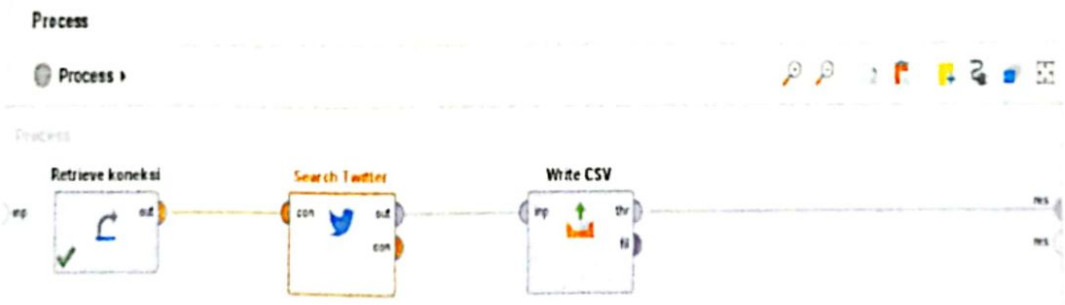
BAB IV
HASIL PENELITIAN DAN PEMBAHASAN

4.1. Hasil Penelitian

4.1.1. Tahap Pengumpulan Data

Data penelitian yang diambil (*crawling*) berasal dari media sosial Twitter yang diambil dari bulan Januari sampai Juni 2023. Menggunakan model yang telah dibuat dengan RapidMiner diperoleh data sebanyak 10.248 data tweet dengan kata kunci pencarian Perpu Cipta Kerja. Tweet yang dikumpulkan adalah tweet berbahasa Indonesia yang berasal dari region Indonesia. Adapun model pengumpulan data dengan RapidMiner sebagaimana gambar 4.1. berikut ini.



Gambar 4. 1. Proses *Crawling* Data Twitter

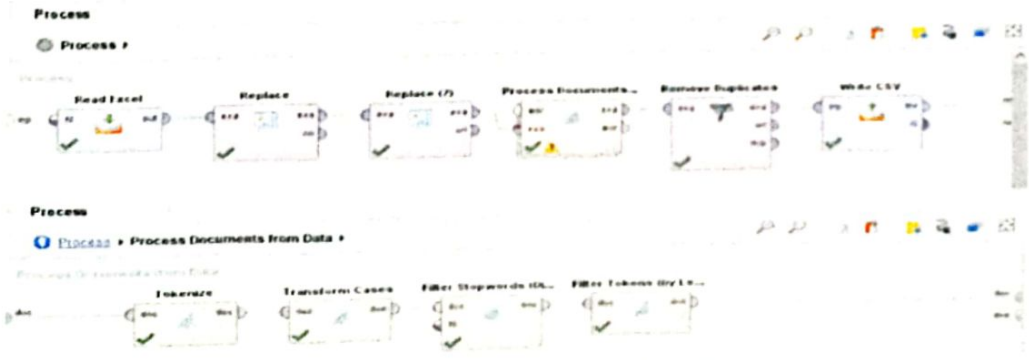
Crawling diawali dengan membuat koneksi twitter. Untuk membuat koneksi ini harus memiliki akun Twitter dan login ke Twitternya untuk memperoleh *Acces Token*. Setelah tahapannya dilalui dan koneksi sudah terbentuk, kemudian dihubungkan operator *Search Twitter* pada RapidMiner. Pada pojok kanan atas panel parameter *search twitter* dilakukan pengaturan untuk memasukkan kata kunci yang akan dicari, jumlah twitter yang diharapkan, waktu

yang ditentukan, lokasi, dan sebagainya. *Crawling* menggunakan RapidMiner dilakukan secara bertahap mengingat terdapat batasan dalam melakukan *crawling*. Penerapannya pada data twitter, *crawling* menggunakan *automation* program dan juga *Application Programming Interface* (API) sebagai jalur komunikasi dalam mendapatkan data. Dengan API ini, data yang lebih spesifik dapat dikumpulkan sesuai dengan kata kunci yang diinput tanpa harus mengetahui elemen HTML pada sebuah website.

Menggunakan model pada gambar 4.1, diperoleh data sebanyak 10.248 data tweet yang kemudian disimpan dalam format .csv. Sebelum dilakukan proses pemodelan klasifikasi, dilakukan pra-pemrosesan data, yang meliputi *case folding*, *tokenisasi*, penghapusan *stopword*, *stemming*, dan pembobotan kata menggunakan metode TF-IDF yang juga dilakukan menggunakan *software* RapidMiner.

4.1.2. Tahap Pengolahan Data (*Preprocessing Data*)

Data hasil *crawling* kemudian dilakukan *preprocessing*. Pada tahapan ini dilakukan beberapa proses dengan model sebagaimana gambar 4.2. berikut.



Gambar 4. 2. Proses Pengolahan Data

Berdasarkan gambar 4.2 di atas, data yang disimpan dalam format CSV atau *Excel* dilakukan pembersihan data (*data cleaning*) dan menghilangkan karakter yang tidak berhubungan dengan topik, seperti HTML link, username, mention, hastag, retweet, angka, tanda baca, spasi berlebih seperti („"~&?!><#%{}([0-9]+:;") menggunakan operator *Replace.* Pada sub proses operator *Process Document From Data* dipilih *vector creation* nya TF-IDF. Di dalam sub process ini terdapat beberapa operator yaitu *Tokenize, Transform Cases, Filter Stopword,* dan *Filter Tokenz (by length). Tokenize* merupakan operator untuk membagi teks dokumen menjadi urutan token. *Transform Cases*, operator ini mengubah semua karakter dalam dokumen menjadi huruf kecil atau huruf besar. *Filter Stopword* (Bahasa Indonesia), fitur ini untuk menghilangkan teks yang tidak berhubungan dengan Analisa sentimen tanpa mengurangi isi sentimen. *Operator Filter Tokenz (by length)* yaitu untuk membatasi dengan minimal 4 karakter dan maksimal 25 karakter yang akan diproses.

Hasil contohnya sebagaimana tabel 4.1. berikut.

Tabel 4. 1. Contoh Hasil *Data Cleaning*

Sebelum	Sesudah
Soal UU Perampasan Aset? Jika memang Presiden Jokowi memandang UU Perampasan Aset itu penting sebagai solusi utk mengatasi masalah korupsi, Presiden Jokowi bisa keluarkan Perpu.Yg tdak penting seperti Perpu Cipta Kerja saja diterbitkan, apalagi Perpu terkait perampasan aset.Mohon Menkopolhukam beritau Presiden Jokowi segera terbitkan Perpu Perampasan Aset.Mau?Berani?#RakyatMonitor#	Soal UU Perampasan Aset Jika memang Presiden Jokowi memandang UU Perampasan Aset itu penting sebagai solusi utk mengatasi masalah korupsi Presiden Jokowi bisa keluarkan Perpu Yg tdak penting seperti Perpu Cipta Kerja saja diterbitkan apalagi Perpu terkait perampasan aset Mohon Menkopolhukam beritau Presiden Jokowi segera terbitkan Perpu Perampasan Aset Mau Berani Rakyat Monitor

Pada tahap *case folding*, semua data pada dokumen dilakukan konversi atau perubahan huruf kapital ke dalam huruf kecil (*lowercase*). Contoh hasilnya sebagaimana tabel 4.2. berikut.

Tabel 4. 2 Contoh Hasil *Case Folding*

Sebelum	Sesudah
Soal UU Perampasan Aset Jika memang Presiden Jokowi memandang UU Perampasan Aset itu penting sebagai solusi utk mengatasi masalah korupsi Presiden Jokowi bisa keluarkan Perpu Yg tdak penting seperti Perpu Cipta Kerja saja diterbitkan apalagi Perpu terkait perampasan aset Mohon Menkopolhukam beritau Presiden Jokowi segera terbitkan Perpu Perampasan Aset Mau Berani Rakyat Monitor	soal uu perampasan aset jika memang presiden jokowi memandang uu perampasan aset itu penting sebagai solusi utk mengatasi masalah korupsi presiden jokowi bisa keluarkan perpu yg tdak penting seperti perpu cipta kerja saja diterbitkan apalagi perpu terkait perampasan aset mohon menkopolhukam beritau presiden jokowi segera terbitkan perpu perampasan aset mau berani rakyat monitor

Selanjutnya, pada tahap normalisasi perubahan dan perbaikan kata yang disingkat ke dalam kata yang memiliki arti sama berdasarkan KBBI agar menjadi informasi yang dapat diproses dengan mudah misalnya "sdg" menjadi "sedang", utk" menjadi "untuk", "tdk" menjadi "tidak" dan sebagainya. Contoh hasilnya sebagaimana tabel 4.3. berikut.

Tabel 4. 3. Contoh Hasil Normalisasi

Sebelum	Sesudah
soal uu perampasan aset jika memang presiden jokowi memandang uu perampasan aset itu penting sebagai solusi utk mengatasi masalah korupsi presiden jokowi bisa keluarkan perpu yg tdak penting seperti perpu cipta kerja saja diterbitkan apalagi perpu terkait perampasan aset mohon menkopolhukam beritau presiden jokowi segera terbitkan perpu perampasan aset mau berani rakyat monitor	soal uu perampasan aset jika memang presiden jokowi memandang uu perampasan aset itu penting sebagai solusi untuk mengatasi masalah korupsi presiden jokowi bisa keluarkan perpu yang tidak penting seperti perpu cipta kerja saja diterbitkan apalagi perpu terkait perampasan aset mohon menkopolhukam beritau presiden jokowi segera terbitkan perpu perampasan aset mau berani rakyat monitor

Pada tahap *stemming*, seluruh kata imbuhan yang terdapat pada data tweet seperti *prefix*, *suffix* dan *konfix* dihilangkan dari kata, seperti menghapus "mem" dan "-kan" dan sebagainya. Adapun contoh hasilnya pada tabel 4.4. berikut.

Tabel 4. 4. Contoh Hasil *Stemming*

Sebelum	Sesudah
soal uu perampasan aset jika memang presiden jokowi memandang uu perampasan aset itu penting sebagai solusi untuk mengatasi masalah korupsi presiden jokowi bisa keluarkan perpu yang tidak penting seperti perpu cipta kerja saja diterbitkan apalagi perpu terkait perampasan aset mohon menkopolhukam beritau presiden jokowi segera terbitkan perpu perampasan aset mau berani rakyat monitor	soal uu rampas aset jika memang presiden jokowi pandang uu rampas aset itu penting sebagai solusi untuk atasi masalah korupsi presiden jokowi bisa keluar perpu yang tidak penting seperti perpu cipta kerja saja terbit apalagi perpu terkait rampas aset mohon menkopolhukam beritau presiden jokowi segera terbit perpu rampas aset mau berani rakyat monitor

Pada tahap *filtering*, dihapus kata yang tidak bermakna ataupun kurang penting seperti "dari", "yang", "untuk", "dan", "di" dan sebagainya. Adapun contoh hasilnya sebagaimana tabel 4.5. berikut.

Tabel 4. 5. Contoh Hasil *Filtering*

Sebelum	Sesudah
soal uu rampas aset jika memang presiden jokowi pandang uu rampas aset itu penting sebagai solusi untuk atasi masalah korupsi presiden jokowi bisa keluar perpu yang tidak penting seperti perpu cipta kerja saja terbit apalagi perpu terkait rampas aset mohon menkopolhukam beritau presiden jokowi terbit perpu rampas aset mau berani rakyat monitor	soal uu rampas aset presiden jokowi pandang uu rampas aset penting solusi atasi masalah korupsi presiden jokowi keluar perpu penting seperti perpu cipta kerja terbit perpu terkait rampas aset menkopolhukam presiden jokowi terbit perpu rampas aset berani rakyat monitor

Menggunakan operator *tokenize* pada RapidMiner, kalimat dipisahkan menjadi per kata agar mendapatkan kata yang memiliki nilai. Adapun hasilnya sebagaimana tabel 4.6. berikut.

Tabel 4. 6. Contoh Hasil *Tokenizing*

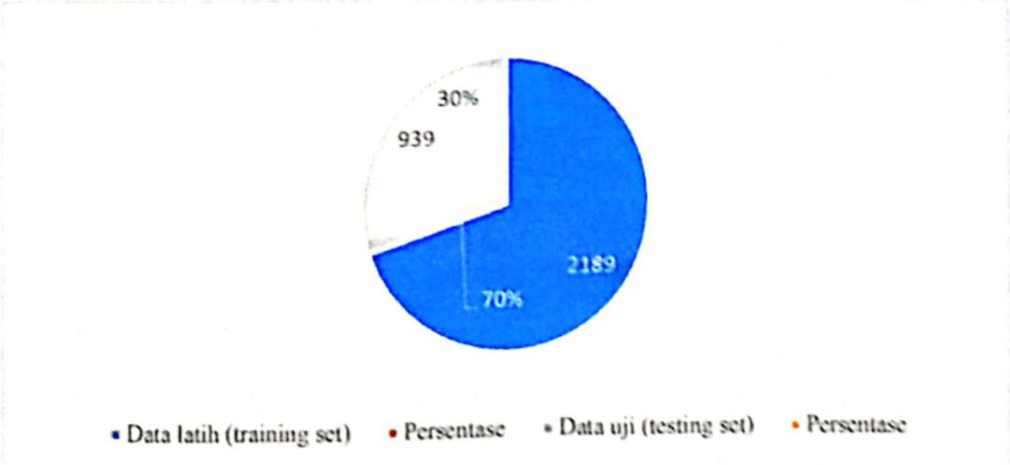
Sebelum	Sesudah
soal uu rampas aset presiden jokowi pandang uu rampas aset penting solusi atasi masalah korupsi presiden jokowi keluar perpu penting seperti perpu cipta kerja terbit perpu terkait rampas aset menkopolhukam presiden jokowi terbit perpu rampas aset berani rakyat monitor	'soal', 'uu', 'rampas', 'aset', 'presiden', 'jokowi', 'pandang', 'uu', 'rampas', 'aset', 'penting', 'solusi', 'atasi', 'masalah', 'korupsi', 'presiden', 'jokowi', 'keluar', 'perpu', 'penting', 'perpu', 'cipta', 'kerja', 'terbit', 'perpu', 'terkait', 'rampas', 'aset', 'menkopolhukam', 'presiden', 'jokowi', 'terbit 'perpu', 'rampas', 'aset', 'berani', 'rakyat', 'monitor',

Menggunakan operator *Remove Duplicate*, dilakukan penghapusan terhadap tweet yang sama supaya tidak terjadi masalah dalam analisis selanjutnya. Setelah dilakukan *preprocessing*, diperoleh data bersih sebanyak 3.128 untuk dilakukan ke tahap selanjutnya. Lebih jelas ditampilkan dalam tabel 4.7 berikut ini.

Tabel 4. 7. Perbandingan Data Tweet Sebelum Dan Sesudah Prepocessing

Sebelum Preprocessing	Sesudah Preprocessing
10.248	3.128

Selanjutnya, data dibagi menjadi 2189 data latih (*training set*) dan 939 data uji (*testing set*), atau 70% data latih dan 30% data uji yang berarti bahwa 70% dari data digunakan untuk melatih model, sementara 30% digunakan untuk menguji kinerja model. Hasil pembagian data latih dan data uji sebagai berikut.



Gambar 4. 3. Perbandingan Data Latih dan Data Uji

4.1.3. Tahap Analisis Sentimen

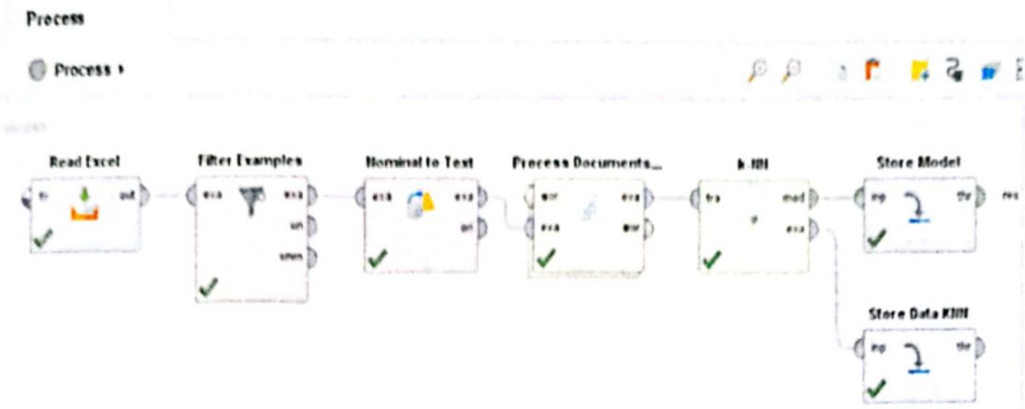
Pada tahap ini dilakukan pelabelan terhadap data yang telah dibersihkan. Data dibagi menjadi data latih data uji. Pelabelan data latih dilakukan secara manual dengan memberikan label positif, negatif, atau netral pada teks tweet pengguna twitter. Peneliti memberikan label kepada sebanyak 2189 data latih yang akan digunakan untuk melabeli data tweet lainnya menggunakan pemodelan pada RapidMiner.

Untuk melakukan pelabelan data latih pada label positif, negatif, atau netral pada teks tweet pengguna Twitter terkait Perpu Cipta Kerja, peneliti menggunakan sejumlah kata kunci atau frasa kunci yang relevan. Kata kunci yang peneliti gunakan sebagaimana tabel 4.8 berikut ini.

Tabel 4. 8. Frasa Kunci Pelabelan

No	Label	Frasa Kunci
1	Positif	1. Peningkatan investasi 2. Pembukaan lapangan kerja 3. Pertumbuhan ekonomi 4. Kemudahan berusaha 5. Perubahan positif 6. Dampak positif 7. Mendukung
2	Negatif	1. Pengurangan hak pekerja 2. Ketidaksetujuan 3. Dampak negatif 4. Demonstrasi 5. Protes 6. Keberatan 7. Menolak
3	Netral	1. Berita 2. Perubahan hukum 3. Kebijakan baru 4. Undang-Undang 5. Perubahan regulasi 6. Komentar 7. Analisis

Adapun model yang digunakan sebagaimana gambar 4.4 berikut.

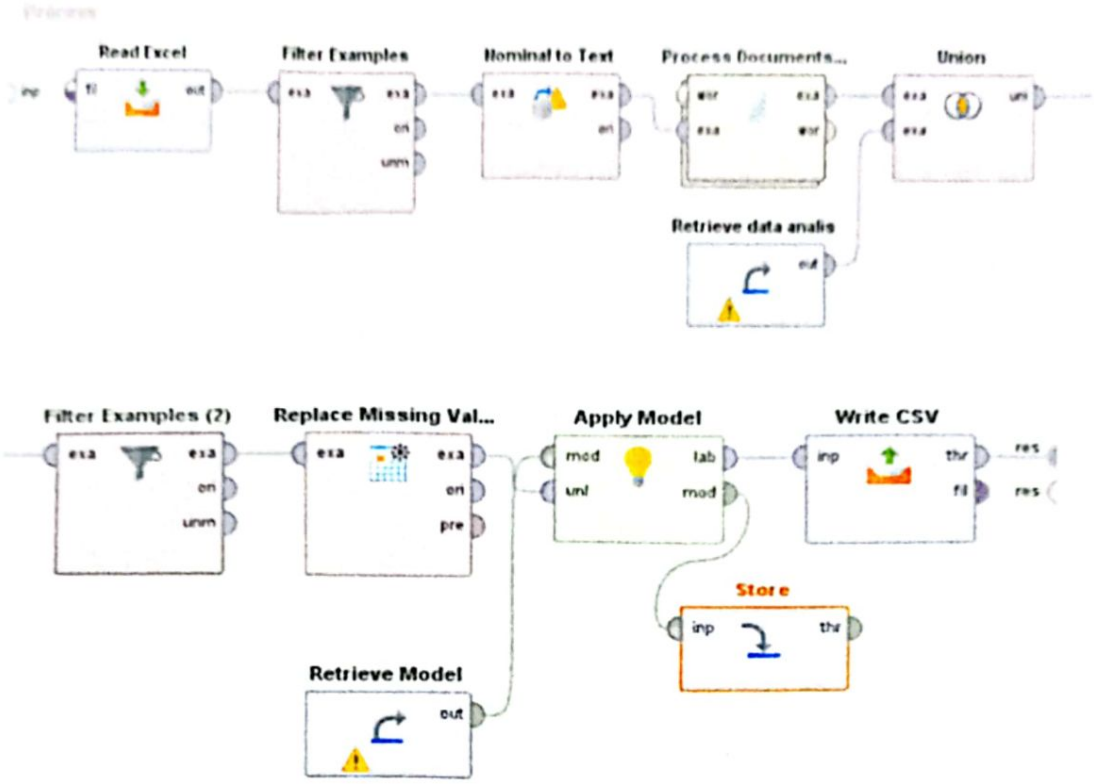


Gambar 4. 4. Model Data Latih

Model diawali dengan memasukkan data uji yang sudah diberi label secara manual ke operator *Read Excel* dilanjutkan dengan *Filter Examples*. Operator ini digunakan untuk mengembalikan contoh yang cocok dengan kondisi yang diberikan. Parameter yang ditentukan adalah sentimen dengan pilihan *is not missing* yaitu *record* yang salah satu atau bahkan lebih atributnya sudah diketahui nilainya dengan pemberian label. Selanjutnya adalah operator *Nominal To Text* untuk mengubah jenis atribut nominal yang dipilih menjadi teks. Hal ini termasuk memetakan semua nilai atribut ini ke nilai string yang sesuai.

Operator selanjutnya *Process Document from data* untuk menghasilkan vektor kata dari atribut string. Di dalamnya terdapat operator *Tokenize, Transform Cases, Filter Stopword,* dan *Filter Tokenz (by length).* Operator KNN dan RF merupakan algoritma yang digunakan untuk melakukan pemodelan. Hasil pemodelan ini dibagi menjadi model data latih yang disimpan pada panel Data menggunakan operator *store* dan hasil pelabelan yang disimpan menjadi *store* data latih pada Panel *Processes* di RapidMiner. Model dan data ini akan

digunakan ke langkah selanjutnya yaitu memberikan label data uji dengan latih yang sudah dibuat ini. Gambar model sebagaimana gambar 4.5. berikut ini.



Gambar 4. 5. Model Data Uji

Berdasarkan hasil pemrosesan data latih di atas, peneliti menyusun model untuk melabeli data uji yang belum diberi label dan menggabungkannya dengan data latih. Proses diawali dengan menginput dataset ke dalam operator *Read Excel.* Selanjutnya digunakan operator *Filter Examples* dengan parameter yang ditentukan adalah sentimen dengan pilihan *is missing* yaitu *record* yang salah satu atau bahkan lebih atributnya belum diketahui nilainya dengan pemberian label. *Nominal to text* digunakan untuk mengubah jenis atribut nominal yang dipilih menjadi teks. Operator *Process document* digunakan untuk memproses data. Operator *Onion* digunakan untuk membangun gabungan dari Set Contoh input (data latih dengan data uji). Set Contoh input digabungkan sedemikian rupa

sehingga atribut dan contoh dari kedua Set Contoh input adalah bagian dari gabungan Set Contoh yang dihasilkan dengan memasukkan data Latih yang sudah dibuat sebelumnya dan mengkoneksikan dengan operator *onion*.

Operator *Filter Examples* selanjutnya dengan set *filters is missing*. Operator *Replace Missing Values* digunakan untuk ini mengganti nilai yang hilang dalam Contoh Atribut yang dipilih dengan pengganti yang ditentukan yaitu nilai 0. Untuk menerapkan model pada *ExampleSet* digunakan Operator *Apply Model*. Setelah itu digunakan operator *Store* untuk menyimpan Objek di repositori data. Lokasi objek yang akan disimpan ditentukan melalui parameter entri repositori. Objek yang disimpan ini digunakan pada proses selanjutnya dengan menggunakan operator *Retrieve*.

Hasilnya diperoleh sebanyak 3.128 *examples*, 6 *special attributes*, dan 3.394 *reguler attributes* sebagaimana gambar 4.6 sebagai berikut.

Row No.	Sentimen	prediction(S...	confidence(...	confidence(...	confidence(...	aamiin	aamiiadiga	aasb
1	Negatif	Positif	0.021	0.483	0.495	0	0	0
2	Positif	Positif	0.018	0.475	0.508	0	0	0
3	Positif	Positif	0.018	0.434	0.548	0	0	0
4	Positif	Positif	0.018	0.436	0.546	0	0	0
5	Negatif	Positif	0.018	0.480	0.502	0	0	0
6	Positif	Positif	0.018	0.395	0.587	0	0	0
7	Negatif	Negatif	0.020	0.533	0.447	0	0	0
8	Negatif	Negatif	0.019	0.524	0.457	0	0	0
9	Positif	Positif	0.018	0.419	0.562	0	0	0
10	Positif	Positif	0.018	0.395	0.587	0	0	0
11	Positif	Positif	0.018	0.395	0.587	0	0	0

ExampleSet (3.128 examples, 5 special attributes, 3.394 regular attributes)

Gambar 4. 6. Output Model Data Uji

Setelah data diberi label secara otomatis, kemudian peneliti memeriksa kembali ketepatan hasil pengujian dengan pandangan peneliti. Hasil tersebut kemudian dikonsultasikan kepada seorang ahli, yaitu Ibu Wistina Seneru, S.Pd.B, M.Pd. selaku dosen mata kuliah Bahasa Indonesia pada Sekolah Tinggi Ilmu Agama Buddha Jinarakkhita Lampung. Hal ini bertujuan untuk memastikan bahwa data yang telah berikan label sentimen yang diberikan secara otomatis sudah tepat atau belum. Ahli ini memiliki pemahaman mendalam tentang bahasa Indonesia, termasuk ekspresi, nuansa, dan makna kata yang mungkin tidak terdeteksi oleh algoritma analisis sentimen.

Ditemukan perbedaan pandangan di antara pengguna Twitter terhadap PERPPU Cipta Kerja dalam analisis sentimen ini. Ada kelompok yang mendukung dan memberikan sentimen positif terhadap peraturan ini, sementara itu ada juga kelompok yang memberikan sentimen negatif dan kritik terhadapnya. Ada juga yang bersikap netral terhadap kebijakan tersebut. Oleh sebab itu, dengan bantuan ahli bahasa, ambiguitas bahasa seperti arti kata dan kalimat yang bisa berubah berdasarkan konteksnya dapat diatasi sehingga hasil analisis sentimen menjadi lebih akurat. Dengan melibatkan seorang ahli bahasa dalam validasi dan interpretasi hasil analisis sentimen ini dan memberikan perspektif profesional terhadap hasil penelitian, peneliti dapat meningkatkan keandalan dan kredibilitas penelitian. Contoh hasil pelabelan sebagaimana gambar 4.7. berikut.

Text	Sentimen
sahkan perppu cipta kerja buruh ancam mogok massal	Negatif
sahkan perppu cipta kerja demokrat walkout	Negatif
sahkan perppu cipta kerja tolak buruh partai oposisi pakar perppu kesah substansial timbul ketidakpastian hukum	Negatif
sahkan perpu cipta kerja rapat paripurna	Positif
sahkan perpu cipta kerja rapat paripurna mayat tanda orang pilih partai	Negatif
sahkan perpu cipta kerja undang	Positif
sahkan perpu cipta kerja leceh hukum	Negatif
sahkan perpu cipta kerja leceh hukum kolosal	Negatif
sahkan perpu cipta kerja rapat paripurna	Negatif
setuju atur perintah ganti undang nomor cipta kerja undang rapat paripurna	Positif
setuju atur perintah ganti undang perppu nomor cipta kerja ciptaker undang rapat paripurna	Positif
setuju atur perintah ganti undang perppu nomor cipta kerja ciptaker undang rapat paripurna jakarta selasa ketua kamar dagang indonesia kadin	Positif
setuju atur perintah ganti undang perppu nomor cipta kerja undang	Positif
setuju kesah atur perintah ganti undang perppu nomor cipta kerja undang harap indeks mudah bisnis indonesia lonjak tajam	Positif
setuju perppu ciptaker ciptakerja	Positif
setuju atur perintah ganti undang perppu nomor cipta kerja ciptaker undang undang	Positif
setuju atur perintah ganti undang perppu nomor cipta kerja ciptaker	Positif
setuju atur perintah ganti perppu nomor cipta kerja ciptaker undang	Positif
setuju atur perintah ganti undang perppu nomor cipta kerja undang	Positif
tunda kesah atur perintah ganti undang cipta kerja perppu ciptaker undang sidang putus tunggu sidang tegakkan hukum	Negatif

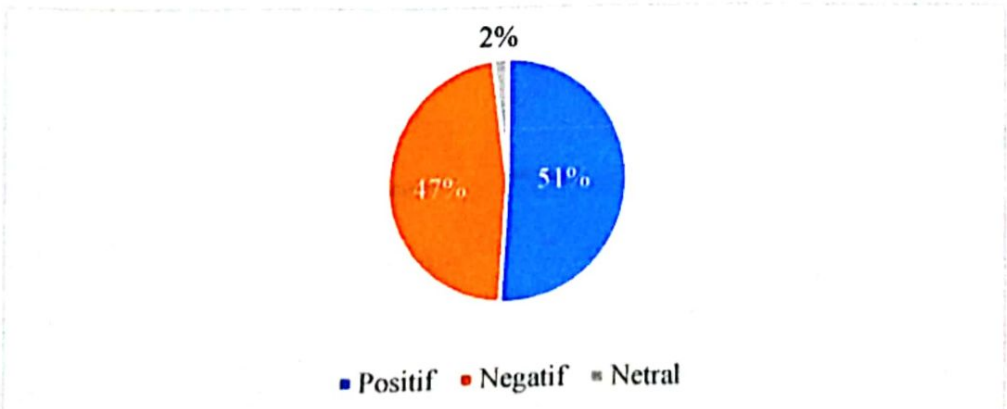
Gambar 4. 7. Pelabelan Manual

Total hasil dari pelabelan berdasarkan sentimen positif, negatif, dan netral sebagaimana tabel 4.9 berikut ini.

Tabel 4. 9. Jumlah Hasil Pelabelan

No	Label	Jumlah
1	Positif	1599
2	Negatif	1473
3	Netral	53
	Jumlah	**3128**

Hasilnya disajikan dalam bentuk diagram pie pada gambar 4.8 berikut.

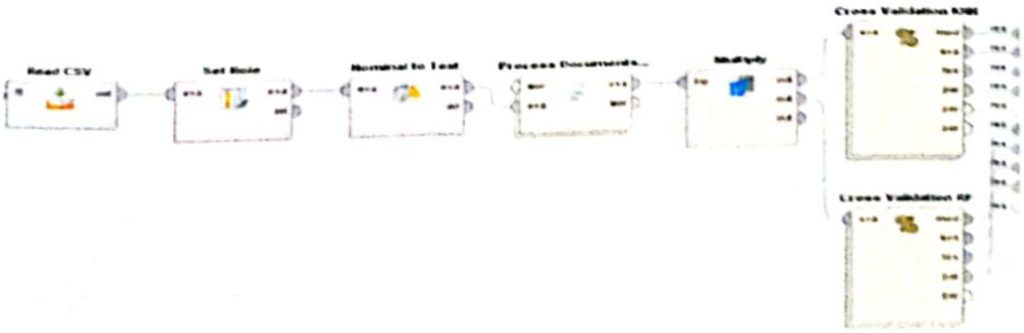


Gambar 4. 8. Persentase Sentimen

Berdasarkan tabel 4.9 dan gambar 4.8 di atas terlihat bahwa sentimen Positif memiliki jumlah tertinggi dengan 51% dari total sampel. Sentimen Negatif memiliki jumlah 47%, mendekati jumlah sentimen Positif, sedangkan sentimen Netral memiliki jumlah yang sangat rendah, yaitu hanya 2%. Hasil ini menunjukkan bahwa dalam data Twitter terkait Perppu Cipta Kerja, mayoritas sentimen yang diekspresikan adalah positif, sementara sentimen negatif juga cukup signifikan. Analisis sentimen ini memberikan wawasan tentang bagaimana masyarakat merespons atau bereaksi terhadap suatu kebijakan atau peraturan. Dengan demikian, penelitian ini dapat membantu pemerintah atau pemangku kepentingan untuk memahami persepsi dan sentimen masyarakat terhadap Perppu Cipta Kerja, apakah positif, negatif, atau netral.

4.1.4. Tahap Implementasi Algoritma

Peneliti membuat dua model, yaitu model implementasi algoritma tanpa menggunakan PSO dan model implementasi algoritma menggunakan PSO sebagaimana gambar 4.9. dan 4.10 berikut.



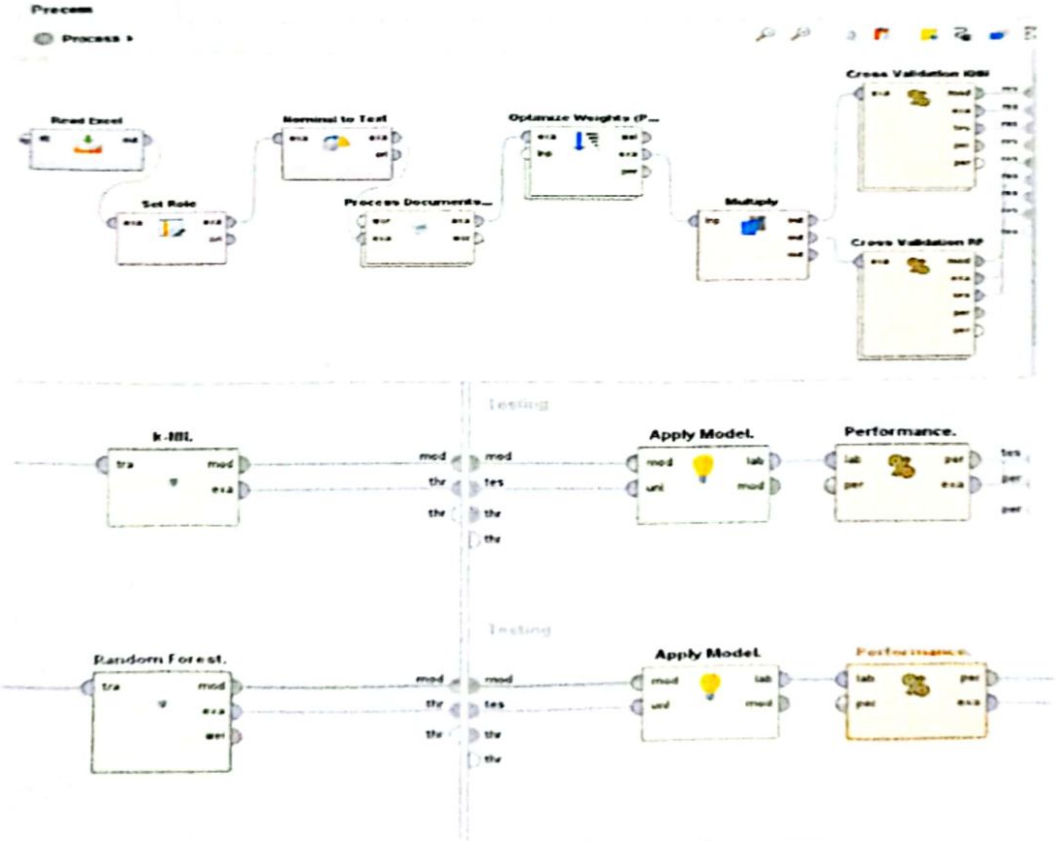
Gambar 4. 9. Model Analisis Sentimen Tanpa PSO

Data set yang sudah siap dilakukan pengujian, diinput ke dalam operator *Read csv*. Untuk mengubah peran dari satu atau lebih *Atribut* yaitu dengan nama atribut Sentimen dan target *role* nya label, digunakan operator *Set Role*. Supaya dapat dilakukan pemrosesan teks, menggunakan Operator *Nominal To Text*. Operator ini mengubah jenis atribut nominal yang dipilih menjadi teks sekaligus memetakan semua nilai atribut ini ke nilai string yang sesuai. Pada *sub Process Document* menghasilkan vektor kata dari atribut string sesuai dengan pengaturan pada sebelumnya. Operator *Multiply* digunakan untuk mengambil Objek RapidMiner dari *port input* dan mengirimkan salinannya ke *port output*. Setiap *port* yang terhubung membuat salinan independen. Jadi mengubah satu salinan tidak berpengaruh pada salinan lainnya ke *Operator Cross* Validation untuk KNN dan *Random Forest*. Operator ini melakukan validasi silang untuk memperkirakan kinerja statistik model data uji. Input *ExampleSet* dipartisi menjadi k subset dengan ukuran yang sama.

Di dalam *Operator Cross Validation* di setting pada Panel *Training* diinput operator masing-masing KNN dengan k-5 dan *Random Forest* dengan *number of tree*-100, kritreria *gain ratio* yang berarti bahwa varian perolehan informasi yang

menyesuaikan perolehan informasi untuk setiap Atribut untuk memungkinkan keluasan dan keseragaman nilai Atribut, dan *maximal depth* pada 10 yang berarti bahwa kedalaman pohon ditentukan pada ukuran dan karakteristik dari *ExampleSet* yaitu 10. Kemudian pada Panel *Testing* ditentukan operator *Apply Model* dan *Operator Performance* untuk menampilkan output akurasi hasil pengujian.

Adapun model dengan PSO seperti pada gambar 4.10 berikut.



Gambar 4. 10 Model Analisis Sentimen dengan PSO

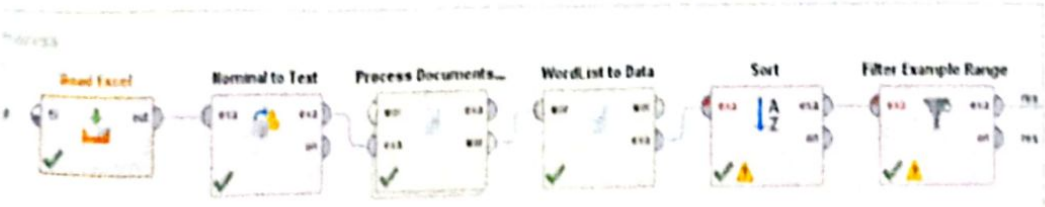
Berdasarkan gambar di atas, dapat dijelaskan bahwa model pengujian dengan optimasi PSO diawali dengan input *datasets* melalui Operator *read Excel*. *Set role* digunakan untuk menjelaskan bagaimana Operator lain menangani Atribut ini. Peran diatur secara khusus dengan regular yaitu *attribute name* nya

adalah "sentiment" dan *target role* nya adalah "label". Operator *Nominal to text* digunakan untuk mengubah semua atribut nominal menjadi atribut *string*. Pada operator ini, setiap nilai nominal hanya digunakan sebagai nilai *string* dari atribut baru. Jika nilai pada atribut nominal tidak ada, maka nilai baru juga akan hilang.

Peneliti kemudian menambahkan operator *Process Documents from Data* yang digunakan untuk menghasilkan vektor kata dari atribut string. Di dalamnya terdapat proses sebagaimana dalam pemrosesan teks. Kemudian dilanjutkan dengan *Operator Multiply* yang digunakan untuk mengambil Objek RapidMiner dari port input dan mengirimkan salinannya ke port output. Setiap port yang terhubung membuat salinan independen. Hal ini supaya mengubah satu salinan tidak berpengaruh pada salinan lainnya. Operator selanjutnya yaitu *Operator Optimize Weights (PSO)* yang di dalamnya terdapat proses pengujian model dengan *Cross Validation* pada 10-fold. Di dalam Operator *Cross Validation* diproses masing-masing algoritma dan kriteria pengujian.

4.1.5. Tahap Visualisasi

Pada tahap ini peneliti memvisualisasikan kata-kata yang sering muncul pada tweet menggunakan menu *word cloud* pada RapidMiner dengan model sebagaimana gambar 4.11. berikut.



Gambar 4. 11. Model *Wordcloud*

Datasets diinput ke RapidMiner dengan Operator *Read Excel*. Kemudian Operator *Nominal to Text* untuk mengubah jenis atribut nominal menjadi teks. Operator *Process Documents* sebagaimana dijelaskan sebelumnya. Kemudian Operator *WordList to Data* yang digunakan untuk membuat kumpulan data dari daftar kata yang berisi baris untuk setiap kata dan atribut untuk kata itu sendiri, jumlah dokumen yang terjadi, jumlah dokumen berlabel yang terjadi dan untuk setiap nomor kelas yang terjadi dalam dokumen kelas ini. Operator *Sort* digunakan untuk mengurutkan kumpulan data input dalam urutan menaik atau menurun menurut beberapa atribut. *Filters* yang digunakan yaitu total kata yang sering muncul dan diurutkan dengan urutan menurun (*descending*). Operator *Filter Example Range* digunakan untuk memilih contoh baris dari Set Contoh yang harus disimpan dan contoh mana yang harus dihapus. Contoh dalam kisaran indeks yang ditentukan disimpan, contoh yang tersisa dihapus. Contoh dalam penelitian ini ditentukan sebanyak 15.

Hasilnya difilter 15 kata yang sering muncul seperti pada tabel 4.10 dan gambar 4.12.

Tabel 4. 10. Output *Wordcloud*

Row No.	*Word*	*in documents*	*total*	*in class (positif)*	*in class (negative)*	*in class (netral)*
1	kerja	2642	3921	1370	2484	67
2	cipta	2612	3598	1254	2285	59
3	perppu	899	1896	287	1576	33
4	dukung	1118	1301	178	1100	23
5	perpu	1136	1251	779	443	29
6	ciptaker	974	1058	71	972	15
7	ciptakerja	800	856	496	346	14
8	undang	531	748	408	335	5
9	perintah	589	707	355	348	4
10	nomor	484	535	325	203	7
11	buruh	384	497	416	75	6
12	atur	447	492	280	210	2

Row No.	*Word*	*in documents*	*total*	*in class (positif)*	*in class (negative)*	*in class (netral)*
13	ganti	387	399	231	166	2
14	ekonomi	361	382	56	326	0
15	indonesia	312	337	111	224	2

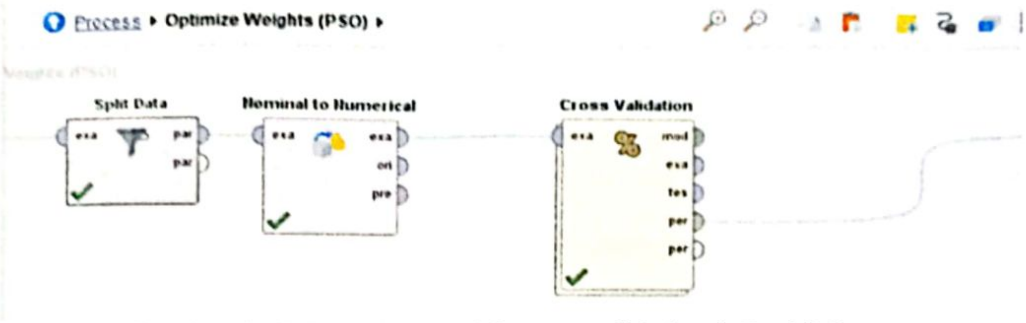
Berdasarkan tabel 4.10 di atas, kata yang sering muncul kata kerja yaitu sebanyak 2642 kata dan tweet cipta yang mencapai 2612 kata hal ini berarti bahwa jumlah kemunculan kata "kerja" dan "cipta" ini mencerminkan seberapa relevan RUU tersebut dengan isu-isu terkini yang berkaitan dengan ekonomi, ketenagakerjaan, dan inovasi di Indonesia. Selain itu, banyaknya jumlah kemunculan kata "cipta" yang lebih tinggi bisa jadi menunjukkan bahwa RUU tentang Cipta Kerja tersebut banyak membahas penciptaan dan inovasi dalam berbagai sektor. Kemunculan kata "kerja" yang signifikan menunjukkan fokus pada aspek ketenagakerjaan yang direspon oleh pengguna twitter.

Dari hasil *wordcloud* tersebut diketahui bahwa PERPPU Cipta Kerja merupakan isu yang kontroversial di kalangan masyarakat, dengan perbedaan pandangan yang signifikan. Adapun plot tipe hasil *wordcloud* kemudian diekspor ke format chart PNG yang hasilnya sebagaimana gambar 4.12 berikut.

Gambar 4. 12. Visualisasi *Wordcloud*

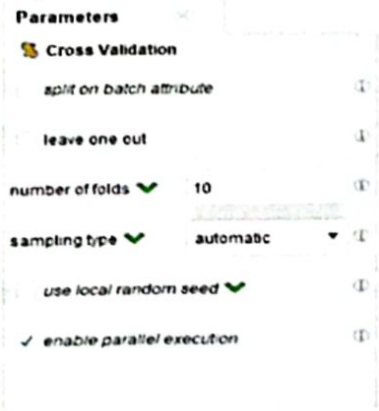
4.1.6. Tahap Validasi Model

Pada tahap ini validasi yang digunakan adalah operator k-*10 Fold cross validation*. Tampilan desain pada RapidMiner sebagaimana gambar 4.13 berikut.



Gambar 4. 13. Tampilan model proses validasi pada RapidMiner

Konfigurasi parameter validasi pada gambar 4.14 berikut.

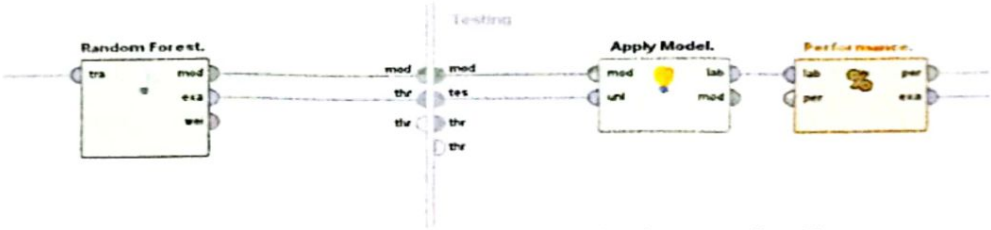


Gambar 4. 14. Tampilan Parameter *Cross Validation* pada RapidMiner

Salah satu subset K disimpan sebagai dataset uji (yaitu, masukan ke subproses uji). Subset k-1 yang tersisa digunakan sebagai dataset pelatihan (yakni input ke subproses Pelatihan). Proses validasi silang kemudian diulang sebanyak 10 kali, dan setiap subset k digunakan tepat satu kali sebagai data uji. K hasil dari 10 pengulangan dirata-ratakan (atau digabungkan) menghasilkan estimasi tunggal. Nilai K dapat ditentukan dengan menggunakan parameter jumlah lipatan.

Evaluasi performa model pada set pengujian independen memberikan estimasi performa yang baik pada set data yang tidak terlihat. Ini juga menunjukkan jika "*overfitting*" telah terjadi. Ini berarti bahwa model merepresentasikan data eksperimen dengan sangat baik, tetapi tidak menggeneralisasi dengan baik ke data baru sehingga kinerjanya bisa jauh lebih buruk berdasarkan data uji.

Contoh proses yang digunakan untuk menghasilkan nilai dari evaluasi terdapat di dalam operator *Cross Validation* menggunakan algoritma *Random Forest* sebagaimana gambar 4.15 berikut.



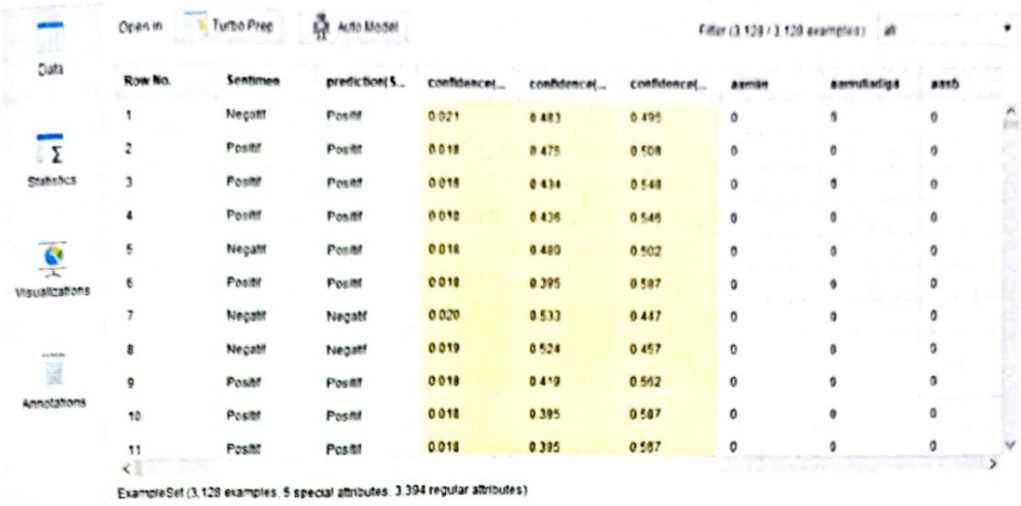
Gambar 4. 15. Proses Validation pada Algoritma *Random Forest*

Setelah dilakukan pengujian, contoh hasil dari evaluasi menggunakan *cross validation* algoritma KNN dan *Random Forest* sebagaimana gambar 4.16 dan 4.17 berikut.

Row No.	Sentimen	prediction(S...	confidence(...	confidence(...	confidence(...	aamiin	aamiiadiga	aasb
1	Positif	Positif	0	0.328	0.672	0	0	0
2	Positif	Positif	0	0	1	0	0	0
3	Positif	Negatif	0	0.613	0.387	0	0	0
4	Negatif	Negatif	0	0.800	0.200	0	0	0
5	Negatif	Negatif	0	0.801	0.199	0	0	0
6	Negatif	Negatif	0	1	0	0	0	0
7	Positif	Positif	0	0	1	0	0	0
8	Positif	Positif	0	0	1	0	0	0
9	Positif	Positif	0	0	1	0	0	0
10	Positif	Positif	0	0	1	0	0	0
11	Positif	Positif	0	0	1	0	0	0

ExampleSet (3.128 examples, 5 special attributes, 3.394 regular attributes)

Gambar 4. 16. Hasil *cross validation* KNN



Gambar 4. 17. Hasil *cross validation Random Forest*

4.2. Pembahasan

4.2.1. Pengujian *Algoritma K-Nearest Neighbors (KNN)* dan *Random Forest*

Untuk menjawab rumusan penelitian pertama, yaitu apakah fitur PSO dapat mengoptimasi sentimen pengguna media sosial Twitter terhadap PERPPU Tentang Cipta Kerja menggunakan algoritma klasifikasi KNN dan *Random Forest*, dilakukan pengujian kedua model menggunakan algoritma KNN dan *Random Forest* yang menghasilkan data sebagaimana tabel 4.11 berikut ini.

Tabel 4. 11. Hasil Pengujian KNN dan *Random Forest* Tanpa PSO

Metode	***PerformanceVektor (Performance)***
	accuracy
KNN	80.40%
Random Forest	77.21%

Sumber: Data penelitian

Berdasarkan tabel 4.11 di atas, dapat dijelaskan bahwa hasil pengujian tanpa menggunakan fitur optimasi PSO, kedua algoritma memperoleh hasil yang "baik". Hal ini dapat dilihat dari nilai *accuracy* dari metode KNN sebesar 80.40%. Hasil

ini berarti bahwa model KNN mampu mengklasifikasikan data dengan benar dan dengan tingkat keakuratan yang tinggi. *Random Forest* memperoleh hasil akurasi sebesar 77.21% yang menunjukkan hasil akurasi "Baik". Hal ini berarti bahwa *Random Forest* juga mampu mengklasifikasikan data sentimen Perppu Cipta Kerja dengan benar dan tingkat akurasi yang baik. Adapun hasil pengujian KNN sebagaimana tabel 4.12. berikut.

Tabel 4. 12. Hasil Pengujian KNN

accuracy: 80.40% +/- 1.31% (micro average: 80.40%)				
	true Negatif	**true Positif**	**true Netral**	**class precision**
pred. Negatif	1079	178	19	84.56%
pred. Positif	377	1420	21	78.11%
pred. Netral	17	1	16	47.06%
class *recall*	73.25%	88.81%	28.57%	

Berdasarkan hasil pada tabel 4.12 terlihat bahwa dari hasil *confusion matrix accuracy* yang dihasilkan sebesar 80.40% yang menunjukkan bahwa sebagian besar sampel telah diklasifikasikan dengan benar. Perhitungan *nilai recall, precision, dan F1-Score* untuk masing-masing kelas sebagai berikut.

1. Kelas Negatif

   a. *Recall (Sensitivitas)* untuk kelas Negatif:

$$Recall\ Negatif = \frac{True\ Negatives}{True\ Negatives + False\ Negatives}$$

$$Recall\ Negatif = \frac{1420}{1420 + 178} = 88.81\%$$

   Nilai *recall* sebesar ini berarti bahwa model berhasil mendeteksi sebagian besar *instance* yang sebenarnya adalah Negatif.

b. *Presisi (Precision)* untuk kelas Negatif:

$$Precision\ Negatif = \frac{True\ Negatives}{True\ Negatives + False\ Positives}$$

$$Precesion\ Negatif = \frac{1420}{1420 + 377} = 79.03\%$$

Presisi sebesar 79.03% ini menunjukkan bahwa sebagian besar prediksi yang dibuat oleh model sebagai Negatif adalah benar-benar Negatif.

c. F1-Score untuk kelas Negatif:

$$F1 - score\ Negatif = \frac{2\ x\ Precision\ Negatif\ x\ Recall\ Negatif}{Precision\ Negatif + Recall\ Negatif}$$

$$F1 - score\ Negatif = \frac{2 \text{ x } 79.03\% \text{ x } 88.81\%}{79.03\% \text{ x } 88.81\%} = 83.63\%$$

F1-Score dengan nilai sebesar 83.63% ini menunjukkan adanya perpaduan yang baik antara presisi dan *recall* untuk kelas Negatif.

2. Kelas Positif

a. *Recall (Sensitivitas)* untuk kelas Positif:

$$Recall\ Positif = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$Recall\ Positif = \frac{178}{178 + 19} = 90.30\%$$

Nilai *recall* sebesar ini berarti bahwa model berhasil mendeteksi sebagian besar *instance* yang sebenarnya adalah Positif.

b. *Presisi (Precision)* untuk kelas Positif:

$$Precision\ Positif = \frac{True\ Positives}{True\ Positivies + False\ Positives}$$

$$Precesion\ Positif = \frac{178}{178 + 377} = 32.01\%$$

Presisi sebesar 32.01% ini menunjukkan bahwa prediksi yang dibuat oleh model sebagai Positif adalah benar-benar Positif.

c. F1-*Score* untuk kelas Positif:

$$F1 - score\ Positif = \frac{2\ x\ Precision\ Positif\ x\ Recall\ Positif}{Precision\ Positif + Recall\ Positif}$$

$$F1 - score\ Positif = \frac{2\ x\ 32.01\%\ x\ 90.30\%}{32.01\%\ x\ 90.30\%} = 47.23\%$$

F1-*Score* dengan nilai sebesar 47.23% ini menunjukkan adanya perpaduan yang kurang baik antara presisi dan *recall* untuk kelas Positif.

3. Kelas Netral

a. *Recall (Sensitivitas)* untuk kelas Netral:

$$Recall\ Netral = \frac{True\ Netral}{True\ Netral + False\ Negatives}$$

$$Recall\ Netral = \frac{16}{16+1} = 94.12\%$$

Nilai *recall* sebesar ini berarti bahwa model berhasil mendeteksi sebagian besar *instance* yang sebenarnya adalah Netral.

b. *Presisi (Precision)* untuk kelas Netral:

$$Precision\ Netral = \frac{True\ Netral}{True\ Netral + False\ Positives}$$

$$Precesion\ Netral = \frac{16}{16+377} = 4.09\%$$

Presisi sebesar 4.07% artinya bahwa dari semua yang diprediksi sebagai sentimen Netral oleh model, hanya sekitar 4.07% yang sebenarnya adalah Netral. Prediksi ini menunjukkan bahwa model memiliki tingkat presisi yang rendah dalam memprediksi sentimen Netral.

c. F1-Score untuk kelas Netral:

$$F1 - score\ Netral = \frac{2\ x\ Precision\ Netral\ x\ Recall\ Netral}{Precision\ Netral + Recall\ Netral}$$

$$F1 - score\ Netral = \frac{2\ x\ 4.09\%\ x\ 09.12\%}{14.09\%\ x\ 09.12\%} = 7.85\%$$

F1-Score dengan nilai sebesar 7.85% ini menunjukkan adanya perpaduan yang baik antara presisi dan *recall* untuk kelas Netral.

Berdasarkan hasil ini, model memiliki kinerja yang baik dalam mendeteksi kelas Negatif, Positif, dan Netral. Untuk kelas Negatif, model memiliki *recall* yang baik (88.81%), yang berarti kemampuannya untuk mengidentifikasi kasus Negatif cukup baik, dan presisinya juga baik (79.03%). Untuk kelas Negatif, *recall* cukup tinggi (90.30%), yang berarti kemampuannya untuk mengidentifikasi kasus Positif juga baik, tetapi presisinya rendah (32.01%), yang mengindikasikan adanya banyak *false positive*. Pada Kelas Netral, *recall* sangat tinggi (94.12%), menunjukkan bahwa model sangat baik dalam mengidentifikasi kasus Netral, tetapi presisinya sangat rendah (4.07%), yang berarti banyak *false positive* dalam klasifikasi ini. Kelas Netral memiliki *F1-Score* yang lebih rendah, menunjukkan bahwa model mungkin memiliki kesulitan dalam mengidentifikasi dengan tepat *instance* yang sebenarnya adalah Netral.

Hasil pengujian algoritma *Random Forest* sebagaimana tabel 4.13. berikut.

Tabel 4. 13. Hasil Pengujian *Random Forest*

accuracy: 77.21% +/- 2.60% (micro average: 77.21%)				
	true Negatif	**true Positif**	**true Netral**	**class precision**
pred. Negatif	1008	192	21	82.56%
pred. Positif	465	1407	35	73.78%
pred. Netral	0	0	0	0.00%
class *recall*	68.43%	87.99%	0.00%	

Berdasarkan hasil pada tabel 4.13 terlihat bahwa dari hasil *confusion matrix accuracy* yang dihasilkan sebesar 77.21% yang menunjukkan bahwa sebagian besar sampel telah diklasifikasikan dengan benar. Perhitungan *nilai recall, precision,* dan *F1-Score* untuk masing-masing kelas sebagai berikut.

1. Kelas Negatif

   a. *Recall (Sensitivitas)* untuk kelas Negatif:

$$Recall\ Negatif = \frac{True\ Negatives}{True\ Negatives + False\ Negatives}$$

$$Recall\ Negatif = \frac{1407}{1407 + 192} = 88.06\%$$

   Nilai *recall* sebesar ini berarti bahwa model berhasil mendeteksi sebagian besar *instance* yang sebenarnya adalah Negatif.

   b. *Presisi (Precision)* untuk kelas Negatif:

$$Precision\ Negatif = \frac{True\ Negatives}{True\ Negatives + False\ Positives}$$

$$Precesion\ Negatif = \frac{1407}{1407 + 465} = 75.16\%$$

   Presisi sebesar 75.16% ini menunjukkan bahwa sebagian besar prediksi yang dibuat oleh model sebagai Negatif adalah benar-benar Negatif.

   c. F1-Score untuk kelas Negatif:

$$F1 - score\ Negatif = \frac{2\ x\ Precision\ Negatif\ x\ Recall\ Negatif}{Precision\ Negatif + Recall\ Negatif}$$

$$F1 - score\ Negatif = \frac{2\ x\ 75.16\%\ x\ 88.06\%}{75.16\%\ x\ 88.06\%} = 81.03\%$$

   F1-*Score* dengan nilai sebesar 81.03% ini menunjukkan adanya perpaduan yang baik antara presisi dan *recall* untuk kelas Negatif.

2. Kelas Positif

a. *Recall (Sensitivitas)* untuk kelas Positif:

$$Recall\ Positif = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$Recall\ Positif = \frac{1407}{1407 + 35} = 97.59\%$$

Nilai *recall* sebesar ini berarti bahwa model berhasil mendeteksi sebagian besar *instance* yang sebenarnya adalah Positif.

b. *Presisi (Precision)* untuk kelas Positif:

$$Precision\ Positif = \frac{True\ Positives}{True\ Positivies + False\ Positives}$$

$$Precesion\ Positif = \frac{1407}{1407 + 465} = 75.16\%$$

Presisi sebesar 75.16% ini menunjukkan bahwa sebagian besar prediksi yang dibuat oleh model sebagai Positif adalah benar-benar Positif.

c. F1-Score untuk kelas Positif:

$$F1 - score\ Positif = \frac{2\ x\ Precision\ Positif\ x\ Recall\ Positif}{Precision\ Positif + Recall\ Positif}$$

$$F1 - score\ Positif = \frac{2\ x\ 75.16\%\ x\ 97.59\%}{75.16\%\ x\ 97.59\%} = 85.09\%$$

F1-Score dengan nilai sebesar 85.09% ini menunjukkan adanya perpaduan yang baik antara presisi dan *recall* untuk kelas Positif.

3. Kelas Netral

Pada kasus ini, *Recall*, dan F1-Score tidak terdefinisi karena pembaginya adalah 0 (dibagi dengan nol). Dalam hal ini, terlihat bahwa model memiliki kinerja yang sangat baik dalam mendeteksi kelas Negatif, dengan F1-Score

hampir 100%. Namun, kinerja model dalam mendeteksi kelas Positif adalah baik, tetapi memiliki F1-Score yang lebih rendah dibandingkan dengan kelas Negatif. Kelas Netral tidak dapat dievaluasi dengan baik oleh model karena tidak ada prediksi yang benar untuk kelas ini.

Dapat disimpulkan bahwa untuk kelas Negatif, model memiliki *recall* yang baik yaitu sebesar 88/06% yang berarti bahwa model memiliki kemampuan untuk mengidentifikasi kasus Negatif dengan cukup Baik. Presisinya juga baik dengan nilai sebesar 75.16%. *Recall* pada kelas Positif sangat tingggi, yaitu sebesar 97.59% yang berarti model memiliki kemampuan yang sangat tinggi mengidentifikasi kasus Positif. Presisinya pada kasus ini juga baik, yaitu sebesar 75.16%. Pada kelas Netral tidak ada hasil yang relevan karena model tidak dapat memprediksi *instance* sebagai Netral.

Pada proses selanjutnya, peneliti melakukan pengujian dengan menggunakan optimasi fitur PSO dan diperoleh hasil sebagaimana tabel 4.14. berikut.

Tabel 4. 14. Hasil Pengujian KNN dan *Random Forest* dengan PSO

Metode	PerformanceVektor (*Performance*)
	accuracy
KNN	85.23%
Random Forest	80.53%

Tabel 4.14 di atas menjelaskan hasil pengujian setelah menggunakan optimasi PSO. Terdapat peningkatan yang signifikan pada kedua model setelah dilakukan optimasi dengan PSO. Tingkat keberhasilan prediksi menjadi lebih tinggi dengan optimasi PSO meskipun terdapat perbedaan besaran peningkatan. Hal ini menandakan bahwa kemampuan masing-masing model dalam

mengklasifikasikan data memiliki perbedaan. Dalam hal ini, melalui penggunaan KNN dan *Random Forest* berbasis PSO, penelitian ini berhasil mengidentifikasi sentimen positif, negatif, dan netral dari pengguna Twitter terhadap PERPPU Cipta Kerja. Hal ini memberikan gambaran tentang bagaimana masyarakat merespons dan merasa terhadap peraturan tersebut.

Berdasarkan uraian di atas dapat disimpulkan bahwa fitur PSO dapat mengoptimasi sentimen pengguna media sosial Twitter terhadap PERPPU Tentang Cipta Kerja menggunakan algoritma klasifikasi KNN dan *Random Forest.* Hal ini dibuktikan dengan perbandingan hasil pengujian kedua model yang menunjukkan adanya kenaikan yang signifikan antara hasil pengujian model sebelum dan sesudah dilakukan optimasi dengan PSO, terutama pada algoritma KNN.

Hasil pengujian algoritma KNN dengan optimasi PSO sebagaimana tabel 4.15 berikut.

Tabel 4. 15. Hasil Pengujian KNN dengan Optimasi PSO

accuracy: 85.23% +/- 1.53% (micro average: 85.23%)				
	true Negatif	**true Positif**	**true Netral**	**class precision**
pred. Negatif	1318	265	29	81.76%
pred. Positif	154	1333	12	88.93%
pred. Netral	1	1	15	88.24%
class *recall*	89.48%	83.36%	26.79%	

Berdasarkan hasil pada tabel 4.15 terlihat bahwa dari hasil *confusion matrix accuracy* yang dihasilkan sebesar 85.23% yang menunjukkan bahwa sebagian besar sampel telah diklasifikasikan dengan benar. Perhitungan nilai *recall, precision,* dan *F1-Score* untuk masing-masing kelas sebagai berikut.

1. Kelas Negatif

   a. *Recall (Sensitivitas)* untuk kelas Negatif:

$$Recall\ Negatif = \frac{True\ Negatives}{True\ Negatives + False\ Negatives}$$

$$Recall\ Negatif = \frac{1333}{1333 + 265} = 83.45\%$$

   Nilai *recall* sebesar ini berarti bahwa model berhasil mendeteksi sebagian besar *instance* yang sebenarnya adalah Negatif.

   b. *Presisi (Precision)* untuk kelas Negatif:

$$Precision\ Negatif = \frac{True\ Negatives}{True\ Negatives + False\ Positives}$$

$$Precesion\ Negatif = \frac{1333}{1333 + 154} = 89.62\%$$

   Presisi sebesar 89.62% ini menunjukkan bahwa sebagian besar prediksi yang dibuat oleh model sebagai Negatif adalah benar-benar Negatif.

   c. F1-*Score* untuk kelas Negatif:

$$F1 - score\ Negatif = \frac{2\ x\ Precision\ Negatif\ x\ Recall\ Negatif}{Precision\ Negatif + Recall\ Negatif}$$

$$F1 - score\ Negatif = \frac{2\ x\ 89.62\%\ x\ 83.46\%}{89.62\%\ x\ 83.46\%} = 86.44\%$$

   F1-*Score* dengan nilai sebesar 86.44% ini menunjukkan adanya perpaduan yang baik antara presisi dan *recall* untuk kelas Negatif.

2. Kelas Positif

   a. *Recall (Sensitivitas)* untuk kelas Positif:

$$Recall\ Positif = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$Recall\ Positif = \frac{1333}{1333 + 12} = 99.10\%$$

Nilai *recall* sebesar ini berarti bahwa model berhasil mendeteksi sebagian besar *instance* yang sebenarnya adalah Positif.

b. *Presisi (Precision)* untuk kelas Positif:

$$Precision\ Positif = \frac{True\ Positives}{True\ Positivies + False\ Positives}$$

$$Precesion\ Positif = \frac{1333}{1333 + 154} = 89.62\%$$

Presisi sebesar 89.62% ini menunjukkan bahwa sebagian besar prediksi yang dibuat oleh model sebagai Positif adalah benar-benar Positif.

c. F1-Score untuk kelas Positif:

$$F1 - score\ Positif = \frac{2\ x\ Precision\ Positif\ x\ Recall\ Positif}{Precision\ Positif + Recall\ Positif}$$

$$F1 - score\ Positif = \frac{2 \text{ x } 89.62\% \text{ x } 99.10\%}{89.62\% \text{ x } 99.10\%} = 94.19\%$$

F1-*Score* dengan nilai sebesar 94.19% ini menunjukkan adanya perpaduan yang baik antara presisi dan *recall* untuk kelas Positif.

3. Kelas Netral

a. *Recall (Sensitivitas)* untuk kelas Netral:

$$Recall\ Netral = \frac{True\ Netral}{True\ Netral + False\ Negativies}$$

$$Recall\ Netral = \frac{15}{15 + 1} = 93.75\%$$

Nilai *recall* sebesar ini berarti bahwa model berhasil mendeteksi sebagian besar *instance* yang sebenarnya adalah Netral.

b. *Presisi (Precision)* untuk kelas Netral:

$$Precision\ Netral = \frac{True\ Netral}{True\ Netral + False\ Positives}$$

$$Precesion\ Netral = \frac{15}{15 + 154} = 8.82\%$$

Presisi sebesar 8.82% ini menunjukkan bahwa sebagian kecil prediksi yang dibuat oleh model sebagai Netral adalah benar-benar Netral.

c. F1-*Score* untuk kelas Netral:

$$F1 - score\ Netral = \frac{2\ x\ Precision\ Netral\ x\ Recall\ Netral}{Precision\ Netral + Recall\ Netral}$$

$$F1 - score\ Netral = \frac{2\ x\ 8.82\%\ x\ 93.75\%}{8.82\%\ x\ 93.75\%} = 16.05\%$$

F1-Score dengan nilai sebesar 16.05% ini menunjukkan adanya perpaduan yang baik antara presisi dan *recall* untuk kelas Netral.

Berdasarkan hasil ini, model memiliki akurasi sekitar 85.23% dengan nilai presisi, *recall*, dan *F1-score* yang berbeda-beda untuk setiap kelas sentimen (Negatif, Positif, Netral). Model tampaknya lebih baik dalam mengklasifikasikan kelas Positif dan Negatif daripada kelas Netral, yang memiliki *recall* yang lebih rendah. Dapat disimpulkan bahwa model memiliki kinerja yang baik dalam mendeteksi kelas Negatif dan Positif, dengan *F1-Score* yang tinggi untuk keduanya. Kinerja model dalam mendeteksi kelas Netral tidak sebaik kelas Negatif dan Positif. *F1-Score* untuk kelas Netral lebih rendah. Ini menunjukkan bahwa model memiliki kesulitan dalam mengidentifikasi atau memprediksi dengan tepat *instance* yang sebenarnya adalah Netral.

Hasil pengujian algoritma *Random Forest* dengan optimasi PSO sebagaiman tabel 4.16. berikut.

Tabel 4. 16. Hasil Pengujian *Random Forest* dengan Optimasi PSO

accuracy: 80.53% +/- 2.98% (micro average: 80.53%)				
	true Negatif	**true Positif**	**true Netral**	**class precision**
pred. Negatif	1148	228	26	81.88%
pred. Postif	325	1371	30	79.43%
pred. Netral	0	0	0	0.00%
class *recall*	77.94%	85.74%	0.00%	

Berdasarkan hasil pada tabel 4.16 terlihat bahwa dari hasil *confusion matrix accuracy* yang dihasilkan sebesar 80.53% yang menunjukkan bahwa sebagian besar sampel telah diklasifikasikan dengan benar. Perhitungan nilai *recall, precision,* dan *F1-Score* untuk masing-masing kelas sebagai berikut.

1. Kelas Negatif

   a. *Recall* (Sensitivitas) untuk kelas Negatif:

$$Recall\ Negatif = \frac{True\ Negatives}{True\ Negatives + False\ Negatives}$$

$$Recall\ Negatif = \frac{1371}{1371 + 228} = 85.75\%$$

   Nilai *recall* sebesar ini berarti bahwa model berhasil mendeteksi sebagian besar *instance* yang sebenarnya adalah Negatif.

   b. *Presisi (Precision)* untuk kelas Negatif:

$$Precision\ Negatif = \frac{True\ Negatives}{True\ Negatives + False\ Positives}$$

$$Precesion\ Negatif = \frac{1371}{1371 + 325} = 80.87\%$$

Presisi sebesar 80.87% ini menunjukkan bahwa sebagian besar prediksi yang dibuat oleh model sebagai Negatif adalah benar-benar Negatif.

c. F1-Score untuk kelas Negatif:

$$F1 - score\ Negatif = \frac{2\ x\ Precision\ Negatif\ x\ Recall\ Negatif}{Precision\ Negatif + Recall\ Negatif}$$

$$F1 - score\ Negatif = \frac{2\ x\ 80.87\%\ x\ 85.75\%}{80.87\%\ x\ 85.75\%} = 83.23\%$$

F1-*Score* dengan nilai sebesar 83.23% ini menunjukkan adanya perpaduan yang baik antara *presisi* dan *recall* untuk kelas Negatif.

2. Kelas Positif

a. *Recall (Sensitivitas)* untuk kelas Positif:

$$Recall\ Positif = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$Recall\ Positif = \frac{1371}{1371 + 30} = 97.86\%$$

Nilai *recall* sebesar ini berarti bahwa model berhasil mendeteksi sebagian besar *instance* yang sebenarnya adalah Positif.

b. *Presisi (Precision)* untuk kelas Positif:

$$Precision\ Positif = \frac{True\ Positives}{True\ Positivies + False\ Positives}$$

$$Precesion\ Positif = \frac{1371}{1371 + 325} = 80.87\%$$

Presisi sebesar 80.87% ini menunjukkan bahwa sebagian besar prediksi yang dibuat oleh model sebagai Positif adalah benar-benar Positif.

c. F1-Score untuk kelas Positif:

$$F1 - score\ Positif = \frac{2\ x\ Precision\ Positif\ x\ Recall\ Positif}{Precision\ Positif + Recall\ Positif}$$

$$F1 - score\ Positif = \frac{2 \text{ x } 80.87\% \text{ x } 97.86\%}{80.87\% \text{ x } 97.86\%} = 88.65\%$$

F1-*Score* dengan nilai sebesar 88.65% ini menunjukkan adanya perpaduan yang baik antara presisi dan *recall* untuk kelas Positif.

3. Kelas Netral

   Pada Kelas Netral, model tidak dapat memprediksi sehingga tidak dapat dihitung nilai *recall*, *Precision* dan *F1-score* nya.

Berdasarkan hasil tersebut dapat disimpulkan bahwa kelas Negatif memiliki kinerja yang baik, dengan *Recall* sebesar 85.75%, *Precision* juga baik dengan nilai 80.87%, *dan F1-Score* sebesar 83.23%. Ini menunjukkan bahwa model dapat dengan baik mendeteksi kelas Negatif. Kelas Positif juga memiliki kinerja yang baik, dengan *Recall* yang sangat tinggi yaitu 97.86%, *Precision* juga baik dengan nilai 80.87%, dan *F1-Score* sebesar 88.65%. Model dapat dengan sangat baik mendeteksi Positif dengan nilai *Recall* lebih tinggi dibandingkan dengan kelas Negatif. Pada kelas Netral tidak dapat dievaluasi dengan metrik *Recall*, *Precision*, dan *F1-Score* karena tidak ada *True Netral,* yang berarti *Recall* adalah 0%. Hal ini menunjukkan bahwa model tidak dapat mendeteksi dengan baik kelas Netral.

4.2.2. Komparasi Hasil Pengujian

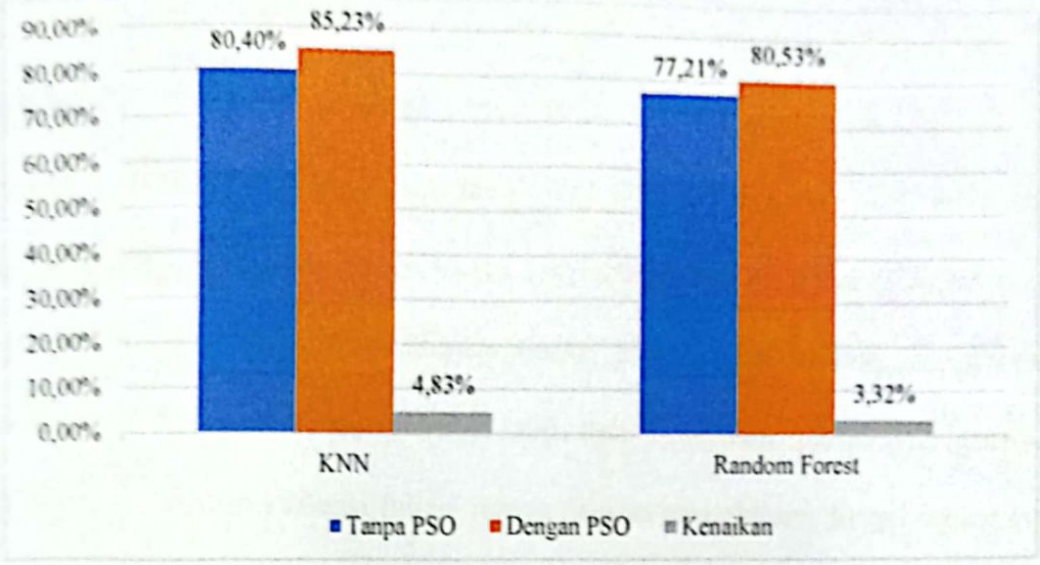
Untuk menjawab pertanyaan penelitian yang kedua yaitu algoritma manakah yang memiliki akurasi lebih tinggi antara KNN dan *Random Forest* berbasis PSO, disajikan hasil pengujian sebagaimana tabel 4.17 berikut ini.

Tabel 4. 17. Komparasi Hasil Pengujian *Random Forest* dengan PSO

Metode	Akurasi Prediksi Tanpa PSO	Akurasi Prediksi Dengan PSO
K-Nearest Neighbors	80.40%	85.23%
Random Forest	77.21%	80.53%

Berdasarkan tabel 4.17 di atas terlihat bahwa nilai akurasi dari algoritma KNN lebih tinggi dibandingkan dengan algoritma *Random Forest.* Terdapat perbedaan akurasi antara sebelum dan sesudah dilakukan optimasi. Penggunaan PSO secara signifikan meningkatkan akurasi prediksi sebesar sekitar 4.83% pada metode KNN dan peningkatan sebesar 3.32%. Ini menunjukkan bahwa PSO berhasil menemukan parameter atau konfigurasi yang lebih optimal untuk model KNN dalam *dataset* tertentu. Oleh karena itu, PSO dapat dianggap sebagai teknik yang efektif dalam meningkatkan kinerja yang positif terhadap kedua metode.

Dengan penerapan PSO, akurasi prediksi KNN meningkat secara signifikan dari 80.40% menjadi 85.23%. Dengan demikian terbukti bahwa PSO telah membantu meningkatkan kualitas prediksi KNN dalam konteks ini. Demikian pula, *Random Forest* juga mengalami peningkatan dalam akurasi prediksi ketika PSO diterapkan. Akurasi meningkat dari 77.21% menjadi 80.53% dengan adanya PSO. Peningkatan nilai akurasi menunjukkan efektivitas dari proses optimasi dalam meningkatkan performa algoritma *Random Forest* terhadap tweet tentang Perppu Cipta Kerja. Penggunaan PSO juga menghasilkan peningkatan akurasi prediksi, meskipun peningkatannya lebih kecil dibandingkan dengan KNN, yaitu sekitar 3.32%. Ini menunjukkan bahwa PSO tetap berperan dalam meningkatkan kualitas model *Random Forest*, tetapi efeknya tidak sekuat pada KNN. Grafik perbandingan hasil akurasi kedua model sebagaimana gambar 4.18. berikut.



Gambar 4. 18. Perbandingan Hasil Akurasi

Secara umum, penggunaan PSO dalam kombinasi dengan metode KNN dan *Random Forest* telah membantu meningkatkan akurasi prediksi dalam kedua kasus. Hal ini menunjukkan potensi besar dari teknik optimasi seperti PSO dalam meningkatkan kualitas hasil dari algoritma *machine learning*. Dalam penelitian ini model KNN dan *Random Forest* dapat mengklasifikasikan data dengan akurat sebelum dilakukan optimasi maupun setelah dilakukan optimasi. Hal ini berarti bahwa *classifier* terbaik dalam menentukan akurasi klasifikasi analisis sentimen pengguna Twitter terhadap Peraturan Pemerintah Pengganti Undang-Undang (PERPPU) Tentang Cipta Kerja ini adalah KNN berbasis PSO. KNN mampu mengklasifikasikan data dengan baik. Kenaikan akurasi yang signifikan diperoleh sebagai dampak dari optimasi yang dilakukan dan menunjukkan kinerja dari metode yang digunakan. Berdasarkan hal ini, dapat dipilih metode yang sesuai dan dioptimalkan penggunaanya dalam melakukan analisis sentimen pengguna Twitter terhadap Perppu Cipta Kerja.

Beberapa langkah yang dilakukan oleh PSO untuk melakukan optimasi yaitu: *Pertama* menginisialisasi jumlah partikel dalam ruang pencarian. Setiap partikel mewakili kemungkinan solusi di wilayah yang akan dioptimalkan. *Kedua,* pengaturan posisi awal dan kecepatan yang ditentukan secara acak dalam ruang pencarian. Posisi partikel mencerminkan solusi yang sedang dieksplorasi, sementara kecepatan menunjukkan arah dan kecepatan pergerakan partikel. *Ketiga*, melakukan evaluasi fungsi tujuan dengan menghitung fungsi tujuan atau nilai kriteria yang akan dioptimalkan. Nilai ini menunjukkan seberapa baik partikel mendekati solusi optimal. *Keempat*, pembaruan lokasi partikel berdasarkan pengalaman pribadi (pBest) dan pengalaman kelompok (gBest). PBest adalah solusi terbaik yang pernah ditemukan oleh partikel itu sendiri, sedangkan gBest adalah solusi terbaik yang pernah ditemukan oleh seluruh kelompok partikel. *Kelima*, partikel mengubah kecepatan dan arahnya berdasarkan pBest dan gBest menuju solusi maksimal.

Langkah selanjutnya adalah iterasi dari langkah ketiga sampai kelima. Partikel terus bergerak dan mengoptimalkan posisinya berdasarkan pengalaman individu dan kelompok. Algoritma akan berhenti ketika mencapai kriteria penghentian yang ditentukan, seperti batas iterasi maksimum atau tingkat akurasi yang diinginkan. Hasil akhir dari PSO adalah solusi yang nilai fungsi tujuannya mendekati nilai optimal.

Selanjutnya, untuk melihat tingkat persetujuan antara pengamat-pengamat, secara substansial dapat dilihat dari nilai Kappa. Dalam konteks penelitian ini, nilai Kappa merupakan ukuran statistik yang digunakan untuk menilai tingkat

kesepakatan antara dua atau lebih *anotator* dalam analisis sentimen. Nilai Kappa berkisar antara -1 dan 1, di mana nilai negatif menunjukkan kesepakatan yang lebih rendah dari yang diharapkan secara acak, nilai nol menunjukkan kesepakatan acak, dan nilai positif menunjukkan kesepakatan yang lebih tinggi dari yang diharapkan secara acak. Nilai Kappa yang tinggi menunjukkan bahwa *anotator* memiliki kriteria yang konsisten dan objektif dalam menetapkan label sentimen kepada teks.

Adapun hasil data Kappa disajikan dalam tabel 4.18 berikut ini.

Tabel 4. 18. Hasil Pengujian Kappa

Metode	Nilai Kappa Sebelum PSO	Nilai Kappa Setelah PSO
K-Nearest Neighbors	0.616	0.712
Random Forest	0.548	0.615

Berdasarkan tabel 4.18, terlihat bahwa nilai Kappa pada metode KNN mengalami peningkatan yang signifikan setelah adanya penerapan PSO. Terlihat bahwa nilai Kappa meningkat secara signifikan dari 0.616 menjadi 0.712 setelah penerapan PSO. Nilai Kappa yang semakin mendekati 1 mengindikasikan tingkat kesepakatan yang lebih tinggi antara prediksi model dengan data aktual. Oleh karena itu, penerapan PSO secara positif mempengaruhi kemampuan model KNN dalam memberikan prediksi yang lebih akurat.

Setelah diterapkan PSO, terjadi peningkatan sebesar 0.096 (9.6%) yaitu dari 0.616 menjadi 0.712 dalam klasifikasi *good*. Peningkatan ini menunjukkan bahwa PSO mampu mengoptimalkan kinerja KNN dalam mengklasifikasikan data sehingga tingkat persetujuan antar pengamat meningkat secara substansial.

Pada *Random Forest*, terlihat bahwa nilai Kappa juga mengalami peningkatan yang signifikan dari 0.548 menjadi 0.615 setelah penerapan PSO yaitu pada klasifikasi *moderate* menjadi *good*. Dalam hal ini mengalami kenaikan sebesar 0.067 (6.7%). Dengan demikian dapat disarikan bahwa PSO berhasil meningkatkan tingkat kesepakatan antara prediksi model *Random Forest* dengan kenyataan. Meskipun peningkatannya tidak sebesar pada KNN, hasil ini masih mengindikasikan perbaikan yang signifikan dalam kualitas prediksi.

Berdasarkan hasil di atas, nilai Kappa yang semakin mendekati 1 (satu) merupakan indikasi bahwa model memiliki kemampuan yang lebih baik dalam melakukan prediksi yang konsisten dan akurat. Peningkatan nilai Kappa yang dicapai setelah penerapan PSO menunjukkan bahwa PSO berhasil menemukan parameter atau konfigurasi yang lebih baik untuk kedua metode yaitu KNN dan *Random Forest* dalam dataset yang digunakan. Hasil ini memberikan dukungan bagi penggunaan PSO sebagai alat optimasi yang efektif dalam meningkatkan kemampuan model *machine learning* khususnya dalam konteks penelitian ini.

Secara keseluruhan, hasil ini menunjukkan bahwa penerapan PSO telah memberikan perbaikan yang signifikan dalam kualitas prediksi untuk kedua metode KNN dan *Random Forest*, sebagaimana tercermin dalam peningkatan nilai Kappa. Dapat disimpulkan bahwa secara keseluruhan hasil pengujian menunjukkan bahwa optimasi PSO memiliki dampak positif pada metode KNN dan *Random Forest* dalam hal meningkatkan Kappa dalam konteks penelitian ini.

Hasil penelitian ini sejalan dengan penelitian sebelumnya yang dilakukan oleh Louis Madaerdo Sotarjua dan Dian Budhi Santoso [14] dengan judul

Perbandingan Algoritma KNN, *Decision Tree*, dan *Random Forest* Pada Data *Imbalanced Class* Untuk Klasifikasi Promosi Karyawan menunjukkan bahwa hasil performa model klasifikasi model KNN memiliki hasil performa yang terbaik nilai metriks evaluasinya dibandingkan dengan *Decision Tree,* dan *Random Forest*. Hasil masing-masing *accuracy* yaitu KNN sebesar 86.57%, *Decision Tree,* 85.29%, dan *Random Forest* sebesar 86.37%.

Begitu juga dengan hasil penelitian yang dilakukan oleh Aprilia Wandani, Fauziah, dan Andrianingsih [2] pada Sentimen Analisis Pengguna Twitter pada Event Flash Sale diperoleh hasil bahwa dari perbandingan tiga algoritma klasifikasi Naïve Bayes, KNN, dan *Random Forest*, akurasi dari *Random Forest* sebesar 80.59%. Hasil ini lebih rendah dibandingkan dengan KNN dengan nilai sebesar 82.94%. Hasil *Random Forest* yang lebih rendah dari KNN juga ditunjukkan oleh oleh Muhamad Fahmi Fakhrezi dkk [36] yang menunjukkan bahwa metode *Naïve Bayes* memiliki nilai akurasi tertinggi sebesar 95.45%. Sementara itu, metode *Decision Tree* hanya mendapatkan nilai akurasi sebesar 56.83%, KNN mendapatkan nilai akurasi sebesar 74.08% yang merupakan metode yang cocok kedua setelah metode *Naïve Bayes,* dan metode *Random Forest* hanya mendapatkan nilai akurasi sebesar 57.50%