NAMED ENTITY RECOGNITION MENGGUNAKAN METODE *CONDITIONAL RANDOM FIELDS* UNTUK DETEKSI PERISTIWA BANJIR DI GERBANG KERTOSUSILA BERDASARKAN DATA TWITTER

SKRIPSI

UIN SUNAN AMPEL
S U R A B A Y A

Disusun Oleh:

IKRIMATUL ULUMIYYAH
NIM. H06216010

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL
SURABAYA

2022

HALAMAN PERNYATAAN KEASLIAN KARYA

Saya yang bertanda tangan di bawah ini,

NAMA	: IKRIMATUL ULUMIYYAH
NIM	: H06216010
PROGRAM STUDI	: SISTEM INFORMASI
ANGKATAN	: 2016

Menyatakan bahwa saya tidak melakukan plagiat dalam penulisan skripsi saya yang berjudul: "NAMED ENTITY RECOGNITION MENGGUNAKAN METODE CONDITIONAL RANDOM FIELDS UNTUK DETEKSI PERISTIWA BANJIR DI GERBANG KERTOSUSILA BERDASARKAN DATA TWITTER". Apabila suatu saat nanti terbukti saya melakukan tindakan plagiat, maka saya bersedia menerima sanksi yang telah ditetapkan.

Demikian pernyataan keaslian ini saya buat dengan sebenar-benarnya.

Surabaya, 4 Agustus 2022
Yang Menyatakan

IKRIMATUL ULUMIYYAH
NIM. H06216010

LEMBAR PERSETUJUAN PEMBIMBING

Skripsi Oleh:

NAMA : IKRIMATUL ULUMIYYAH

NIM : H06216010

JUDUL : *NAMED ENTITY RECOGNITION* MENGGUNAKAN METODE *CONDITIONAL RANDOM FIELDS* UNTUK DETEKSI PERISTIWA BANJIR DI GERBANG KERTOSUSILA BERDASARKAN DATA *TWITTER*

Ini telah diperiksa dan disetujui untuk diujikan.

Surabaya, 4 Agustus 2022

Dosen Pembimbing 1	Menyetujui, Dosen Pembimbing 2
Muhammad Andik Izzuddin, MT	Dwi Rolliawati, MT
NIP. 98403072014031001	NIP.197909272014032001

PENGESAHAN TIM PENGUJI SKRIPSI

Skripsi Ikrimatul Ulumiyyah ini telah dipertahankan
di depan tim penguji skripsi di Surabaya, 8 Agustus 2022.

Mengesahkan,

Dewan Penguji

Penguji 1

Dr. Eng. Anang Kunaefi, M.Kom
NIP. 197911132014031001

Penguji 2

Mujib Ridwan, S.Kom., M.T
NIP. 198604272014031004

Penguji 3

Muhammad Andik Izzuddin, MT
NIP. 98403072014031001

Penguji 4

Dwi Rolliawati, MT
NIP.197909272014032001

Mengetahui,
Dekan Fakultas Sains dan Teknologi
UIN Sunan Ampel Surabaya

Dr. A. Saepul Hamdani, M.Pd
NIP. 196507312000031002

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA
PERPUSTAKAAN
Jl. Jend. A. Yani 117 Surabaya 60237 Telp. 031-8431972 Fax.031-8413300
E-Mail: perpus@uinsby.ac.id

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika UIN Sunan Ampel Surabaya, yang bertanda tangan di bawah ini, saya:

Nama : IKRIMATUL ULUMIYYAH
NIM : H06216010
Fakultas/Jurusan : FAKULTAS SAINS DAN TEKNOLOGI / SISTEM INFORMASI
E-mail address : ikrimatul@gmail.com / H06216010@uinsby.ac.id

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan UIN Sunan Ampel Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah :
☒ Sekripsi ☐ Tesis ☐ Desertasi ☐ Lain-lain (……)
yang berjudul :
NAMED ENTITY RECOGNITION MENGGUNAKAN METODE *CONDITIONAL RANDOM FIELDS* UNTUK DETEKSI PERISTIWA BANJIR DI GERBANG KERTOSUSILA BERDASARKAN DATA TWITTER

beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Ekslusif ini Perpustakaan UIN Sunan Ampel Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain secara ***fulltext*** untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan atau penerbit yang bersangkutan.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Perpustakaan UIN Sunan Ampel Surabaya, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Surabaya, 25 Agustus 2022

Penulis

(. Ikrimatul Ulumiyyah)

ABSTRACT

NAMED ENTITY RECOGNITION USING CONDITIONAL RANDOM FIELDS METHOD FOR FLOOD DETECTION IN GERBANG KERTOSUSILA BASED ON TWITTER DATA

Oleh :
Ikrimatul Ulumiyyah

Every rainy season, several regions in Indonesia are urged to be alert for flooding, such as the Gerbang Kertosusila national strategic area. One of the existing efforts is to place flood sensors at several flood-prone points. However, it is constrained by minimal equipment to handle the many areas in need. So it is necessary to develop technology for the dissemination of flood information. Dissemination of flood information was quickly obtained from social media Twitter. One way is to use Twitter's text data source for a Named Entity Recognition-based detection model to help detect flood events and their location. To achieve this goal, a Named Entity Recognition (NER) model build using the Conditional Random Fields (CRFs) method. This research adds the handling of slang word handles at the preprocessing phase to increase model performance and use the BIO format in the labeling process and POS Tagging in the Feature Extraction process. The evaluation results with the kfold five scenario, 80% training data, and 20% testing data show that the NER CRFs model has excellent performance with a Precision value of 0.981, Recall 0.926, and f-measure 0.950. So this result can help the community and the government related to flood distribution information.

Keyword: Flood Detection, Natural Language Processing, Named Entity Recognition, Conditional Random Fields

ABSTRAK

NAMED ENTITY RECOGNITION MENGGUNAKAN METODE *CONDITIONAL RANDOM FIELDS* UNTUK DETEKSI PERISTIWA BANJIR DI GERBANG KERTOSUSILA BERDASARKAN DATA TWITTER

Oleh :
Ikrimatul Ulumiyyah

Setiap musim hujan sejumlah wilayah di Indonesia dihimbau untuk waspada terjadinya banjir seperti kawasan strategis nasional Gerbang Kertosusila, Salah satu upaya yang ada adalah meletakkan sensor banjir di beberapa titik rawan banjir. Namun terkendala perangkat yang sangat terbatas untuk menangani banyaknya wilayah yang membutuhkan. Sehingga diperlukan pengembangan teknologi tentang penyebaran informasi banjir. Penyebaran informasi banjir dengan cepat didapatkan dari media sosial *Twitter.* Salah satu caranya memanfaatkan sumber data teks *Twitter* untuk model deteksi berbasis *Named Entity Recognition* untuk membantu mendeteksi peristiwa banjir dan lokasinya. Agar tujuan tersebut bisa tercapai, dibuatlah model *Named Entity Recognition* (NER) dengan metode *Conditional Random Fields* (CRFs). Riset ini menambahkan penanganan *handle slang word* pada tahap preprosessing untuk memaksimalkan performa model, Sekaligus menggunakan format *BIO* pada proses *labellin*g dan *POS Tagging* dalam proses *Extraction Feature*. Hasil evaluasi dengan skenario kfold = 5, 80% data *training* dan 20% data *testing* menunjukkan model *NER CRFs* memiliki performa yang sangat baik dengan nilai *Precision* 0.981, *Recall* 0.926 dan *f-measure 0.*950*.* Sehingga dengan hasil ini dapat membantu masyarakat dan pemerintahaan terkait informasi distribusi banjir.

Kata Kunci: Deteksi Banjir, Natural Language Processing, *Named Entity Recognition, Conditional Random Fields*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR PSEUDOCODE

BAB I
PENDAHULUAN

1.1 Latar Belakang

Peristiwa Banjir merupakan bencana alam hidrometeorologi yang sering terjadi di Indonesia maupun Global. Banjir dapat disebabkan oleh curah hujan dan kelalaian manusia pada proses pembuangan sampah. Akibat curah hujan intens dan berdurasi lama sejumlah wilayah Indonesia dihimbau untuk waspada terhadap bencana hidrometeorologi. Kawasan Gresik, Bangkalan, Mojokerto, Surabaya, Sidoarjo dan Lamongan (Gerbang Kertosusila) di Provinsi Jawa timur salah satu kawasan strategis nasional. Gerbang Kertosusila memiliki iklim tropis dengan rata – rata temperatur 28,5%, kelembaban udara rata – rata 75% dan Curah hujan yang rendah dengan rata – rata 1.290,50 mm per tahun (Sifataru, 2019).

Namun Badan Nasional Penanggulangan Bencana Daerah Jawa Timur menyatakan Jawa Timur tercatat sebanyak 65 kejadian bencana sepanjang awal Januari 2021 (CNN Indonesia, 2021). Bencana yang terjadi didominasi oleh peristiwa banjir sebanyak 49 kejadian. Banjir mempunyai dampak buruk terhadap lingkungan, dan menimbulkan penurunan di sektor ekonomi (Baranowski dkk., 2020; Riny Sulistyowati, Hari Agus Sujono, 2015). Selain itu banjir memberikan dampak terhadap kesehatan masyarakat, termasuk meningkatnya penyakit psikologis manusia (Mutawalli dkk., 2020).

Dampak peristiwa banjir yang tidak ditanggulangi secara cepat akan memperbesar kerugian masyarakat dan pemerintahan. Banyak upaya yang dilakukan untuk mempermudah masyarakat dalam mendeteksi banjir. Salah satunya menggunakan cara konvensional yaitu meletakkan sensor deteksi banjir. Namun perangkat yang dimiliki sangat terbatas dan tidak cukup untuk menangani banyak wilayah yang membutuhkan (Utami dan Marzuki, 2020). Selain itu, cara tersebut cenderung memakan biaya yang besar. Sehingga diperlukan terobosan yang baru agar informasi dan penanganan banjir bisa lebih cepat. Yaitu dengan memanfaatkan media sosial *Twitter*.

Twitter pantas menyandang *microblogging* populer. Faktanya *Twitter* mempunyai 353 Juta pengguna di dunia dengan *tweets* lebih dari 500 Juta perhari.

Fakta lainnya, pada laporan *We are social* dan *Hootsuite* mengungkapkan jumlah pengguna *twitter* di Indonesia mencapai 14,05 Juta Pengguna. Pengguna *Twitter* dibedakan menjadi dua, yaitu pengguna umum dan *user influence* (Awalludin dkk., 2018). Tidak hanya bertukar informasi, *Twitter* dimanfaatkan untuk ajang curhat, protes, media Pendidikan dan kampanye. Hal tersebut menjadikan banyak *user influence* memilih Twitter sebagai sarana menyampaikan berita suatu kejadian. Seperti *event*, konser music, lalu lintas bencana longsor, gempa bumi dan banjir. Terkait hal itu memungkinkan *Twitter* untuk dimanfaatkan sebagai sumber informasi studi penelitian, seperti teknologi pencarian, menganalisa sentimen terhadap topik penelitian, pengukuran validitas pengguna, informasi pemetaan lalu lintas dan pengamatan bencana (Awalludin dkk., 2018). Pendekatan *Text Mining* bisa digunakan untuk ekstraksi informasi dalam jumlah besar dan tidak terstruktur. Pada pendekatan *Text Mining* terdapat teknik *Natural Language Processing* (NLP) untuk ekstraksi dan menyaring informasi yang relevan dengan banjir.

Proses penting ekstraksi informasi pada teknik NLP adalah *Named Entity Recognition* (NER). Tujuan NER adalah mengidentifikasi nama – nama *entity* kedalam kelompok label terstruktur. Beberapa pendekatan studi penelitian yang mempelajari NER dikelompokkan menjadi tiga, yaitu *rule based*, *machine learning*, dan *deep learning* (Sun dkk., 2019). Studi penelitian sebelumnya NER diimplementasikan untuk ekstraksi informasi lokasi kejadian lalu lintas dengan pendekatan *rule based* (Ermawati dan Buliali, 2018; Yuda Munarko dkk., 2015).

Conditional Random Fields (CRFs) merupakan pendekatan *probabilistic* untuk segmentasi, pelabelan pada sekuen data, seperti urutan, pohon atau kisi. CRFs digunakan untuk menghitung nilai probabilitas bersyarat pada node output yang didesain dengan node input yang didesain. Kemudian model probabilitas urutan tersebut digunakan untuk mendeteksi *entity* secara otomatis. Seperti penelitian yang telah dilakukan dalam pengenalan *entity* bernama pada teks Bahasa Indonesia oleh (Jaariyah dan Rainarli, 2017). Jaariyah memodelkan NER menggunakan CRFs dengan memberikan tingkat akurasi terbaik sebesar 87,06%. Dilanjutkan pada penelitian pengenalan *entity* data *Twitter* berbahasa Indonesia (Y Munarko dkk., 2018). Monarko melakukan pengujian model pada tiga data uji yaitu, formal,

informal dan *mixed tweets* dan menemukan nilai *precision* yang tinggi untuk pengujian pada semua data uji. Namun untuk mendapatkan nilai *precision* yang tinggi, proses pengujian bergantung pada pemilihan model yang tepat terhadap data uji. Dari beberapa penelitian tersebut dapat diketahui keunggulan metode CRFs. CRFs mampu menentukan banyaknya fitur yang diperlukan untuk membangun sebuah model, berbeda dengan model *Hidden Markov Model* yang bersifat lokal dan setiap kata hanya bergantung pada label saat ini dengan setiap label sebelumnya (Jaariyah dan Rainarli, 2017). Dengan kemampuan tersebut metode CRFs mampu mengatasi tingginya tingkat asumsi yang terjadi pada metode Hidden Markov Model. Sebagian besar penelitian yang berhasil dilakukan, penerapan NER menggunakan CRFs membutuhkan *feature vector* (untuk mempermudah proses feature extraction) seperti fitur Bahasa, *POS Tagging*, *Morphological Analyze*, *Gazetteers* dan *NE annotated corpus* (Patil dkk., 2020).

Berdasarkan permasalahan diatas penelitian dengan judul "*Named Entity Recognition* Menggunakan Metode *Conditional Random Fields* Untuk Deteksi Peristiwa Banjir di Gerbang Kertosusila Berdasarkan Data *Twitter*" menawarkan solusi mempermudah masyarakat mengetahui kejadian banjir di Kawasan Gerbang Kertosusila. Teknik NER dilibatkan untuk membantu proses deteksi *entity* bernama berupa kejadian peristiwa dan lokasi kejadian peristiwa pada *Tweets*. Metode yang digunakan adalah CRFs dan *Feature Vector Post Tagging*. Dengan mempertimbangkan penggunaan *Conditional Random Fields* sebelumnya berhasil dilakukan oleh (Jaariyah dan Rainarli, 2017; Muhammad dan Khodra, 2015) yang menggunakan *POS Tagging* pada pelabelan kelas kata sebagai *feature vector* untuk meningkatkan *accuracy* dan *performance* pengenalan *entity*.

1.2 Rumusan Masalah

Sesuai latar belakang permasalahan rumusan masalah yang akan dibahas yaitu:

1. Bagaimana penyajian hasil deteksi *Named Entity Recognition* banjir di Gerbang Kertosusila ?
2. Bagaimana pengujian dan evaluasi *Named Entity Recognition* menggunakan metode *Conditional Random Fields* untuk deteksi banjir?

1.3 Batasan Masalah

Batasan masalah sengaja dibuat untuk menjaga penelitian tidak melebihi batas inti permasalahan, berikut paparan batasan masalah.

1. Dataset topik *tweets* yang dideteksi diasumsikan sebagai kejadian bencana *banjir*
2. *Tweets* kejadian banjir yang terjadi di wilayah Jawa Timur Indonesia, khususnya kawasan Gerbang Kertosusila.
3. *Named Entity Recognition* yang memuat label *Event* dan *Location*.

1.4 Tujuan Penelitian

Penelitian dilakukan memiliki maksud dan tujuan yang dipaparkan sebagai berikut:

1. Menyajikan hasil deteksi peristiwa banjir di Gerbang Kertosusila
2. Melakukan evaluasi penerapan *Named Entity Recognition* yang menggunakan metode *Conditional Random Field*.

1.5 Manfaat Penelitian

Penelitian bermanfaat secara aplikatif dan akademik. Manfaat penelitian dipaparkan sebagai berikut

1.5.1 Aplikatif

1. Penelitian ini memberikan informasi kejadian banjir bagi masyarakat di Kawasan Gerbang Kertosusila.
2. Penelitian ini menyajikan lokasi kejadian banjir, dengan data lokasi tersebut akan mempermudah pemerintah melakukan keputusan dan tindakan penanggulangan banjir.

1.5.2 Akademik

1. Penelitian ini memberikan informasi kejadian banjir bagi masyarakat di Kawasan Gerbang Kertosusila.
2. Penelitian ini menyajikan lokasi kejadian banjir, dengan data lokasi tersebut akan mempermudah pemerintah melakukan keputusan dan Tindakan penanggulangan banjir.

1.6 Sistematika Penulisan

1. BAB I PENDAHULUAN

   Pada penelitian ini menjelaskan permasalahan penelitian pada bab I yaitu, bagaimana mendeteksi peristiwa banjir di Gerbang Kertasusila karena terbatasnya perangkat sensor yang dimiliki, tujuan penelitian, batasan dan manfaat dari penelitian yang dilakukan

2. BAB II TINJAUAN PUSTAKA

   Bab yang berisi penjelasan dari penelitian terdahulu yang relevan dengan penelitian yang dilakukan, dasar teori yang digunakan, seperti konsep sistem informasi, NER, dan Metode CRFs serta integrasi keilmuan yaitu bagaimna sudut pandang islam dalam upaya deteksi peristiwa banjir.

3. BAB III METODE PENELITIAN

   Pada bab ketiga berisi penjelasan tentang rangkaian tahapan atau langkah - langkah yang logis dan terstruktur dalam menyelesaikan penelitian yang dilakukan, yaitu tahap perumusan masalah, studi literatur, pengumpulan data, pembuatan model NER menggunakan metode CRFs, pengembangan sistem dan evaluasi system.

4. BAB IV HASIL DAN PEMBAHASAN

   Bab empat menjadi dokumentasi penting dalam penelitian skripsi yaitu berisi paparan hasil dan pembahasan dari penelitian tentang NER menggunakan CRFs untuk deteksi peristiwa banjir di Gerbangkertosusila.

5. BAB V PENUTUP

   Bab lima adalah bab terakhir yang berisi kesimpulan hasil dan pembahasan dari penelitian ini. Terdapat saran dan masukan pengembangan untuk penelitian yang akan dilakukan di masa depan.

BAB II
KAJIAN PUSTAKA

Kajian pustaka menjadi dasar wawasan penelitian *Named Entity Recognition* (NER) dan mengintegrasi dasar keilmuan penelitian dengan keislaman.

2.1 Penelitian Terdahulu

Tabel 2.1 pemaparan penelitian terdahulu menjadi sumber wawasan dan landasan teori terkait *Named Entity Recognition* (NER) berbahasa Indonesia.

Tabel 2.1 Penelitian Terdahulu

No	Topik	Hasil dan Pembahasan	GAP
1.	Conditional Random Fields Untuk pengenalan entitas bernama pada text bahasa Indonesia (Jaariyah dan Rainarli, 2017)	Pengenalan entitas pada text Bahasa Indonesia yang dilakukan menggunakan model NER CRF yang menghasilkan akurasi pengukuran 90,53% pada fitur 1 dan 89,71% pada fitur 2 dan 89.13% pada fitur 3. Tahapan CRF yang dilakukan pada penelitian ini adalah penaksiran parameter dan inferensi. Menggunakan IPOSTagger hasil penelitian Alfan Wicaksono dan Stanford NER	Diharapkan dapat diimplementasikan untuk pengenalan entitas dengan studi kasus mendeteksi peristiwa banjir – dan penggunaan Korpus Post Tagging Indonesia
2.	*Named Entity Recognition* model for Indonesian tweet using CRF classifier (Y Munarko dkk., 2018)	Penelitian ini menggunakan dataset twitter berbahasa indonesia, Data yang diberi tag dan diolah menggunakan CRF classifier, menghasilkan tiga model. Untuk menghasilkan model tersebut, menggunakan Stanford NER yang telah mengimplementasikan pengklasifikasi CRF	
3.	Event Information Extraction from Indonesian Tweets using Conditional Random Field (Muhammad dan Khodra, 2015)	Event Information Extraction from Indonesian Tweets using Conditional Random Field	

Tabel 2.2 memanfaatkan NER untuk deteksi peristiwa dapat dilakukan dengan berbagai metode baik *rule based* atau *machine learning*. Pengembangan yang bisa dilakukan salah satunya adalah menerapkan metode CRFs pada model NER untuk mendeteksi kejadian banjir di kawasan Gerbang Kertasusila dan memanfaatkan media twitter sebagai sumber data penelitian. Penelitian ini diharapkan memberikan kontribusi terkait pengembangan NER dengan Metode CRFs untuk deteksi peristiwa banjir.

Tabel 2.2 Penelitian Terdahulu Lanjutan

No.	Topik	Hasil dan Pembahasan	GAP
1.	Ekstraksi Nama Lokasi dari Tweets Informasi lalu lintas (Yuda Munarko dkk., 2015)	Penggunaan NER rule based dan stanford NER mampu mengidentifikasi nama lokasi dari data twitter, dataset yang digunakan adalah 1500 tweet dari masing - masing akun 500 tweet (RTMC_jatim, Sby *Traffic service* dan Radio Surabaya)	Terbatasnya penelitian ini masih memanfaatkan model NER *Rule Based*
2.	*Text Based Approach for traffic incident Detection from Twitter* (Ermawati dan Buliali, 2018)	Dataset yang digunakan sebanyak 2360 tweet dengan penggunaan NER Word Dictionary memberikan representasi yang lebih baik, dimana penelitian mempertimbangkan tweet yang serupa, yaitu memiliki konten insiden lalu lintas yang sama, dengan informasi insiden lalu lintas tertentu	Dapat dilakukan pengembangan lanjutan pada NER dengan pendekatan selain rule based dan pada kejadian peristiwa lainya.

Tabel 2.2 memanfaatkan NER untuk deteksi peristiwa dapat dilakukan dengan berbagai metode baik *rule based* atau *machine learning*. Pengembangan yang bisa dilakukan salah satunya adalah menerapkan metode CRFs pada model NER untuk mendeteksi kejadian banjir di kawasan Gerbang Kertasusila dan memanfaatkan media twitter sebagai sumber data penelitian. Penelitian ini diharapkan memberikan kontribusi terkait pengembangan NER dengan Metode CRFs untuk deteksi peristiwa banjir.

2.2 Peristiwa Banjir

Peristiwa banjir menjadi salah satu bencana alam yang sering terjadi secara global yang mengakibatkan kerusakan pada sosial, ekonomi dan Kesehatan masyarakat. Kerentanan populasi terhadap dampak buruk peristiwa banjir bergantung pada kondisi alam, serta tingkat perkembangan sosial dan ekonomi. Wilayah Indonesia terletak di daerah iklim tropis dengan dua musim yaitu musim

panas dan hujan. Adanya ciri – ciri perubahan cuaca, suhu dan arah angin yang ekstrim dapat menimbulkan akibat buruk seperti bencana hidrometeorologi seperti banjir, tanah longsor, kebakaran hutan dan kekeringan Gambar 2.1 menunjukan infografis bencana di Indonesia sepanjang tahun 2020.

Gambar 2.1 Peta Kejadian Banjir (BNPB, 2021)

Gambar 2.1 Peristiwa banjir menjadi bencana alam yang paling tinggi terjadi sepanjang tahun 2020. Peristiwa banjir terjadi secara merata di Wilayah Indonesia baik di awal tahun, pertengahan dan akhir tahun. Ancaman banjir semakin besar, bahkan bagi daerah perkotaan yang semakin padat karena faktor populasi penduduk, ekonomi maupun perubahan iklim (Baranowski dkk., 2020). Sebagaimana Kawasan Gerbang Kertosusila sebagai Kawasan strategis nasional. Banjir menjadi permasalahan rutin, baik akibat curah hujan dengan intensitas tinggi ataupun kelalaian manusia pada pengolahan sampah.

Dalam berita CNN Indonesia, BNPB Jawa Timur telah mencatat sebanyak 49 Peristiwa banjir sepanjang Januari 2021. Banjir dapat menimbulkan kerusakan lingkungan, menghambat kegiatan masyarakat dan menurunnya perekonomian pemerintah. Dampak lain peristiwa banjir dapat mempengaruhi Kesehatan fisik seperti penyakit kulit, ataupun secara psikologi masyarakat (Mutawalli dkk., 2020). Dampak banjir akan semakin besar jika tidak segera ditangani. Banyak upaya yang dilakukan untuk mempermudah masyarakat dalam mendeteksi dan menangani banjir. Seperti sistem peringatan dini yang memanfaatkan sistem sensor deteksi

banjir, Sistem Koordinasi Pemerintah setempat dengan masyarakat secara kolekting. Namun cara konvensional ini membutuhkan jumlah yang banyak sensor, dan memerlukan biaya yang tinggi (Utami dan Marzuki, 2020).

2.3 *Twitter*

Twitter dilaporkan memiliki 330 juta pengguna aktif di dunia dan setiap hari memiliki lebih dari 500 juta cuitan. Twitter merupakan platform *microblogging* yang digunakan sebagai media informasi dimana pengguna dapat melakukan pembaruan dan berlangganan pengguna lain yang disebut *Following* untuk menerima pembaruan *Microblogging* dari pengguna lain (Ahmed dkk., 2017). *Tweets* atau cuitan menjadi fitur utama *platform* ini dapat berupa *microblogging* pesan teks, gambar dan video. Baru – baru ini *Twitter* menambahkan fitur baru yaitu *fleet* (berupa cuitan sementara yang bertahan dalam waktu 24 jam) dan *space* (berupa fitur audio siaran langsung percakapan pengguna). *Twitter* telah menjadi sumber data yang dapat dimanfaatkan seperti layanan darurat saat terjadi bencana. Data *Twitter* bersifat terbuka dibandingkan dengan *platform* media sosial lainnya seperti *Facebook*. *Tweets* juga dapat berisi meta-data yang berharga, termasuk data geospasial (Ahmed dkk., 2017).

Penting untuk memahami fitur *platform* media sosial sepenuhnya sebelum proyek penelitian dimulai, atau bahkan dipertimbangkan. Berikut fitur – fitur yang berhubungan dengan fitur utama *tweet* (Ahmed dkk., 2017).

1. *Tweet* adalah pesan singkat yang juga dikenal sebagai cuitan, kiriman, status, atau mikroblog dari pengguna di Twitter dan terdiri dari <280 karakter.
2. *Tweet* dapat berisi pembaruan tentang aktivitas pengguna, atau berbagi informasi berguna. *Tweet* dapat berisi tautan ke halaman *web, blog*, dan lain - lain. Untuk menghindari URL yang panjang, pengguna *Twitter* akan menggunakan versi singkat dari URL yang pendek diaktifkan oleh layanan eksternal seperti http://bit.ly/
3. *Hashtag*, dilambangkan dengan kata yang diawali dengan simbol '#', (misalnya, #PialaMenpora2021). *Hashtag* adalah konvensi platform untuk

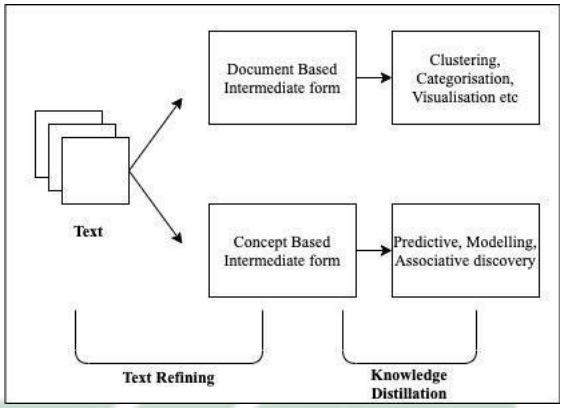
topik yang ditentukan pengguna, dan yang dimaksudkan untuk mengidentifikasi topik komunikasi.

4. Fitur balasan adalah platform yang disediakan untuk berkomunikasi dengan penulis *tweet* dengan mengklik tombol 'Balas' Twitter sebagai tanggapan atas *tweet*.
5. Fungsi *retweet* meneruskan *tweet* dari pengguna ke pengikut mereka dan ini mirip dengan meneruskan email ke kontak email seseorang, misalnya. Fitur 'sebutkan' mengenali pengguna dengan tanda '@' simbolis, tetapi ini tidak menggunakan fitur platform balasan, misalnya, 'Terima kasih @ pengguna - pegangan."
6. Twitter memungkinkan pengguna untuk me-*retweet* dengan komentar. Pengguna sekarang dapat mengutip *tweet* dan melampirkan komentar padanya, misalnya, pengguna *tweet* '[*Original tweet*]' sebagai *@userhandle* Saya setuju [*@ userhandle1* hari ini adalah Selamat siang]
7. Tren yang juga dikenal sebagai '*trending*' di Twitter mengacu pada saat suatu topik (kata kunci atau *hashtag*) populer pada waktu tertentu. Twitter menyediakan daftar topik yang sedang *trending* bagi pengguna, berdasarkan frekuensi *hashtag*.

2.4 *Text Mining*

Peningkatan tren saat ini terjadi dalam penggunaan komputer untuk menyimpan data. Data dapat berupa data terstruktur, semi struktur dan tidak terstruktur. Teks termasuk di dalam data tidak terstruktur. Teks menjadi alat paling umum untuk bertukar informasi seperti *update* status melalui media sosial. Akibatnya terdapat volume data yang besar berbentuk teks. Volume data teks yang besar tersebut memungkinkan untuk dilakukan *text mining*. *Text mining* dapat diartikan sebagai teknik identifikasi pola tersembunyi, berguna, dan menarik secara otomatis dari sekumpulan teks. Untuk mendapatkan informasi yang dibutuhkan, *Text mining* menjadi bidang ilmu pengetahuan yang menggabungkan *data mining, web mining, information retrieval, information extraction, computational linguistic*, dan *natural language processing* (L.Sumathy dan Chidambaram, 2013). *Text mining* dapat dianggap memiliki dua tahap secara umum dimulai dengan tahap

text refining dan dilanjutkan tahap *knowledge distillation*. Gambar 2.2 memaparkan *framework* pada *text mining*.



Gambar 2.2 *Framework Text Mining* (L.Sumathy dan Chidambaram, 2013)

Tahap *text refining*, merupakan tahap untuk transformasi teks menjadi bentuk peralihan atau bentuk yang terstruktur yang kemudian dilanjutkan tahap knowledge distillation dengan melakukan penyulingan pola dan pengetahuan yang relevan menggunakan metode dan proses yang sama pada data mining (L.Sumathy dan Chidambaram, 2013). Berdasarkan tujuan *text mining* untuk mendapatkan informasi yang berguna dan bentuk data yang tidak terstruktur sangat diperlukan Langkah awal untuk mempersiapkan agar teks dapat diubah menjadi lebih terstruktur. Langkah - langkah yang terlibat pada proses text mining sebagai berikut:

1. *Preprocessing*, Langkah *preprocessing* dibagi menjadi tokenisasi, penghapusan *stop word* dan *stemming*.
   a. tokenisasi, Dokumen teks berisi kumpulan pernyataan. Langkah ini membagi seluruh teks menjadi kata-kata dengan menghapus spasi kosong, koma, dll.
   b. penghapusan *stop word*, Langkah ini melibatkan penghapusan HTML, tag XML dari halaman web. Kemudian proses penghapusan kata-kata berhenti seperti 'a', 'adalah', 'dari' dll dilakukan.
   c. *Stemming*, Langkah ini mengacu pada proses mengidentifikasi akar kata tertentu. Pada dasarnya ada dua jenis stemming (i) infleksi dan (ii) derivasional. Algoritma yang paling umum digunakan adalah algoritma porter untuk stemming.

2. *Text Transformation*, Dokumen teks diwakili oleh kata-kata yang dikandungnya dan kemunculannya. Dua pendekatan yang digunakan untuk representasi dokumen adalah kantong kata-kata dan ruang *vector*
3. *Feature Selection*/*extraction*, Pada titik ini penambangan teks menjadi penambangan data. Metode peniruan data seperti pengelompokan, pengambilan informasi klasifikasi, dll., Dapat digunakan untuk penambangan teks

2.5 *Natural Language Processing*

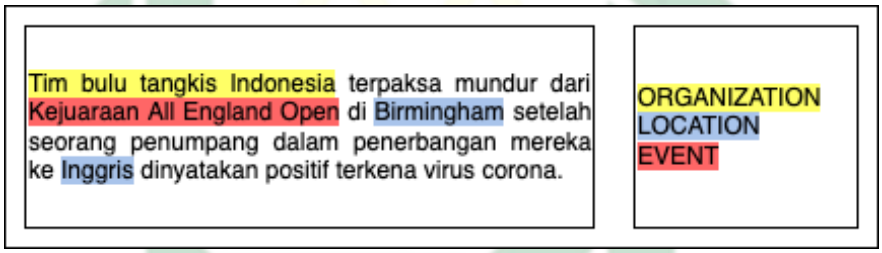
Natural Languange Processing (NLP) didefinisikan sebagai pemrosesan dan interaksi bahasa manusia secara otomatis dengan mesin. Istilah alternatifnya seperti *Language Technology*. NLP pada dasarnya multidisiplin dengan linguistik (walaupun sejauh mana NLP secara terbuka mengacu pada teori linguistik sangat bervariasi) (Ann Copestake, 2004). NLP juga memiliki tautan ke penelitian dalam ilmu kognitif, psikologi, filsafat dan matematika (terutama logika). Dalam *Computer Science*, NLP berkaitan dengan teori bahasa formal, teknik kompiler, pembuktian teorema, *machine learning*, dan interaksi manusia-komputer.

Algoritma Modern *Machine Learning* pada NLP, yaitu berdasarkan pendekatan *supervised learning* implementasi model NLP membutuhkan *feature* yang dihasilkan dari data inputan berlabel. Teknik dan tugas yang termasuk dalam NLP, yaitu (Grimley, 2016):

1. *Named entity recognition,*
2. *sentiment analysis,*
3. *speech segmentation,*
4. *text segmentation,*
5. *part of speech,*
6. *word sense ambiguation,*
7. *information extraction,*
8. *information retrieval,*
9. *machine transalation,*
10. *relationship extraction,* dan
11. *speech recognition*

2.6 *Named Entity Recognition*

Named Entities (NEs) adalah frasa yang dikategorikan dalam entitas dengan atribut yang serupa. NEs dapat membawa informasi kunci dari sebuah kalimat. NEs yang sering ditemukan adalah orang (PER), organisasi (ORG), lokasi (LOC) dan sebagainya (Béchet dan Mohit, 2011). Istilah NEs diperkenalkan pada konferensi MUC-6. Acara MUC-6 menjadikan sistem pengenalan entitas sebagai tolak ukur untuk tugas ekstraksi informasi dan diakui sebagai *Named Entity Recognition*. *Named Entity Recognition* (NER) adalah proses penting *Information Extraction* untuk menemukan dan mencari NEs pada bidang *Natural Language Processing* (Sun dkk., 2019). NER melibatkan pemrosesan dokumen struktur dan tidak terstruktur untuk mengidentifikasi entitas yang merujuk pada atribut orang, tempat, organisasi, peristiwa dan perusahaan. Gambar 2.3 menunjukkan contoh visualisasi kerja NER.



Gambar 2. 3 Contoh NER

Cara Kerja NER pada Gambar 2.3 dengan mengekstrak informasi nama dengan tepat secara otomatis dapat berguna banyak untuk masalah seperti *translator machine, information extraction,* dan *question answering*, Misal kunci dari proses *question answering* adalah untuk mengidentifikasi titik bertanya (siapa, apa, kapan, di mana, dll). Jadi dalam kasus *question answering*, titik yang ditanyakan harus sesuai dengan NEs. Contoh lainya dalam data teks biologi, sistem NER dapat secara otomatis mengekstrak nama yang telah ditentukan sebelumnya (seperti nama protein dan DNA) dari dokumen *raw*.

Secara umum NER menggunakan teknik pelabelan untuk mengklasifikasikan NEs sesuai domain yang disebut *Sequence Labelling*. Teknik *Sequence Labelling* memberi label pada setiap NEs pada frasa. Berdasarkan MUC-6 anotasi dan pelabelan dibedakan menjadi 3 yaitu anotasi ENAMEX(organisasi, orang, lokasi), anotasi TIMEX(tanggal, dan waktu) dan anotasi kuantitas NUMEX (persentase,

dan nilai moneter) (Béchet dan Mohit, 2011). Pengenalan entitas secara otomatis menjadi penelitian populer, pendekatan NER dibedakan menjadi tiga, yaitu:

1. *Rule Based* : Pendekatan *rule based* berfokus pada ekstraksi NEs dengan menggunakan sekumpulan aturan dan pola buatan manusia. Semua parameter sistem dibuat dan diatur dalam pemodelan. Secara umum sistem rule based terdiri dari sekumpulan pola yang menggunakan tata Bahasa, sintaksis, dan fitur ortografi (Sun dkk., 2019).
2. *Machine Learning*: Parameter pendekatan NER berbasis machine learning, entah bagaimana dapat diperkirakan oleh komputer. Pendekatan Machine Learning bertujuan merubah permasalahan identifikasi NEs menjadi permasalah klasifikasi dengan menggunakan pemodelan statistik klasifikasi pada proses menyelesaikannya (Sun dkk., 2019).
3. *Hybrid Model*: Pendekatan *Hybrid* berfokus menggabungkan metode rule based dan berbasis *machine learning*, dan membuat metode baru menggunakan poin terkuat dari masing-masing metode (Sun dkk., 2019).

2.7 Metode *Modelling*

Pendekatan *machine learning* dilakukan dalam proses pelatihan model, yaitu:

2.7.1 *Supervised Learning*

Supervised learning adalah pelatihan model dengan data yang telah ditentukan sebelumnya. Sehinnga model *supervised learning* memerlukan inputan data beruapa *feature* dan output label (Amershi, 2009). Dimana mesin belajar menggunakan algoritma tertentu dan melalui observasi data yang tersedia. Berdasarkan *feature*, *mechine learning* akan memetakan pattern atau pola sehingga menjadi *output* label pada data baru (Ann Copestake, 2004).

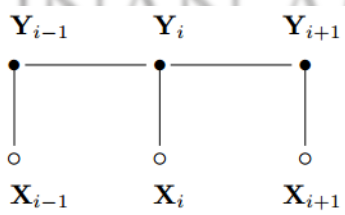
2.7.2 *Unsupervised Learning*

Berbeda dengan model *supervised*. Pelatihan Model Unsupervised Learning dapat mencari pola dari kumpulan data yang tersedia secara otomatis (Amershi, 2009). Dimana Unsupervised mengelompokkan data berdasarkan pattern pola yang belum ditemukan. Sehingga data inputan pada model pelatihan tidak membutuhkan

label, dengan algoritma yang digunakan mesin akan mengenali polanya sendiri (Siska dkk., 2018).

2.8 *Conditional Random Fields*

Conditional Random Fields (CRFs) merupakan *framework* probabilistik untuk pelabelan segmentasi pada *sequence* data, berdasarkan pendekatan bersyarat (Klinger, 2007). CRFs termasuk model pengklasifikasi Diskriminatif yang memodelkan batas keputusan antara kelas yang berbeda. Secara umum prinsip dasar CRFs adalah menerapkan *regression logistic* pada inputan *sequence* (Sutton dan McCallum, 2011). CRFs memiliki kelebihan dibandingkan model probabilitas klasik lainya. CRFs mengatasi adanya beban ketergantungan perkiraan asumsi yang tinggi dalam *Hidden Markov Model* (HMM). Terkait permasalahan tersebut CRFs bisa memilih sendiri berapa feature vector yg dibutuhkan untuk membentuk sebuah model CRFs (Jaariyah dan Rainarli, 2017). Selain itu CRFs bisa memiliki bobot yg bebas sedangkan HMM wajib memenuhi bobot tertentu. CRFs juga mengatasi adanya label bias dalam *Maximum-entropy Markov* Model (MEMM). Lantaran CRFs menghitung formula distribusi kondisional label untuk sequence data secara keseluruhan dibandingkan MEMM yang menghitung formula distribusi kondisional label untuk setiap elemen data (Jaariyah dan Rainarli, 2017; Y Munarko dkk., 2018). Tugas menetapkan urutan label ke sekumpulan urutan dan penerapan CRFs muncul di banyak bidang, termasuk bioinformatika, speech recognition, computer vision dan natural language processing (Wallach, 2004).



Gambar 2.4 *Linier Chain* CRF (Sutton dan McCallum, 2011)

Gambar 2.4 menunjukkan X = (X1, X2, X3 …Xn) merupakan himpunan data observasi dan Y= (Y1,Y2,Y3,….Yn) menjadi himpunan label yang mungkin diprediksikan. Sehingga dituliskan dalam persamaan (1) dan (2) (Klinger, 2007; Sutton dan McCallum, 2011; Wallach, 2004).

$$p(y|x) = \frac{1}{Z(x)} \prod_C \psi_C(y_c, x) \quad (1)$$

$$Z(x) = \sum_c \prod_c \psi_C(y_c, x) \quad (2)$$

Sehingga $\psi_C(y_c, x)$ menjadi fungsi potensial dan Z(x) menjadi fungsi normalisasi dari distribusi probabilitas kondisional label untuk semua x pada sequence data (Klinger, 2007). Fungsi potensial menetapkan jumlah yang mengikat label dengan fitur pada waktu bersamaan. Contoh sederhana fungsi potensial dapat berupa eksponensial dari jumlah bobot semua fungsi fitur.

Namun algoritma CRFs ditingkatkan dengan menerapkan *feature vector*. Penerapan *feature vector* bertujuan mengekspresikan jenis karakteristik urutan yang diwakili suatu data. Secara umum proses pembuatan model CRFs dilakukan dengan tahap *extraction feature function* (proses mengubah *feature vector* menjadi nilai fitur yang dapat dihitung) yang digunakan terlebih dahulu. Pemilihan *feature vector* dapat mempengaruhi hasil dan meningkatkan akurasi. Kemudian tahap penaksiran parameter untuk mendapatkan nilai optimal parameter *feature function*. Nilai optimal dapat dihitung dengan prosedur maksimum *likelihood*, sehingga dapat menunjukan jumlah parameter yang ada pada data *training*. Selanjutnya tahap inferensi yaitu menerapkan model terhadap data *testing*.

2.9 *Indonesian POS Tagging*

Pemberian label pos ke kata menjadi salah satu proses fundamental yang mendukung implementasi NLP. Pelabelan POS dilakukan dengan format *grammatical category*. Pos dapat mengenali suatu kata benda, kata kerja, kata sifat yang dapat memberikan informasi *linguistic* yang penting. Desain Pos tag Indonesia dibangun dengan 10.000 kalimat terdiri 262.330 *lexical* token dan terdiri dari 23 tags (Dinakaramani dkk., 2014). Tabel 2.3 *POS Tag* yang didesain oleh (Dinakaramani dkk., 2014).

Tabel 2.3 Indonesian POS Tagging (Dinakaramani dkk., 2014)

No	Pos	Contoh
1.	CC (Coordinating Conjunction)	Dan, tetapi, atau
2.	CD (Cardinal Number)	dua, juta, enam, 7916, sepertiga, 0,025, 0,525, banyak, kedua, ribuan, 2007, 25
3.	OD (Ordinal Number)	ketiga, ke-4, pertama

Tabel 2. 4 Indonesian POS Tagging Lanjutan (Dinakaramani dkk., 2014)

No	Pos	Contoh
4.	*DT (Determiner)*	Si, para, sang
5.	*FW (Foreign Word)*	Term, condition, random
6.	*IN (Preposition)*	Untuk, dalam, kepada, oleh, pada, di, dengan
7.	*JJ (Adjective)*	Bersih, luas, pendek, biru.
8.	*MD (Modal and Auxiliary Verb)*	Tentu, pasti, harus, boleh
9.	*NEG (Negation)*	Tidak, jangan, belum
10.	*NN (Noun)*	Sepeda, kucing, gelas, kipas, balon
11.	*NNP (Proper Noun)*	Kota, Surabaya, Indonesia, Idul adha, Bank BRI
12.	*NND (Classifier)*	Orang, lembar, jurusan
13.	*PR (Demonstrative Pronouns)*	Disana, ini, itu, situ
14.	*PRP (Personal Pronoun)*	Saya, kita, kami, kalian, meraka, dia
15.	*RB (Adverb)*	Segera, mungkin, justru
16.	*RP (Particle)*	Pun, kah, lah,
17.	*SC (Subordinating Conjunction)*	sejak, jika, seandainya, supaya, meski, seolah-olah, sebab,
18.	*SYM (Symbol)*	$, &, @, IDR
19.	*UH (Interjection)*	Oh, ooh, aduh
20.	*VB (Verb)*	Membuat, melakukan, bekerja
21	*WH (Question)*	W5+1H
22.	*X (Unknown)*	kalimat
23.	*Z (Punctuation)*	"…",?!

2.10 *BIO Labelling*

Standar *encoding* BIO membagi label sebagai B-*tag* dan I-*tag*. Jika kata adalah bagian dari entitas maka ditetapkan sebagai label kata awalan (B-*tag*) dan memberikan label pada kata setelahnya sebagai entitas lanjutan (I-*tag*). Sedangkan untuk label *outside/other* (O) ditetapkan sebagai kata yang berada diluar satu entitas (Kapetanios dkk., 2013; Ye dkk., 2020). BIO format dapat diterapkan pada pelabelan baik POS Tag maupun *Named Entities*. Contoh NER BIO format pada Tabel X

Tabel 2.5 Contoh *Labelling BIO* (Azarine dkk., 2019)

Token	**NER BIO**
ridwan	B-PER
kamil	O-PER
berkunjung	O
ke	O
telkom	B-ORG
university	I-ORG
yang	O
terletak	O
di	O
bandung	B-LOC

2.11 *Model Evaluation*

2.11.1 K-Fold

Pada tahap pembuatan model diperlukan skenario partisi data, yaitu membagi dataset untuk tahap training dan testing. Salah satu populer dalam partisi data adalah k-fold. K-fold dalah metode cross-validation yang melipat data set sebanyak k partisi. misalnya dalam penelitian terdapat dataset sebanyak 180. Ibarat memakai k=5, artinya 180 data terbagi menjadi 5 partisi dan isinya masing-masing 36 data. Namun perlu menetukan mana data untuk training dan testing. Karena perbandingannya 80:20. Maka dataset training sebanyak data dari 36x4 partisi dan dataset testing sebanyak 36. Kemudian dilakukan proses pelatihan model menggunakan data yang telah dipartisipasi sebanyak 5 kali (K=5). Namun posisi partisi dataset testing berbeda setiap iterasinya. Gambar 2.5 memaparkan gambaran iterasi setiap partisi.

	Total Keseluruhan Dataset				
Iterasi:1	Testing				
Iterasi:2		Testing			
Iterasi:3			Testing		
Iterasi:4				Testing	
Iterasi:5					Testing

Gambar 2.5 Konsep Kfold

Berdasarkan 2.5 iterasi pertama posisi dataset terletak pada partisi pertama, sedangkan iterasi kedua dataset testing terletak pada partisi kedua, menggunakan partisi ketiga untuk terasi ketiga dan seterusnya.

Setelah memabagi dataset. *Kfold cross validation* menghitung rata-rata dari nilai yang dihitung dalam iterasi. Pendekatan ini bisa mahal secara komputasi, tetapi tidak membuang terlalu banyak data (seperti halnya ketika memperbaiki set validasi arbitrer), yang merupakan keuntungan utama dalam masalah seperti inferensi terbalik di mana jumlah sampel sangat kecil.

2.11.2 *Confusion Matrix*

Confusion Matrix adalah method klasifikasi *binary* oleh dua tabel yang dibentuk dengan menghitung jumlah empat hasil dari pengklasifikasi *binary* yaitu disebut positif dan negative. Matriks terdiri dari empat sel, yang dapat diberi label *True Positive*, *True Negatif, False Positive* dan *False Negatif* seperti pada Tabel 2.5

Tabel 2.6 Label *Confusion Matrix* (Charoenpong dkk., 2019)

Confusion Matrix		***Predictive***	
		Positive	***Negative***
Actual	***Positive***	*True Positive*	*False Negatif*
	Negative	*False Positive*	*True Negatif*

Dimana label tersebut digunakan untuk membedakan hasil prediksi model yaitu:

1. *true positive* adalah jumlah positif kasus yang diklasifikasikan sebagai positif.
2. *True negative* adalah jumlah contoh negatif yang diklasifikasikan sebagai negatif.
3. *False positive* adalah jumlah negatif kasus yang diklasifikasikan sebagai positif.
4. *False negative* adalah jumlah kasus positif yang diklasifikasikan sebagai positif.

Pada penelitian ini entitas *location* dan *event* yang diidentifikasi secara benar dihitung sebagai *true positive*, sedangkan yang salah dihitung sebagai *false positive*. Sedangkan untuk entitas *other* yang diidentifikasikan secara benar dihitung sebagai *true negative* sedangkan yang salah dihitung sebagai *false negative.*

2.11.3 *Measure Performance*

Setelah model NER dengan CRFs dibuat berdasarkan partisi Kfold, dilakukan metode pengujian *precision, recall*, dan *f-measure*. Pengujian ini secara umum dapat digunakan untuk mengukur performa model yang dikembangkan. berdasarkan *confusion matrix*. Sebelum melakukan perhitungan Persamaan *precision, recall*, dan *f-measure* diberikan pada persamaan (3), (4) dan (5).

1. *Precision,* merupakan *positive predictive value*. Dimana persamaan 3 dilakukan dengan menghitung perbandingan jumlah prediksi benar (*true positive*) dengan jumlah keseluruhan prediksi.

$$precision = \frac{true\ positif}{true\ positif + false\ positif} \quad (3)$$

2. *Recall,* merupakan sensivity proporsi dari kasus positif. Dimana persamaan 4 menghitung perbandinan jumlah prediksi benar (*true positive*) dengan jumlah keseluruhan yang seharusnya benar.

$$recall = \frac{true\ positif}{true\ positif + false\ negatif} \quad (4)$$

3. *F-measure,* atau F1 merupakan nilai gabungan *recall* dan *precision*. Sehingga persamaan 5 memberikan nilai *harmonic* antara *recall* dan *precision*. Nilai *f-measure* terbaik adalah 1,0 dan yang terburuk adalah 0,0. Secara umum, nilai *f-meaure* lebih rendah dari ukuran akurasi karena mereka menerapkan *precision* dan *recall* ke dalam perhitungan. Sebagai aturan praktis, rata-rata tertimbang *f-measure* harus digunakan untuk membandingkan model pengklasifikasi, bukan akurasi global.

$$f - measure = \frac{true\ positif + true\ negatif}{True\ Positif + False\ Positif + True\ Negatif + False\ Negatif} \quad (5)$$

Dengan mengetahui nilai *recall, precision* hingga *f-measure* maka dapat dinilai apakah model berjalan dengan baik atau kurang baik.

2.12 *Python Language Programming*

Bahasa pemrograman python memiliki daya tarik yang kuat. Sejak kemunculan pada tahun 1991 python telah menjadi bahasa pemrograman populer. Python termasuk Bahasa pemrograman yang bersifat *open source, high-level* dan multifungsi (Van Rossum dan Muller, 2009). Python memungkinkan pemrograman dalam paradigma berorientasi objek, procedural dan fungsional. Selain itu, secara fleksibel python dapat digunakan dalam *web development, machine learning* dan lainnya. Kelebihan lainya python mempunyai koleksi *library* yang banyak dan besar sehingga memungkinkan untuk melakukan berbagai hal dalam penelitian. Dalam beberapa tahun terakhir, dukungan python yang ditingkatkan untuk pustaka

(seperti pandas dan *scikit-learn*) telah menjadikannya pilihan populer untuk tugas analisis data. Tabel 2.4 *library* yang digunakan dalam penelitian.

Tabel 2.7 *Library Python*

No.	***Library Package***	**Keterangan**
1.	Twint	Tool Web Scraping untuk mengumpulkan data twitter tanpa menggunakan API Twitter.
2.	Natural Language Toolkit Python	Tool untuk melakukan pemrosesan bahasa.
3.	Pandas	*Package* yang menyediakan struktur data cepat, fleksibel dan dirancang untuk membuat data "relasional" dan "berlabel" menjadi lebih mudah dipahami.
4.	*Sastrawi*	*Library* yang memungkinkan untuk mengurangi infleksi dalam teks bahasa Indonesia ke bentuk dasar ("Sastrawi · GitHub," n.d.)
5.	*all_indo_man_tag_corpus_model.crf.tagger*	Model *pretrained* bahasa Indonesia untuk melakukan pos tagging dengan CRFTagger (Yudi Wibisono, 2018)
6.	*Scikit-Learn*	*Tools Machine learning* pada analisis data
7.	*CRF Suite* dengan *wrapper Skicit-Learn*	*Library* yang implementasi *Conditional Random Fields* (Okazaki, 2007)

Dari berbagai implementasi lainnya CRFsuite pada Tabel 2.7 mempunyai keunggulan yaitu:

1. Proses *training* model yang cepat
2. Format sederhana pada training dan testing
3. Pengguanaan konsep linier chain (Okazaki, 2007).
4. File format yang efisien untuk melakukan penyimpanan dan akses model
5. Evaluasi performa dapat dilakukan bersamaan dengan proses *training* model (Okazaki, 2007).

Sebelumnya CRF *suite* dikembangkan pada bahasa pemograman C++ (Okazaki, 2007). Dengan adanya teknologi SWIG API, *CRF suite* dapat diimplementasikan dengan bahasa pemograman lain seperti python. Oleh karena itu pada penelitian ini menggunakan *library CRF suite* dengan *wrapper skicit-learn* yang disebut *Sklearn-crfsuite*.

2.13 Integrasi Keilmuan dan Keislaman

Penelitian ini dijalankan untuk deteksi peristiwa banjir dengan batasan dan metode yang telah dijelaskan pada latar belakang. Banjir berdampak buruk dan kerugian bagi masyarakat yang tertimpa. Jika ditelaah banyak pihak yang harus introspeksi terjadinya banjir karena adanya kerusakan sistem alam yang tidak dapat dihindari sebab akibat ulah manusia. Masih banyak masyarakat membuang sampah sembarangan, beralihnya lahan tanah resapan air menjadi limbah beton dan perbaikan sungai yang tidak maksimal.

Sebagaimana pandangan Islam terhadap banjir adanya faktor tersebut yang bisa menyebabkan banjir. Berdasarkan kisah Nabi Nuh AS. Terdapat kisah yang terkenal yaitu kapal yang mengarungi banjir. Kisah tersebut dimulai akibat kaum Nabi Nuh yang mempertanyakan dan menantang Nabi Nuh untuk mendatangkan azab. Diterangkan pada surah hud ayat 32 yaitu:

قَالُوْا يٰنُوْحُ قَدْ جَادَلْتَنَا فَاَ كْثَرْتَ جِدَالَنَا فَأْتِنَا بِمَا ٣٢
تَعِدُنَآ اِنْ كُنْتَ مِنَ الصّٰدِقِيْنَ

Terjemahan:

"Mereka berkata, "Wahai Nuh! Sungguh, engkau telah berbantah dengan kami, dan engkau telah memperpanjang bantahan terhadap kami, maka datangkanlah kepada kami azab yang engkau ancamkan, jika kamu termasuk orang yang benar.".

Kemudian surah asyuraah ayat 119 – 120:

فَاَنْجَيْنٰهُ وَمَنْ مَّعَهٗ فِى الْفُلْكِ الْمَشْحُوْنِ ١١٩
ثُمَّ اَغْرَقْنَا بَعْدُ الْبَاقِيْنَ ١٢٠

Terjemahan:

"Kemudian Kami menyelamatkannya Nuh dan orang-orang yang bersamanya di dalam kapal yang penuh muatan. Kemudian setelah itu Kami tenggelamkan orang-orang yang tinggal.".

Merujuk pada dua surah di atas, dilakukan wawancara untuk memperoleh Integarasi keislaman terkait penelitian ini. Wawancara yang telah dilakukan dengan ahlinya, bapak Syuhada, M.Ei. Selaku dosen PAI di Salah satu kampus swasta di

Lamongan dan Gresik. Beliau menjelaskan surah hud ayat 32 dan asyuraah 119 - 120 tersebut menjalesakan nabi nuh menyiapkan kapar besar untuk kaumnya, dalam rangka mengantisipasi datangnya banjir atas perintah allah.

Bencana adakalanya berbentuk ujian, azab dari ulah manusia dan apapun itu pasti kehendak Allah SWT. Untuk menghadapi bencana perlu dilakukan antisipasi. sehingga antisipasi dianjurkan dalam islam. Artinya antisipasi sebagai bentuk persiapan dan waspada pada bencana. Tetapi bukan berarti bencana tidak akan terjadi.

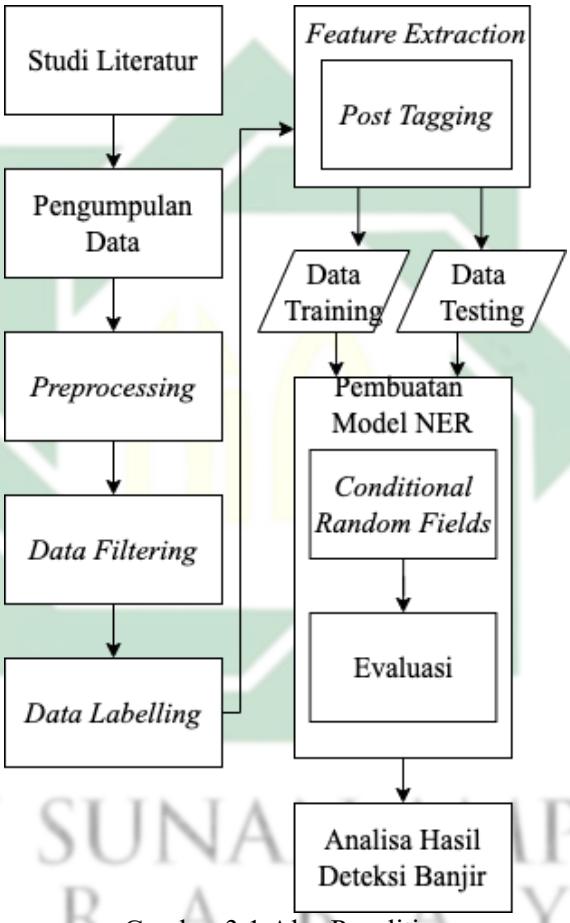
Dilanjutkan dengan wawancara bersama bapak Drs. Achmad Chusaini, selaku pengelolah Yayasan TPQ Darul Ulum Sukolilo Sukodadi Lamongan. Beliau menjelaskan ada hadis riwayat turmudzi dengan arti ikatlah untamu dahulu baru berpasrahlah artinya berupayalah dahulu sebelum bertawakal. Integrasi Keislaman dengan penelitian ini, deteksi peristiwa banjir dapat diartikan sebagai bentuk upaya, usaha sebelum berserah dan bertawakal terhadap dampak dari banjir.

Dimana deteksi peristiwa banjir menjadi salah satu dari serangkaian kegiatan untuk menghimbau dampak terjadinya peristiwa banjir. Berlandasan al quran, hadis dan hasil wawancara para ahli, penulis berupaya mengembangkan *NAMED ENTITY RECOGNITION* MENGGUNAKAN METODE *CONDITIONAL RANDOM FIELDS* UNTUK DETEKSI PERISTIWA BANJIR DI GERBANG KERTOSUSILA BERDASARKAN DATA *TWITTER*. Karena Upaya deteksi banjir dapat menjadi Tindakan awal pada proses penyampaian informasi terhadap himbauan peristiwa bencana kepada masyarakat. Selain itu, dapat membantu masyarakat untuk melakukan penanganan dini pada peristiwa banjir sehingga mampu mengurangi dampak dari peristiwa banjir tersebut.

BAB III
METODE PENELITIAN

Metode penelitian *Named Entity Recognition* (NER) menggunakan *Conditional Random Fields* (CRFs) untuk deteksi lokasi peristiwa banjir ditampilkan pada Gambar 3.1.

3.1 Desain Penelitian



Gambar 3.1 Alur Penelitian

Gambar 3.1 bentuk alur dari tahapan perancangan penelitian. Alur tersebut mempermudah penyampaian informasi dari tahapan penelitian yang dilakukan. *Named Entity Recognition* untuk deteksi banjir berdasarkan data *Twitter* menggunakan *Conditional Random Fields* melalui tahapan studi literatur, pengumpulan data, data *preprocessing*, data *filtering*, *data labelling*, *feature extraction*, membagi data menjadi 2 (*data training* dan *testing*) dan membuat model NER (yang meliputi proses implementasi metode *Conditional Random Fields* dan evaluasi model) kemudian melakukan analisa hasil deteksi banjir.

3.2 Uraian Desain Penelitian

3.2.1 Studi Literatur

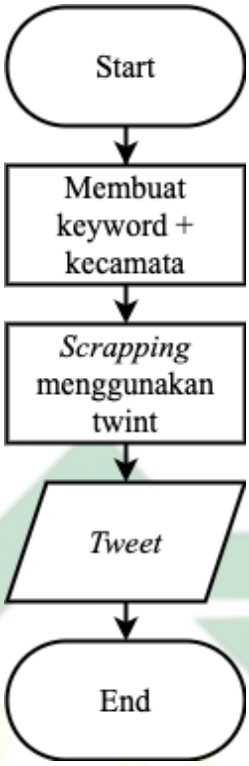
Tahap studi literatur dilakukan dengan melakukan kajian literatur yang berkaitan dengan topik yang berkaitan dengan NER terlebih pada kasus kejadian banjir dan media sosial *Twitter*. Referensi yang digunakan pada penelitian ini bersumber dari jurnal dan buku yang berkaitan tentang *Text Mining*, NER, CRFs. Dari referensi tersebut dapat diketahui informasi sebagai berikut:

a. Penggunaan sumber data *Twitter* pada aplikasi deteksi kejadian banjir menjadi lebih ekonomis dibandingkan dengan cara konvensional
b. Performa penggunaan NER berbahasa Indonesia khususnya berbasis *Twitter* masih berlanjut menjadi tantangan dalam perkembangan penelitian *Natural Language Processing* dan masih bergantung pada algoritma dan data *training* yang digunakan.
c. Penggunaan Algoritma CRFs membutuhkan *feature vector* untuk proses *feature extraction*.

Berdasarkan informasi tersebut penelitian ini meninjau hasil penelitian oleh (Jaariyah dan Rainarli, 2017; Muhammad dan Khodra, 2015) untuk menggunakan post tagging sebagai *feature vector* dalam proses *feature extraction* pada model NER menggunakan CRFs.

3.2.2 Pengumpulan Data

Sesuai dengan tujuan, penelitian ini menggunakan data yang bersumber dari *Twitter*. Data *Twitter* yang dikumpulkan berupa *tweets* terkait informasi kejadian banjir di Gerbang Kertosusila. Penarikan data pada twitter dilakukan dengan teknik *web scraping* dengan *python programming language*. Penelitian ini menggunakan *tools* yang tersedia yaitu twint. *Twint* dapat melakukan ekstrak semua *tweet* yang memiliki kata kunci tertentu dan pada periode waktu tertentu. Selain itu, *Twint* dapat melakukan ekstrak *tweet* dari akun tertentu dengan memasukkan nama akun sebagai parameter. Kemudian, data tweet yang telah diperoleh diubah menjadi file dalam format csv. Gambar 3.2 menampilkan alur pengumpulan data.



Gambar 3.2 *Flowchart* Pengumpulan Data

Data yang dikumpulkan dari tahun 2016 hingga Mei 2022. Contoh *tweets* yang telah dikumpulkan dipaparkan pada Tabel 3.1. Untuk mendapatkan data seperti Tabel 3.1. Penelitian ini menggunakan *keyword* "banjir", dan dikombinasikan dengan *keyword* kota, kabupaten dan kecamatan di Kawasan Gerbang Kertosusila. Seperti "banjir surabaya", "banjir wonocolo", "banjir lamongan", "banjir gresik", "banjir bangkalan" dan seterusnya.

Tabel 3.1 Contoh Dataset

Tweets
07.48: Banjir di Raya Juanda. Sebaiknya HINDARI jalur ini
#SSinfo Banjir di Kecamatan Kalitengah Lamongan ini melanda delapan desa dengan ketinggian air di jalan raya paling tinggi 50 cm
09.39: Kondisi banjir di Jalan Raya Morowudi Cerme Gresik, Senin (15/3/2021) pagi tadi. Havid pendengar SS via WhatsApp SS melaporkan, ketinggian air sekitar 50-80 cm. Beberapa sepeda motor yg nekat melintas mogok. Untuk mobil, diarahkan lewat Metatu Benjeng-Balongpanggang. (hm) https://t.co/AvLLwc5jXM"

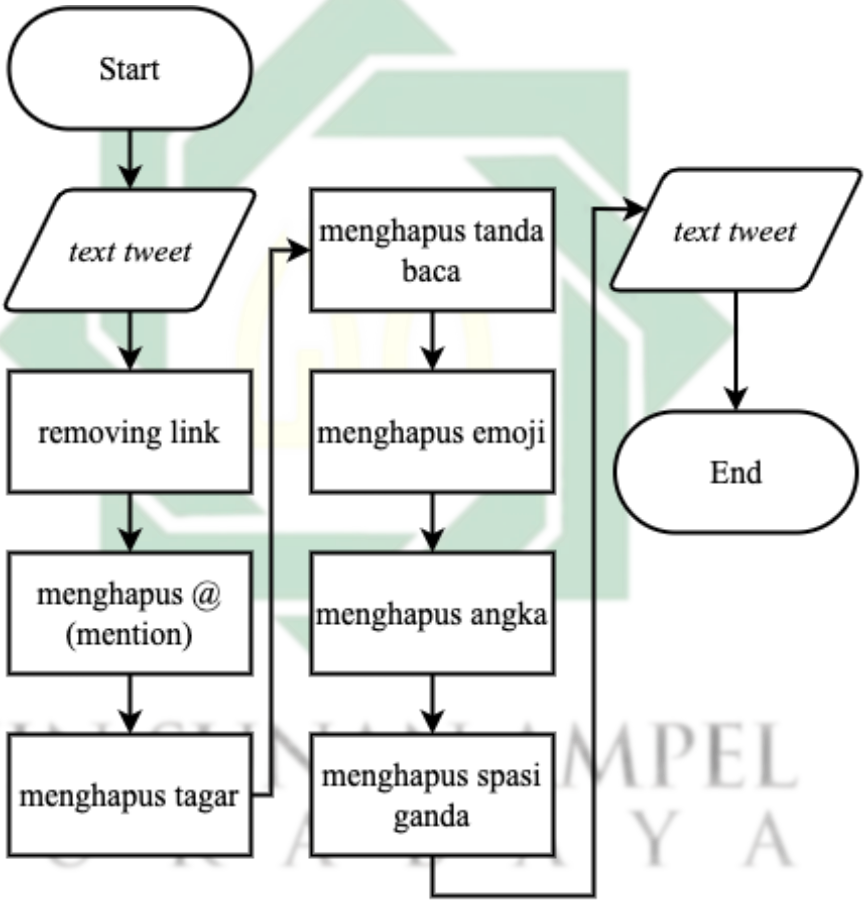
3.2.3 *Preprocessing*

Tahap *Preprocessing* pada penelitian ini dilakukan untuk merubah data text yang didapatkan dari proses sebelumnya menjadi data struktur yang siap di *training* dan di *testing*. untuk pengolahan data dilakukan dengan bantuan *library* berbasis

python programming language, yaitu NLTK. *Preprocessing* yang dilakukan pada data sebagai berikut:

1. *Cleaning*

   Alur proses cleaning tergambarkan pada Gambar 3.3. Proses ini dilakukan untuk membersihkan kata dari karakter yang tidak mempengaruhi hasil pengenalan entitas. Karakter yang dihilangkan adalah angka, tanda baca, *username, mention, hashtag* (#), simbol, teks "RT" , teks "FAV" dan *url website*.

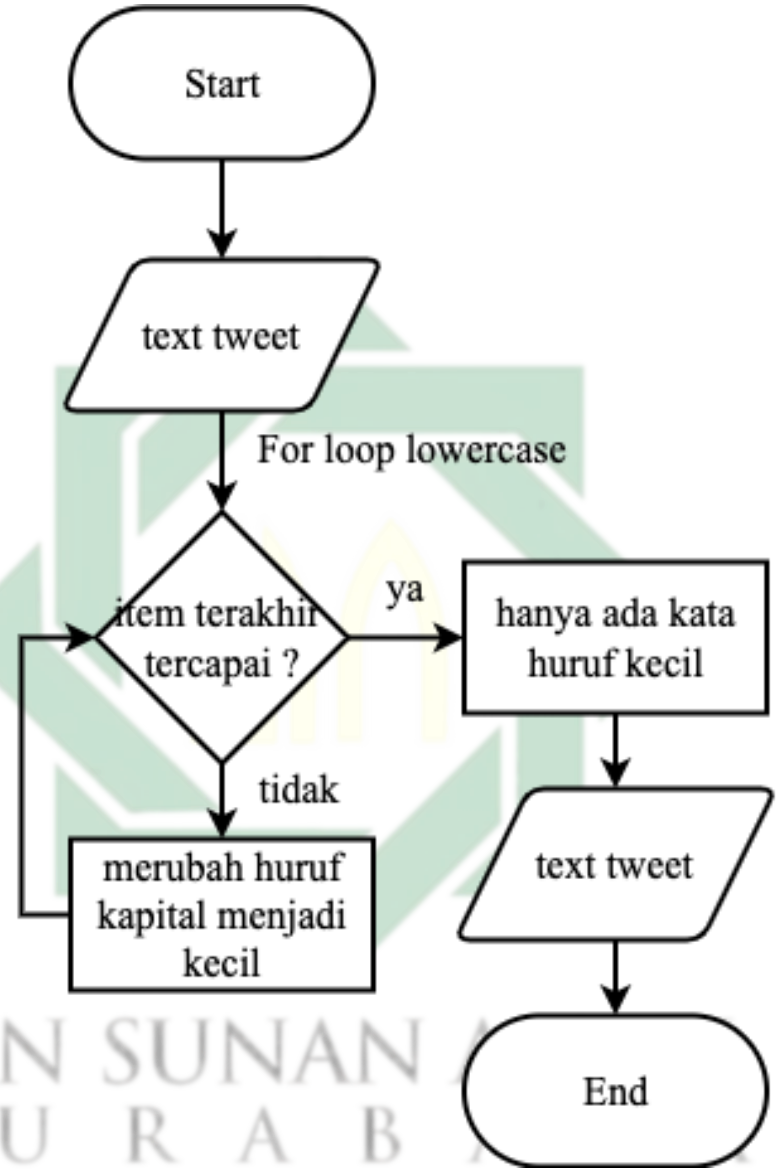


Gambar 3.3 *Flowchart Cleaning*

2. *Case folding*

   Pada proses *case folding* pada dasarnya adalah proses yang diterapkan pada *sequence character.* Dimana karakter yang diidentifikasi tidak berhuruf besar diganti dengan dengan padanan huruf besar karakter atau sebaliknya tidak berhuruf kecil diganti dengan padanan huruf kecil. Sehingga seluruh karakter

huruf memiliki karakteristik sama. Pada penelitian ini akan merubah semua karakter huruf besar menjadi kecil untuk mempermudah memproses kata-kata. Gambar 3.4 memaparkan gambar alur *Case Folding*.

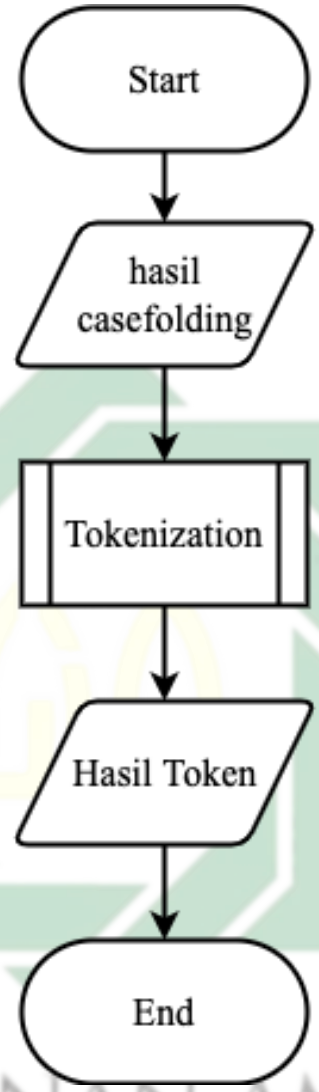


Gambar 3. 4 *Flowchart Case Folding*

3. *Tokenization*

   Tokenization adalah proses akan memisahkan teks yang berupa dokumen, paragraf, dan kalimat menjadi bagian tertentu (token). Batas pemisahan kata bergantung pada karakteristik data yang digunakan. Pada *word-base tokenizing* seluruh kata dapat dipisahkan berdasarkan tanda baca, spasi *delimiters* dan lain – lain. Penelitian ini memisahkan kaliamat pada tweets

menjadi token. Karakter spasi pada tweet banjir digunakan sebagai batas pemisah kata pada tahap ini. Gambar 3.5 memaparkan alur *tokenizing*

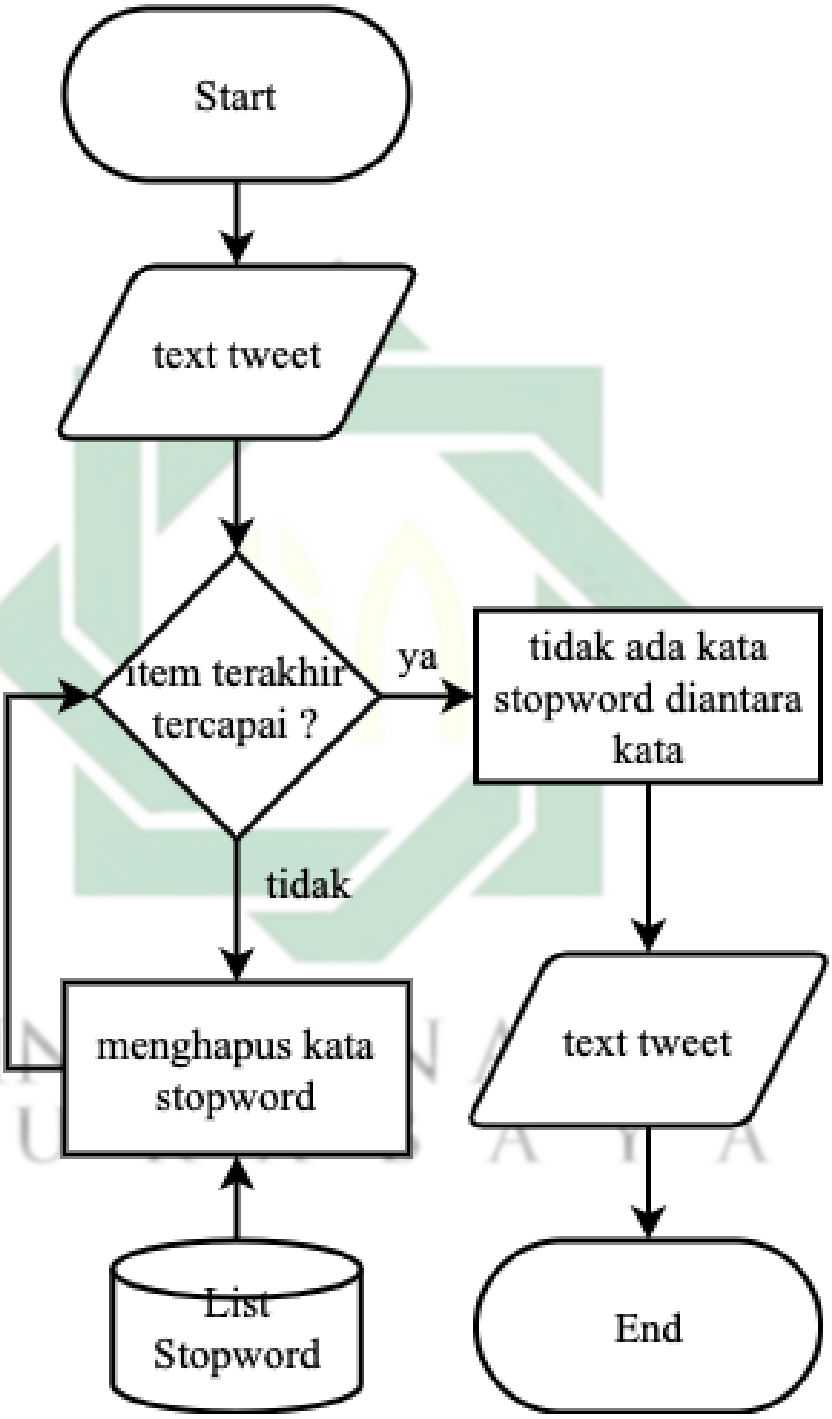


Gambar 3.5 *Flowchart Tokenizing*

4. *Stop Word*

   Stop word, sebenarnya adalah kata – kata yang umum digunakan dan sengaja dihindari untuk menghemat ruang dan waktu pemrosesan data. Selain itu, kata yang dianggap tidak mengandung banyak informasi didalam text. Penelitian ini menggunakan *stop word* bahasa Indonesia yang dikumpulkan oleh Owen. Misalnya "adalah", "gimana", "yakni", "dengan", "yang", "dan", "di", dan seterusnya (Owen, 2022). Gambar 3.6 dilakukan proses *stop word* untuk menghilangkan kata-kata yang memiliki informasi rendah dari sebuah teks.

Dimana target *text tweet* berupa token, kemudian membandingkan target dengan daftar stop word, jika sama maka stop word dihilangkan dan proses tersebut berulang hingga item target tercapai.

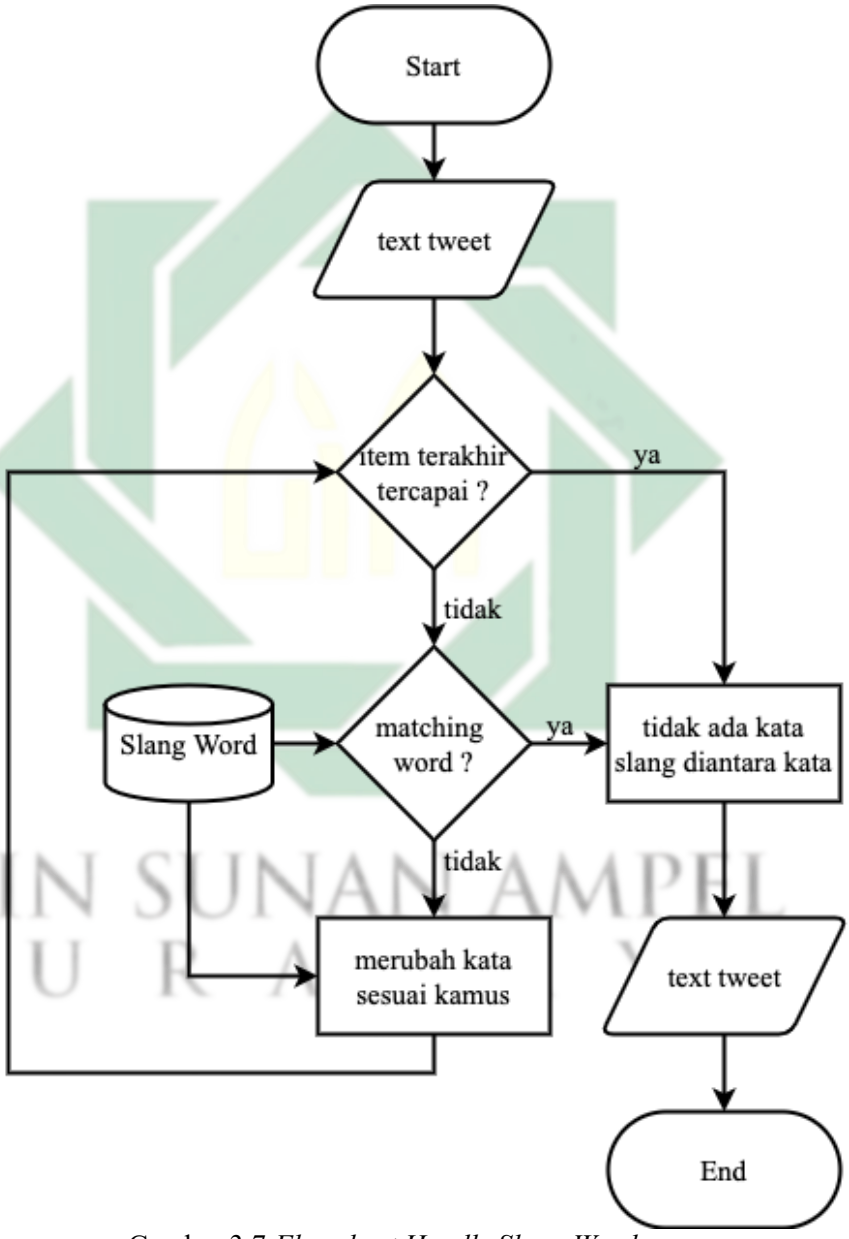


Gambar 3.6 *Flowchart Stop Word*

5. *Handle Slang Word*

Telah dilakukan proses *cleanning* hingga *stopword* pada data t*weets*, namun masih banyak kata yang menggunakan bahasa tidak formal, baik *slang*

maupun bentuk pendek dari kata lain. Salah satu cara menguraikan kata pendek, dan *slang* dengan menggunakan sumber daya ekternal. Itu sebabnya banyak kamus bahasa gaul atau slang. Pada penelitian ini menggunakan *corpus Colloquial Indonesian Lexicon* (Salsabila dkk., 2018) dan *IndoCollex* (Wijaya, 2021). Gambar 3.7 melakukan proses dengan merubah kata – kata gaul dan slang ke bahasa yang formal.

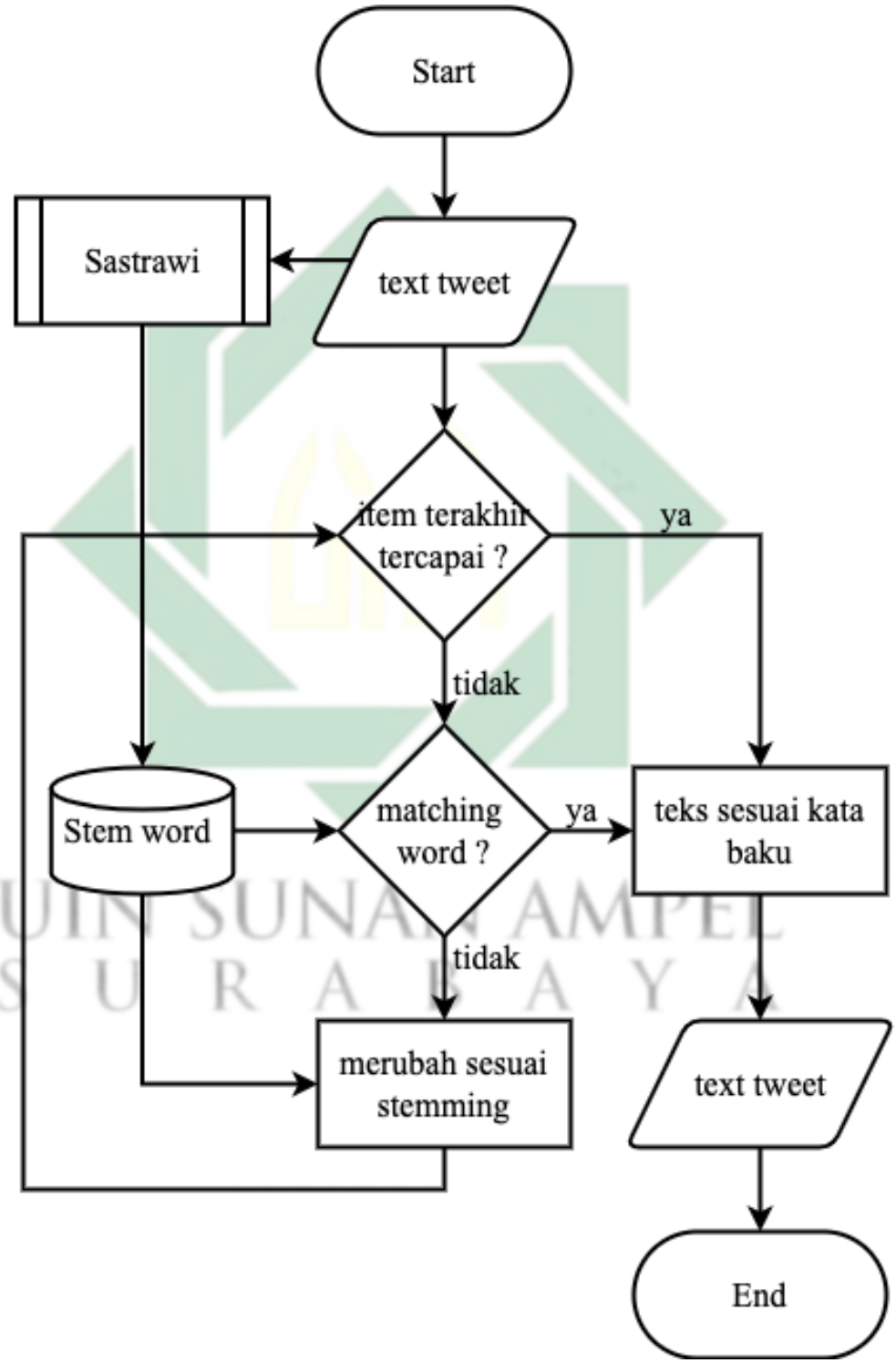


Gambar 3.7 *Flowchart Handle Slang Word*

6. *Stemming*

Setelah melakukan proses merubah kata menjadi formal, dilakukan proses *stemming* untuk mencari kata dasar dan kata akar. Berbeda dengan bahasa

inggris, proses *stemming* pada teks berbahasa Indonesia dilakukan dengan membuang kata imbuhan *sufiks* dan *prefiks*. Gambar 3.8 melakukan proses *stemming* menggunakan *library stemming sastrawi python* yang telah menerapkan algoritma Nazief dan Adriani.

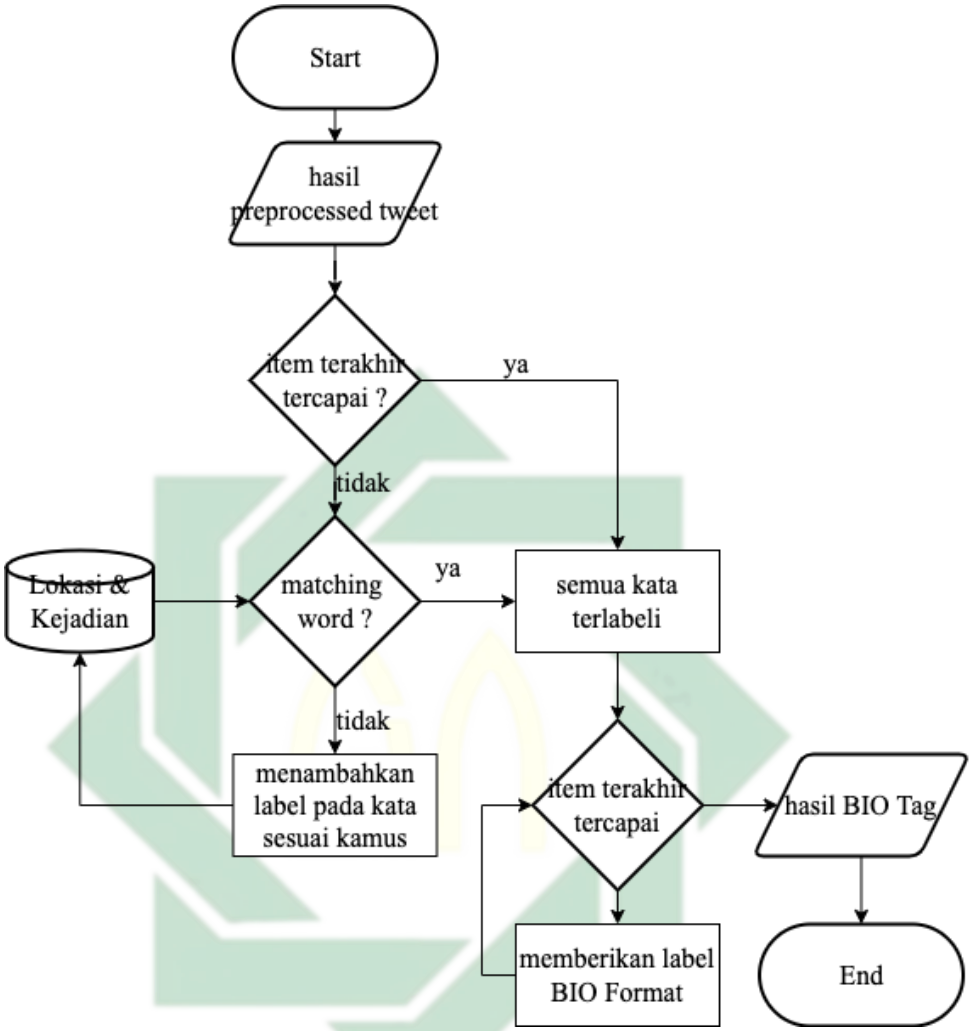


Gambar 3.8 *Flowchart Stemming*

3.2.4 *Filtering Duplicate*

Proses ini dilakukan dengan menghapus data *tweets* yang sama, karena Twitter memiliki fitur *retweet* yang menyebabkan jumlah *tweets* yang sama di-*retweet* oleh pengguna lain dan menyebabkan teks berulang dengan topik yang sama. Selain menghapus data *tweets* dilakukan juga proses penyaringan data. Penyaringan dilakukan setelah proses pengumpulan data, selama proses pelabelan data, dan setelah proses pelabelan data. Proses ini dilakukan secara manual dengan mengecek *tweets* satu per satu, dengan tujuan untuk memilih dan menghapus data *tweets* yang tidak sesuai dengan yang diinginkan, seperti *tweets* yang berisi iklan terkait penggunaan kata banjir misalnya "banjir hadiah" atau "banjir gol", *tweets* yang tidak mencantumkan lokasi terjadinya bencana banjir, *tweets* banjir yang tidak terjadi di Kawasan Gerbang Kertosusila dan tweet ganda yang salah satunya berisi informasi lokasi yang salah. Proses penyaringan setelah pengumpulan data dilakukan dengan mengecek data *tweet* yang ada. Kemudian penyaringan saat proses pelabelan data, Ketika ditemukan data *tweet* yang tidak cocok atau terkait bencana banjir maka *tweet* tersebut dihapus. Dan proses penyaringan setelah pelabelan data dilakukan dengan mengecek adanya duplikasi data, jika ada *tweet* yang sama atau salah satunya memiliki informasi lokasi yang salah. *Tweets* dengan informasi lokasi yang salah akan dihapus.

3.2.5 *Data Labelling*



Gambar 3.9 *Flowchart BIO Tag*

Gambar 3.10 Alur *tagging* pada *tweets* yang diberi label menggunakan notasi BIO (*Begin, Inside dan Other*) sebagai skenario pelabelan yang menunjukkan urutan yang kemudian diklasifikasikan menjadi tiga kelas kategori, yaitu *Begin-Tag* dan *Inside-Tag* dan *Other.* Tabel 3.2 menjelaskan kategori kelas notasi BIO yang digunakan dalam penelitian.

Tabel 3.2 Aturan BIO

Notasi	Keterangan
B-event	*Begin-Tag* entitas kejadian banjir dengan notasi awal entitas
I-event	*Inside-Tag* entitas lanjutan kejadian banjir
B-location	*Begin-Tag* entitas lokasi kejadian banjir yang berada di awal entitas.
I-location	*Inside-Tag e*ntitas lanjutan lokasi banjir
O	Entitas *Other, Outside* atau entitas selain entitas kejadian dan lokasi

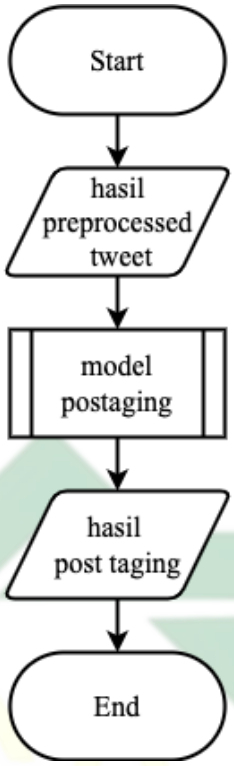
Gambar 3.10 Proses pelabelan tweets dilakukan dengan membangun *dictionary* lokasi dan *event*. *Dictionary* digunakan untuk proses *mapping* diantara token dengan kata lokasi dan *event*. Kemudian dilakukan proses penetapan begin dan inside dengan aturan Tabel 3.2 seperti contoh pada skenario tabel 3.3

Tabel 3.3 Contoh skenario BIO

Token	NER BIO Format
Lapor	O
Musim	B-event
Hujan	I-event
Terjadi	O
Banjir	B-event
Karanggeneng	B-location
Lamongan	I-location

3.2.6 *Feature Extraction*

Pada penelitian ini *feature extraction* digunakan untuk ekstraksi fitur yang digunakan dalam pemrosesan. *Feature extraction* juga melakukan tugas untuk merubah data semula berformat *text* menjadi *numeric* untuk proses . Selesai proses ini data akan diproses dalam kalkulasi menggunakan conditional random fields. Untuk melakukan proses ini, membutuhkan *feature vector* atau *function*. *Feature vector* yang digunakan penelitian ini adalah bagian kata, *pos tag, title, upper, lower,* dan *nearby word*. Pada penelitian ini menetapkan pos tag seperti pada alur gambar 3.10



Gambar 3.10 *Flowchart POS Tagging*

Gambar 3.10 Alur *POS tagging* dilakukan untuk melakukan pelabelan jenis kata. Penelitian ini menggunakan korpus Pos Tag Indonesia, hasil penelitian oleh (Dinakaramani dkk., 2014). Dengan *format tab separated value*. Korpus *POS Tagging* Indonesia terdiri dari sepuluh ribu kalimat berbahasa Indonesia yang dibangun dari 256.683 token leksikal dan didesain terdiri dari 23 kelas kata. Dengan POS *tagging* tersebut conditional random fields akan memodelkan perilaku data inputan berurutan dan mempertimbangkan konteks sebelumya saat membuat prediksi. menggunakan *feature extractor* yang mempunyai inputan berikut menjadi.

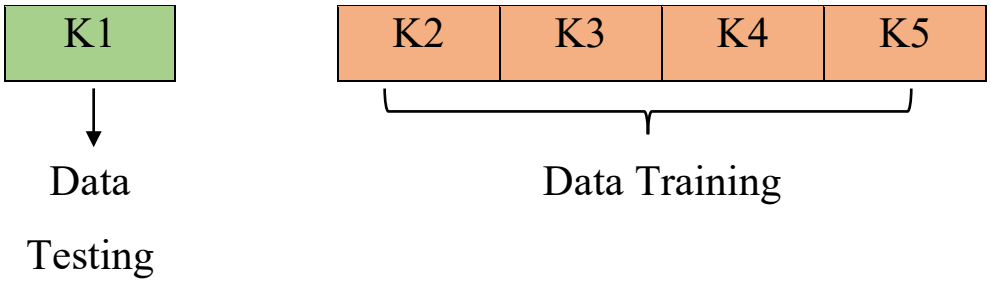
1. Set input *feature*, X
2. Posisi i pada data point yang sedang di prediksi
3. Label data point i-1 di X
4. Label data point i di X

3.2.7 Pembuatan Model NER

1. Pembagian Proporsi Dataset *Training* dan *Testing*

   Setelah menetapkan apa saja *feature* yang digunakan. Tahap ini dilakukan untuk membagi proporsi dataset training dan testing pembuatan model NER.

Konfigurasi pembagian dilakukan dengan menggunakan teknik Kfold *Cross Validation*. Dimana skenario digambarkan pada Gambar 3.11.



Gambar 3.11 Skenario Kfold K=5

Skenario Gambar 3.11 membagi data menjadi *training* dan *testing*. Kemudian dataset dilakukan proses silang dimana data testing menjadi data training dan sebaliknya. Dimana detail skenario Kfold *Cross validation* pada penelitian ini dilakukan dengan proses sebagai berikut:

a. Penelitian ini membagi jumlah data sebanyak 5 Fold. sehingga setiap fold terdiri dari 20% dari dataset.
b. Pada setiap fold, dimisalkan sebagai *K1 – K5*. Dan menetapkan 1 proporsi (K1) sebagai testing, sisanya sebagai data training (*K2 – K5*). Artinya terdapat proporsi 20% untuk data *testing* dan 80% untuk data *training*.
c. Melakukan iterasi proses a dan b hingga seluruh *fold* partisi telah digunakan sebagai testing.

Tahap ini menghasilkan 5 fold masing - masing partisi terdiri dari 392 data. Penelitian menggunakan rasio 80:20, maka tahap *training* dilakukan dengan data sebanyak 1568 dan tahap testing sebanyak 392 data. Sehingga iterasi dilakukan dengan kemungkinan data testing yang berbeda setiap fold yang digunakan.

2. *Conditional Random Fields*

Conditional Random Fields melihat *previous word*, *current word*, dan *next word* secara berurutan seperti yang digambarkan dalam formula (1) sehingga akan diketahui *feature vector*. Fitur tersebut akan dijadikan fitur ciri khusus untuk mendesain *feature extractor*. Inisialisasi penimbang awal akan digunakan untuk prediksi. Setiap *feature function* akan dikalikan dengan penimbang. *Log Likelihood* dihitung kemudian dilakukan iterasi sebanyak

100 kali. Feature Extractor digunakan sebagai penimbang sampai mendapatkan urutan dari hasil probabilitas tertinggi. Ketika ada kalimat baru, melakukan pengecekan *previous word, current word, next word*, dan akan menghitung nilai peluang maksimal. Nilai peluang tertinggi akan mengarah ke ke satu kategori yang paling sesuai. Sesuai dengan formula (2) *Conditional Random Field* akan mencoba menentukan bobot berbagai fungsi fitur yang akan memaksimalkan kemungkinan label dalam data training. Penelitian ini menggunakan library CRF digunakan yaitu *CRFSuite* yang tersedia dengan menggunakan *library Skicit Learn CRF*.

3. Evaluasi dan Pengujian

   Proses ini dilakukan dengan mengukur ketepatan kinerja *Named Entity Recognition* berjalan dengan baik atau kurang baik. Penelitian ini menggunakan pengukuran performa *f-measure* atau *F1*. Proses *f-measure* dihasilkan dari *recall* dan *precision result* pada persamaan (5). Untuk entitas lokasi dan *event* yang diidentifikasi secara benar dihitung sebagai *true* positif, sedangkan yang salah dihitung sebagai *false* positif. Sedangkan untuk entitas *other* yang diidentifikasikan secara benar dihitung sebagai *true* negatif, sedangkan yang salah dihitung sebagai *false* negatif. Kemudian dilakukan perhitungan *precision* dengan rumus (3) dan nilai recall dengan rumus (4). Dengan pengukuran *f-measure* penelitian ini akan mengetahui performa model berjalan dengan baik atau kurang baik.

3.2.8 Analisa Hasil Deteksi Banjir

Setelah model dibuat, model mendeteksi entitas lokasi kejadian banjir dan digambarkan distribusi lokasi kejadian banjir di Kawasan Gerbang Kertosusila. Kemudian dilakukan pemaparan hasil evaluasi dan pengujian model NER dari model yang telah dibuat dan penarikan kesimpulan hasil deteksi banjir.

3.3 Tempat Dan Waktu

Penelitian terkait akan dimulai akhir Februari hingga Juli 2022. Penelitian dilaksanakan di Lamongan dan Surabaya.

3.4 Jadwal Penelitian

Penelitian dilaksanakan dengan jadwal penelitian pada Tabel 3.4.

Tabel 3.4 Jadwal Penelitian

Jadwal	**Bulan**					
	Feb	**Mar**	**April**	**Mei**	**Juni**	**Juli**
Pengumpulan data						
Preprocessing Filtering Duplicate Data Labelling						
Feature Extraction						
Pembuatan NER						
Evaluasi dan Pengujian						
Analisis Hasil Deteksi						

BAB IV
HASIL DAN PEMBAHASAN

Pada bagian ini akan dipaparkan hasil dan pembahasan penelitian sebagaimana tahapan pada metodologi penelitian.

4.1 Hasil Pengumpulan Data

Data peristiwa banjir dikumpulkan melalui teknik *scraping twitter* menggunakan *library TWINT* dengan kata kunci kombinasi banjir dan kota/kecamatan. *Pseudocode* 4.1 pengumpulan data sebagai berikut.

Pseudocode 4.1 *Collecting Data*

```
Collecting Data
Algorithm:
FOR keyword in subdistrict do
    WRITE on list query_district with f'{keyword
subdistrict} banjir'
ENDFOR
SET twint.Config() as c function
FOR todo in query_district do
    CALL c.Search todo
    CALL c.Lang Indonesian
    CALL c.output as fulldata.csv
ENDFOR
```

Data yang didapatkan berjumlah 2408 cuitan banjir dengan 36 kolom yaitu *'id', 'conversation_id', 'created_at', 'date', 'time', 'timezone', 'user_id', 'username', 'name', 'place', 'tweet', 'language', 'mentions', 'urls', 'photos', 'replies_count', 'retweets_count', 'likes_count', 'hashtags', 'cashtags', 'link', 'retweet', 'quote_url', 'video', 'thumbnail', 'near', 'geo', 'source', 'user_rt_id', 'user_rt', 'retweet_id', 'reply_to', 'retweet_date', 'translate', 'trans_src',* dan *'trans_dest'*. Tabel 4.1 menunjukan bahasa tweet yang telah dikumpulkan.

Tabel 4.1 Bahasa

Bahasa	**Jumlah**
Indonesia (Id)	2400
Inggris (En)	2
Tagalog (Tl)	2
Danish (Da)	1
Spanish (Es)	1
Haitian (Ht)	1
Undertemined	1

Mempertimbangkan informasi Tabel 4.1, penelitian telah dilakukan hanya pada bahasa Indonesia. Dari 36 kolom penelitian sudah dilakukan dengan

menggunakan *dataset* kolom *tweet* dengan karakteristik berbahasa Indonesia, dan tipe data object. Tabel 4.2 menampilkan contoh sampel *tweet* yang telah didapatkan pada proses pengumpulan data.

Tabel 4.2 Sampel *Dataset Twitter*

Tweet
@e100ss melaporkan banjir di desa pucangro kec. Karanggeneng kab. Lamongan. Banjir kurang lbh 20-25 cm setengah ban mobil sepanjang 200m dan banyak jalan berlubang https://t.co/ckFbffEmim
Banjir Jl. Morowudi Kulon, Cerme, Gresik. @e100ss @infoGRESIK @gresikNEWS https://t.co/8AlHcXRvUR
Laporan #banjir Kali Blega di Kab Bangkalan, Senin 28/1, menggenangi sekitar 100 rumah. https://t.co/QFTQ29cwHv
Radio KARIMATA 14.11 Pak Fadli, Blega ,Bangkalan, mengabarkan, Jl. Raya Blega, Bangkalan, tepatnya di Dusun Laok Songai ketinggian air sekitar 25 cm. Arus lalu lintas terpantau lumayan padat akibat banjir. Hati-hati saja untuk semua pengendara. (foto:pak Fadli Via Wa-mel)
07.57: Kawasan KH Mukmin Sidoarjo terendam banjir sehingga kendaraan dari arah Candi dilewatkan Pasar Larangan.... https://t.co/bPaXf8mEBP
Buduran Sidoarjo menuju Surabaya banjir plus macet https://t.co/1HzAAsKS5F
19.08: Lokasi-lokasi ini banjir: 1.Ketintang Madya 2.Jalur Lakasantri -Menganti. Lalu lintas MACET; 3.Jetis Kulon- Simpang 3 Karangrejo Gg Makam; 4.Berbek Industri; 5.JL Raya Kandangan; 6.Balongsari; 7.Banjarsugihan 8.Wisma Tengger. 10.Satelit Utara (odp-rt) https://t.co/b3Ya9pZEQP

Dari Tabel 4.1 – 4.2 data berhasil dikumpulkan dan diketahui karakteristiknya. Selanjutnya data disimpan dalam format CSV dan dilakukan proses *preprocessing* seperti pada Gambar 3.3. Pada proses *preprocessing* dilakukan beberapa pengolahan data, hasil *preprocessing* dipaparkan secara sistematik

4.2 Hasil Preprocessing

Pengolahan data yang dilakukan pada proses ini bertujuan menormalkan data, dengan menghilangkan karakter spesial, merubah menjadi huruf kecil (*case folding*), pemotongan kata (*tokenizing*), penghapusan *stop word*, mengatasi kata gaul atau tidak formal, dan *stemming*. Hasil pengolahan data sebagai berikut:

4.2.1 Hasil Cleansing

Tahap *cleansing* menghasilkan *dataset tweet* tanpa karakter angka, emoji, tanda baca dan spasi ganda. Prinsip utama *pseudocode cleansing* adalah pengolahan *string* pada data *tweet* dengan menghilangkan karakter unik yang tidak diinginkan pada teks. *Pseudocode* 4.2 dipaparkan implementasi tahap *cleansing* pada *dataset*.

Pseudocode 4.2 *Cleansing*

Cleansing

```
algorthm:
  FOR each in text[tweet] do remove mention('@')
      FOR each in text[tweet] do remove tagger('#')
          FOR each in text[tweet] do remove link('https')
              FOR each in text[tweet] do remove numbers
                  FOR each in text[tweet] do remove emoji
                      FOR each in text[tweet] do remove
punctuation
                          IF lenght of each under 2 THEN
                              FOR each in text[tweet] do
remove each
                              with not word
                                  IF space in each do
remove spacing
                                  ENDIF
                              ENDFOR
                          ENDIF
                      ENDFOR
                  ENDFOR
              ENDFOR
          ENDFOR
      ENDFOR
  ENDFOR
```

Hasil *cleansing* terdapat pada Tabel 4.3. Tabel 4.3 menunjukan perubahan sebelum dan sesudah tahap *cleansing* dilakukan.

Tabel 4.3 Hasil *Cleansing*

Tweet	Hasil *Cleansing*
@e100ss melaporkan banjir di desa pucangro kec. Karanggeneng kab. Lamongan. Banjir kurang lbh 20-25 cm setengah ban mobil sepanjang 200m dan banyak jalan berlubang https://t.co/ckFbffEmim	melaporkan banjir di desa pucangro kec Karanggeneng kab Lamongan Banjir kurang lbh cm setengah ban mobil sepanjang dan banyak jalan berlubang
Banjir Jl. Morowudi Kulon, Cerme, Gresik. @e100ss @infoGRESIK @gresikNEWS https://t.co/8AlHcXRvUR	Banjir Jl Morowudi Kulon Cerme Gresik
Radio KARIMATA 14.11 Pak Fadli, Blega ,Bangkalan, mengabarkan, Jl. Raya Blega, Bangkalan, tepatnya di Dusun Laok Songai ketinggian air sekitar 25 cm. Arus lalu lintas terpantau lumayan padat akibat banjir. Hati-hati saja untuk semua pengendara. (foto:pak Fadli Via Wa-mel)	Radio KARIMATA Pak Fadli Blega Bangkalan mengabarkan Jl Raya Blega Bangkalan tepatnya di Dusun Laok Songai ketinggian air sekitar cm Arus lalu lintas terpantau lumayan padat akibat banjir Hati hati saja untuk semua pengendara foto pak Fadli Via Wa mel
Buduran Sidoarjo menuju Surabaya banjir plus macet https://t.co/1HzAAsKS5F	Buduran Sidoarjo menuju Surabaya banjir plus macet

Tabel 4.4 Hasil *Cleansing* Lanjutan

Tweet	Hasil *Cleansing*
07.57: Kawasan KH Mukmin Sidoarjo terendam banjir sehingga kendaraan dari arah Candi dilewatkan Pasar Larangan.... https://t.co/bPaXf8mEBP	Kawasan KH Mukmin Sidoarjo terendam banjir sehingga kendaraan dari arah Candi dilewatkan Pasar Larangan
19.08: Lokasi-lokasi ini banjir: 1.Ketintang Madya 2.Jalur Lakasantri - Menganti. Lalu lintas MACET; 3.Jetis Kulon- Simpang 3 Karangrejo Gg Makam; 4.Berbek Industri; 5.JL Raya Kandangan; 6.Balongsari; 7.Banjarsugihan 8.Wisma Tengger. 10.Satelit Utara (odp-rt) https://t.co/b3Ya9pZEQP	Lokasi lokasi ini banjir Ketintang Madya Jalur Lakasantri Menganti Lalu lintas MACET Jetis Kulon Simpang Karangrejo Gg Makam Berbek Industri JL Raya Kandangan Balongsari Banjarsugihan Wisma Tengger Satelit Utara odp rt

4.2.2 Hasil *Case Folding*

Case Folding menjadi lanjutan pengolahan data *string*. Hasil tahap *case folding* merubah teks yang tergolong berhuruf besar *uppercase* menjadi *lowercase*. *Pseudocode* 4.3 tahap *case folding* sebagai berikut.

Pseudocode 4.3 *Case Folding*

Casefolding
algorthm: *FOR each in text do* *each.lower()* *ENDFOR*

Tabel 4.4 memaparkan perubahan teks seluruhnya menjadi huruf kecil setelah dilakukan proses *case folding*.

Tabel 4.5 Hasil *Case Folding*

Tweet	**Hasil Case folding**
melaporkan banjir di desa pucangro kec Karanggeneng kab Lamongan Banjir kurang lbh cm setengah ban mobil sepanjang dan banyak jalan berlubang	melaporkan banjir di desa pucangro kec karanggeneng kab lamongan banjir kurang lbh cm setengah ban mobil sepanjang dan banyak jalan berlubang
Banjir Jl Morowudi Kulon Cerme Gresik	Banjir Jl Morowudi Kulon Cerme Gresik
Radio KARIMATA Pak Fadli Blega Bangkalan mengabarkan Jl Raya Blega Bangkalan tepatnya di Dusun Laok Songai ketinggian air sekitar cm Arus lalu lintas terpantau lumayan padat akibat banjir Hati hati saja untuk semua pengendara foto pak Fadli Via Wa mel	radio karimata pak fadli blega bangkalan mengabarkan jl raya blega bangkalan tepatnya di dusun laok songai ketinggian air sekitar cm arus lalu lintas terpantau lumayan padat akibat banjir hati hati saja untuk semua pengendara foto pak fadli via wa mel
Kawasan KH Mukmin Sidoarjo terendam banjir sehingga kendaraan dari arah Candi dilewatkan Pasar Larangan	kawasan kh mukmin sidoarjo terendam banjir sehingga kendaraan dari arah candi dilewatkan pasar larangan
Buduran Sidoarjo menuju Surabaya banjir plus macet	buduran sidoarjo menuju surabaya banjir plus macet

Tabel 4.6 Hasil *Case Folding* Lanjutan

Tweet	**Hasil *Case folding***
Lokasi lokasi ini banjir Ketintang Madya Jalur Lakasantri Menganti Lalu lintas MACET Jetis Kulon Simpang Karangrejo Gg Makam Berbek Industri JL Raya Kandangan Balongsari Banjarsugihan Wisma Tengger Satelit Utara odp rt	lokasi lokasi ini banjir ketintang madya jalur lakasantri menganti lalu lintas macet jetis kulon simpang karangrejo gg makam berbek industri jl raya kandangan balongsari banjarsugihan wisma tengger satelit utara odp rt

4.2.3 Hasil *Tokenization*

Hasil tahap *tokenization* mencetak pemisahaan kata dari data tweet. Pemotongan kata dilakukan berdasarkan *whitespace*. *Pseudocode* 4.4 dipaparkan tahap *tokenizing*.

Pseudocode 4.4 Tokenizing

Tokenizing

```
algorthm:
FUNCTION whitepace_tokenizer to the text
    CALL RegexpTokenizer for pattern
    CALL tokenizer for text
ENDFUNCTION
FOR each in text
    CALL whitespace_tokenizer(text)
ENDFOR
```

Tabel 4.5 menunjukkan bentuk perubahan tweet menjadi token dan ditampilkan pada dataframe dan berbentuk *list* python.

Tabel 4.7 Hasil Tahap *Tokenizing*

Tweet	**Token**
melaporkan banjir di desa pucangro kec karanggeneng kab lamongan banjir kurang lbh cm setengah ban mobil sepanjang dan banyak jalan berlubang	['melaporkan', 'banjir', 'di', 'desa', 'pucangro', 'kec', 'karanggeneng', 'kab', 'lamongan', 'banjir', 'kurang', 'lbh', 'cm', 'setengah', 'ban', 'mobil', 'sepanjang', 'dan', 'banyak', 'jalan', 'berlubang']
Banjir Jl Morowudi Kulon Cerme Gresik	['banjir', 'jl', 'morowudi', 'kulon', 'cerme', 'gresik']
radio karimata pak fadli blega bangkalan mengabarkan jl raya blega bangkalan tepatnya di dusun laok songai ketinggian air sekitar cm arus lalu lintas terpantau lumayan padat akibat banjir hati hati saja untuk semua pengendara foto pak fadli via wa mel	['radio', 'karimata', 'pak', 'fadli', 'blega', 'bangkalan', 'mengabarkan', 'jl', 'raya', 'blega', 'bangkalan', 'tepatnya', 'di', 'dusun', 'laok', 'songai', 'ketinggian', 'air', 'sekitar', 'cm', 'arus', 'lalu', 'lintas', 'terpantau', 'lumayan', 'padat', 'akibat', 'banjir', 'hati', 'hati', 'saja', 'untuk', 'semua', 'pengendara', 'foto', 'pak', 'fadli', 'via', 'wa', 'mel']
buduran sidoarjo menuju surabaya banjir plus macet	['buduran', 'sidoarjo', 'menuju', 'surabaya', 'banjir', 'plus', 'macet']

Tabel 4.8 Hasil *Tokenaizing* Lanjutan

Tweet	**Token**
lokasi lokasi ini banjir ketintang madya jalur lakasantri menganti lalu lintas macet jetis kulon simpang karangrejo gg makam berbek industri jl raya kandangan balongsari banjarsugihan wisma tengger satelit utara odp rt	['lokasi', 'lokasi', 'ini', 'banjir', 'ketintang', 'madya', 'jalur', 'lakasantri', 'menganti', 'lalu', 'lintas', 'macet', 'jetis', 'kulon', 'simpang', 'karangrejo', 'gg', 'makam', 'berbek', 'industri', 'jl', 'raya', 'kandangan', 'balongsari', 'banjarsugihan', 'wisma', 'tengger', 'satelit', 'utara', 'odp', 'rt']
kawasan kh mukmin sidoarjo terendam banjir sehingga kendaraan dari arah candi dilewatkan pasar larangan	['kawasan', 'kh', 'mukmin', 'sidoarjo', 'terendam', 'banjir', 'sehingga', 'kendaraan', 'dari', 'arah', 'candi', 'dilewatkan', 'pasar', 'larangan']

4.2.4 Hasil *Stop word*

Pseudocode tahap *stopword* melakukan *filter* dan menghilangkan kata umum yang dianggap tidak mengandung informasi penting. Dengan mengambil data eksternal kata umum berformat *txt* dan diubah dalam format *list python*. Penelitian ini menggunakan *stop word* yang dikumpulkan oleh Owen. *Pseudocode* 4.5 tahap stopword dipaparkan.

Pseudocode 4.5 *Stop Word*

Stopword
Algorithm: *FOR term in text[tweet] match with key* *remove term* *ENDFOR*

Terdapat 675 kata umum bahasa Indonesia yang digunakan (Owen, 2022). Contoh kata umum yang dihilangkan pada text tweet sebagai berikut: “di”, “adalah”, “gimana”, “yakni”, “dengan”, ”orang”, “bahwa”, “namun”, “dua”, “kepada” , ”lalu”, “lain”, “banyak”, “beberapa”, “besar”, “merupakan”. Dari *Pseudocode* 4.5 ditampilkan perubahan setelah dilakukan tahap *stop word* sebagaimana Tabel 4.6.

Tabel 4.9 Hasil *Tahap Stopword*

Token	**Hasil *Filter Stop word***
['melaporkan', 'banjir', 'di', 'desa', 'pucangro', 'kec', 'karanggeneng', 'kab', 'lamongan', 'banjir', 'kurang', 'lbh', 'cm', 'setengah', 'ban', 'mobil', 'sepanjang', 'dan', 'banyak', 'jalan', 'berlubang']	['melaporkan', 'banjir', 'desa', 'pucangro', 'kec', 'karanggeneng', 'kab', 'lamongan', 'banjir', 'lbh', 'cm', 'ban', 'mobil', 'jalan', 'berlubang']
['banjir', 'jl', 'morowudi', 'kulon', 'cerme', 'gresik']	['banjir', 'jl', 'morowudi', 'kulon', 'cerme', 'gresik']

Tabel 4.10 Hasil Tahap Tokenizing Lanjutan

Token	Hasil *Filter Stop word*
['radio', 'karimata', 'pak', 'fadli', 'blega', 'bangkalan', 'mengabarkan', 'jl', 'raya', 'blega', 'bangkalan', 'tepatnya', 'di', 'dusun', 'laok', 'songai', 'ketinggian', 'air', 'sekitar', 'cm', 'arus', 'lalu', 'lintas', 'terpantau', 'lumayan', 'padat', 'akibat', 'banjir', 'hati', 'hati', 'saja', 'untuk', 'semua', 'pengendara', 'foto', 'pak', 'fadli', 'via', 'wa', 'mel']	['radio', 'karimata', 'fadli', 'blega', 'bangkalan', 'mengabarkan', 'jl', 'raya', 'blega', 'bangkalan', 'tepatnya', 'dusun', 'laok', 'songai', 'ketinggian', 'air', 'cm', 'arus', 'lintas', 'terpantau', 'lumayan', 'padat', 'akibat', 'banjir', 'hati', 'hati', 'pengendara', 'foto', 'fadli', 'via', 'wa', 'mel']
['kawasan', 'kh', 'mukmin', 'sidoarjo', 'terendam', 'banjir', 'sehingga', 'kendaraan', 'dari', 'arah', 'candi', 'dilewatkan', 'pasar', 'larangan']	['kawasan', 'kh', 'mukmin', 'sidoarjo', 'terendam', 'banjir', 'kendaraan', 'arah', 'candi', 'dilewatkan', 'pasar', 'larangan']
['buduran', 'sidoarjo', 'menuju', 'surabaya', 'banjir', 'plus', 'macet']	['buduran', 'sidoarjo', 'menuju', 'surabaya', 'banjir', 'plus', 'macet']
['lokasi', 'lokasi', 'ini', 'banjir', 'ketintang', 'madya', 'jalur', 'lakasantri', 'menganti', 'lalu', 'lintas', 'macet', 'jetis', 'kulon', 'simpang', 'karangrejo', 'gg', 'makam', 'berbek', 'industri', 'jl', 'raya', 'kandangan', 'balongsari', 'banjarsugihan', 'wisma', 'tengger', 'satelit', 'utara', 'odp', 'rt']	['lokasi', 'lokasi', 'banjir', 'ketintang', 'madya', 'jalur', 'lakasantri', 'menganti', 'lintas', 'macet', 'jetis', 'kulon', 'simpang', 'karangrejo', 'gg', 'makam', 'berbek', 'industri', 'jl', 'raya', 'kandangan', 'balongsari', 'banjarsugihan', 'wisma', 'tengger', 'satelit', 'utara', 'odp']

4.2.5 Hasil *Handle Slang Word*

Pseudocode tahap *handle slang word* melakukan proses mapping kata tidak formal baik bahasa gaul ataupun kependekan kata pada *tweet* dan diubah menjadi kata formal bahasa Indonesia. Pada proses ini bergantung pada seberapa banyak kata yang dianggap bahasa gaul, kependekan, bahasa tidak formal yang terdapat pada *corpus* yang digunakan pada penelitian ini. *Pseudocode* 4.6 tahap *handle slang word* dipaparkan.

Pseudocode 4.6 *Slang Word*

```
Slangword
Algorithm:
SET key,value in slangword_dict as term
FOR term in text[tweet] match with key
    change key with value
ENDFOR
```

Kata tidak formal yang digunakan penelitian terdapat pada *corpus Colloquial Indonesian Lexicon* (Salsabila dkk., 2018) dan IndoCollex (Wijaya, 2021) berjumlah 6773 kata. Contoh tidak formal dan bahasa formal pada corpus yaitu “sblm” : “sebelum”, “krn” : ”karena”, “nyebelin” : “menyebalkan”, “min”:”admin”, Tabel 4.7 memaparkan contoh perubahan pada *data set*.

Tabel 4.11 Hasil Tahap *Slang Word*

Tweet	Hasil Handle *Slang word*
['kemacetan', 'padat', **'sblm'**, 'pabrik', 'kopi', 'kapal', 'api', 'trosobo', 'arah', **'sby',** 'krian', 'hr', 'macet', 'akibat', 'banjir', 'skr', 'krn', 'bis', 'truck', 'mogok']	['kemacetan', 'padat', 'sebelum', 'pabrik', 'kopi', 'kapal', 'api', 'trosobo', 'arah', **'surabaya'**, 'krian', 'hari', 'macet', 'akibat', 'banjir', 'sekarang', **'karena',** 'bis', 'truk', 'mogok']

4.2.6 Hasil *Stemming*

Tahap stemming dilakukan untuk merubah kata berimbuhan *prefix* atau *suffix* menjadi kata dasar. Penelitian ini menggunakan *library sastarwi.* Sehingga *Pseudocode* tahap *stemming* membangun aturan berdasarkan *library sastrawi* dengan data token *tweet* yang ada dan menghasilkan kamus aturan penghapusan *prefix* dan *suffix*. Kemudian *pseudocode* merubah token sesuai kamus yang dibangun yaitu terdiri 6128 token. Berikut *Pseudocode* 4.7 *stemming* dijalankan.

Pseudocode 4.7 *Stemming*

Stemming

```
FUNCTION stemmed_wrapper(term)
#build term wrapper from libary
    CALL Stemmer.stem(term)
ENDFUNCTION

SET term_dict
FOR document in text[tweet]
    FOR term in document
        IF term in term_dict then
            APPEND term in the term_dict
        else
            APPEND ' ' in term_dict
        ENDIF
    ENDFOR
ENDFOR

FOR term in term_dict do term_dict[term] =
stemmed_wrapper(term)
    OUTPUT term,":" ,term_dict[term]

#applying to dataframe
FUNCTION todo_stemmed_term
    FOR term in document do term_dict[term]
    ENDFOR
ENDFUNCTION
CALL todo_stemmed_term to the dataframe
```

Tabel 4.9 memaparkan contoh sampel token hasil *stemming* pada *tweet.* Dimana contohnya pada token semalam imbuhan dihilankan menjadi kata dasar malam dan seterusnya.

Tabel 4.12 Hasil Tahap *Stemming*

Token	*Stemming*
semalam	malam
anginnya	angin
menimbulkan	timbul
hujannya	hujan
terbawa	bawa
pembagian	bagi
kebanjiran	banjir
tuntaskan	tuntas
kerugian	rugi
menyusuri	susur

4.3 Hasil *Filtering Duplicate*

Pada tahap *filtering duplicate* ditemukan 384 *tweet* duplikasi. *Pseudocode* 4.8 dipaparkan tahap penghapusan.

Pseudocode 4. 8 *Filtering Duplicate*

```
Filtering Duplicate
Algorthm:
CALL duplicates #to check what the text has duplicate
IF text[tweet] has duplicate then
    SET keep = 'first' #for not delete the first text
    SET inplace boolean True #for delete the duplicate
from existing datafram
    CALL drop_duplicates
ELSE
    Keep text[tweet]
ENDIF
```

Namun *Pseudocode* 4.7 tahap *filtering duplicate* tidak bisa mengatasi penghapusan kata kunci yang mengandung konten bukan bencana banjir seperti banjir hadiah, lokasi bebas banjir. Sehingga dilakukan proses penghapusan secara manual. Tabel 4.9 contoh teks *tweet* yang dihapus secara manual.

Tabel 4.13 Contoh *Tweet* yang Dihapus Manual

Tweet	Tweet (Setelah Proses Preprocessing)
Banjir Gol, Persela Lamongan Hajar Barito Putera	['banjir', 'gol', 'sela', 'lamongan', 'hajar', 'barito', 'putra']
Candi Elektronik Sale (CES) BANJIR HADIAH!!! Ayo belanja di Candi Elektronik sebelum 31 Desember 2021! Banyak diskonnya, banyak hadiahnya!	['candi', 'elektronik', 'sale', 'ces', 'banjir', 'hadiah', 'ayo', 'belanja', 'candi', 'elektronik', 'desember', 'diskon', 'hadiah']

4.4 Hasil *Labelling*

Setelah *filtering duplicate*, *dataset* yang siap berjumlah 1960 tweet. Dilakukan tahap *Data labelling*. Prinsip penting pada *pseudocode* penetapan label adalah pemberian aturan *begin inside dan other* pada token. Penelitian telah

menetapkan 5 label seperti pada aturan Tabel 3.2. yaitu B-location, I-location, B-event, I-event, dan Other. Penetapan label BIO dipaparkan pada *Pseudocode* 4.9 dan *pseudocode* 4.10.

Pseudocode 4.9 Penetapan *Named Entity*

Named Entity

```
Algorithm:
#term to assign ne in dataset
FUNCTION ne_term assign ne in dataframe.
    FOR term in text[tweet] do
        IF term in ne_dict[term] then
            add value of term
            # mapping token as location or event
            ELSE
            Fillna with "O"
        ENDIF
    ENDFOR
ENDFUNCTION
#apply function ne term to the dataframe and save in
column ne
CALL text[tweet].apply(ne_term) #to the column then print
out to the column ne
```

Setelah *Pseudocode* 4.10 dijalankan token memiliki *named entity location* dan *event* berdasarkan *dictionary* yang telah dibangun sebelumnya. kemudian dilanjukan *Pesudocode* 4.11 untuk menetapkan BIO format pada *named entity*. Dimana jika pertama adalah O maka tag selanjutnya adalah Begin dan Jika tag sebelumnya adalah Begin maka Tag selanjutnya Inside, Namun Jika Tag sebelumnya adalah Inside dan berentitas bukan *location* ataupun *event* maka ditetapkan sebagai *Other*/*Outside*.

Pseudocode 4.10 Penetepan BIO *Labelling*

BIO

```
Algoritm:
#rule to assign bio format
FUNCTION bio_term #assign bio format
    FOR tag in tagged do
        IF tag is "O" THEN
            ASSIGN prev_tag as none
            ASSIGN new_tag as "O"
        ELSEIF tag is not "O" and didnt have prev_tag THEN
            APPEND new_tag with "B-tag"
            ASSIGN prev_tag as "B-tag"
        ELSEIF tag is not "O" and the prev_tag is start
with "B-" THEN
            APPEND new_tag with "I-tag"
            ASSIGN prev_tag as "I-tag"
```

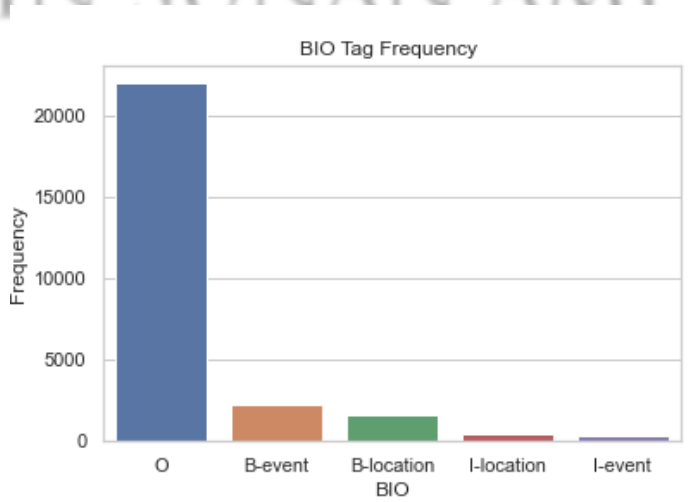
```
        ELSE tag is not o and the prev_tag is start with
"I-" THEN
            APPEND new_tag with "I-tag"
            ASSIGN prev_tag as "I-tag"
        ENDIF
    ENDFOR
ENDFUNCTION
# apply function to the dataframe
CALL text[tweet].apply(bio_term)
```

Tabel 4.10 contoh penetapan hasil BIO. Dari tersebut diketahui *pseudocode* berjalan dengan baik sesuai dengan anturan tabel 3.2 dan dapat mengenali entitas *event* pada kata banjir, entitas *location* kecamatan cerme sebagai B-location dan I-location pada kabupaten Gresik. Namun belum dictionary masih terbatas belum mencakup entitas lokasi jalan.

Tabel 4.14 Contoh hasil BIO

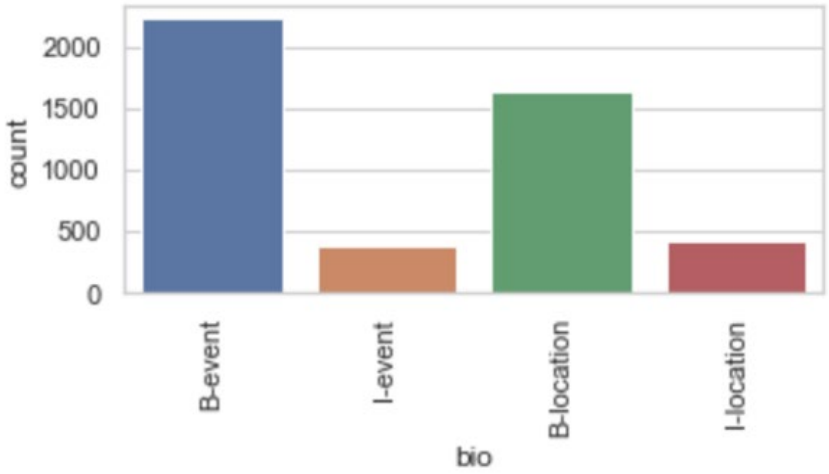
Token	**BIO**
banjir	B-Event
jalan	O
morowudi	O
kulon	O
cerme	B-location
gresik	I-location

Selain itu Gambar 4.2 memaparkan distribusi hasil penetapan bio format. Berdasarkan Gambar 4.2 diketahui terdapat 26679 kata telah ditetapkan nama entitasnya. Sebagian besar kata ditetapkan sebagai *outside/other* diluar potongan *begin* atau *inside*. *entitas Other* berjumlah 22019 kata. Kata – kata ini dapat dianggap sebagai pengisi atau pelengkap dan kehadirannya dapat mempengaruhi kinerja model.



Gambar 4.1 Diagram BIO dengan Entitas *Other*

Gambar 4.3 memaparkan *dataset* tanpa *named entity other* diketahui sebagian besar data terkait kata peristiwa dan lokasi kejadian peristiwa dengan format *begin* yang lebih besar dibandingkan jumlah *inside*.



Gambar 4.2 Diagram BIO Tanpa Entitas *Other*

4.5 Hasil Penerapan Conditional Random Fields

Penerapan *Conditional Random Fields* telah dilakukan dengan menggunakan *wrapper sklearn-crfsuite* pada dataset *twitter* yang telah disiapkan dari tahap *preprocessing* hingga *labelling*. *Pseudocode* berikut memerankan fungsi mengambil POS dan entitasnya.

4.5.1 *Extraction Feature*

Sebelum melakukan tahap *extraction feature*, dilakukan penetapan POS Tag *Pseudocode* 4.11 akan memerankan proses pemberian label *part of speech* (POS) atau kelas kata pada token.

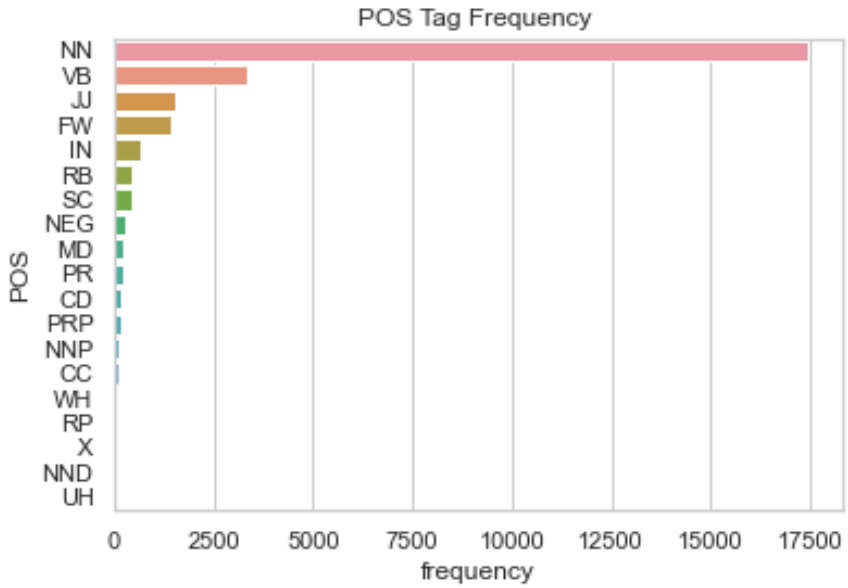
*Pseudocod*e 4.11 *POS Tagging*

Postaging

```
Algorithm:
IF text[tweet] with true format then
    CALL ct.tag_sent to apply tagging to the text[tweet]
    then printout to the new column postag
ENDIF
```

Hasil pemberian kelas kata dipaparkan pada Gambar 4.3. berdasarkan Gambar 4.3 diketahui terdapat 26679 kata dengan rician POS NN (*Noun*) memiliki jumlah sebanyak 17438. Kemudian disusul dengan POS VB (*Verb*) sebanyak 3350. POS JJ (*Adjective*) menempati posisi ketiga sebanyak 1549. Dan posisi terakhir

pada POS UH (*Interjection*) sebanyak 3 kata. Dengan informasi tersebut diketahui bahwa dataset memiliki proporsi yang tidak seimbang pada kelas kata *noun*.



Gambar 4.3 *POS Tag Frequency*

Meskipun begitu tidak ada penanganan khusus untuk mengatasi proporsi yang tidak seimbang pada kelas kata NN. Karena pada Tabel 4.10 memaparkan kata kunci penting misalnya banjir, hujan dan gresik di tetapkan sebagai POS NN, dan NNP. Sebanyak 2160 kata banjir, 389 untuk kata Gresik dan seterusnya.

Tabel 4.15 Contoh *Word Frequency POS Noun*

Word	**Jumlah**
Banjir	2160
Gresik	389
Jalan	324
Hujan	313
Desa	278

Selanjutnya dilakukan tahap *extraction feature* untuk mengekstrak fitur yang digunakan, yaitu bagian kata, *pos tag, title, upper, lower,* dan *nearby word*. Kemudian fitur – fitur tersebut dikonversi kedalam format *library scikit-learn*. *Pseudocode* 4.12 menjalankan proses konversi ekstraksi fitur.

Pseudocode 4.12 *Extraction Feature*

Extraction Fitur

Algorithm:
`FUNCTION word2feature(array which result came from pos and bio labbeling as sent, index of word)`
`SET word[i][0]`

```
            SET Postag[i][0]
            SET array features with {
                'bias',
                'word.lower()',
                'word[-3:]',
                'word[-2:]',
                'word.isupper()',
                'postag',
                'postag[:2]'
            }

            IF word of sent is not first word THEN
                SET word1 = sent[i-1][0]
                SET postag1 = sent[i-1][1]
                UPDATE features with {
                    '-1:word.lower()': word1.lower(),
                    '-1:word.istitle()': word1.istitle(),
                    '-1:word.isupper()': word1.isupper(),
                    '-1:postag': postag1,
                    '-1:postag[:2]': postag1[:2],
                }
            ELSE
                SET features BOS is boolean True
            ENDIF

            IF word of sent is not the last word THEN
                SET
                SET
                UPDATE features with {
                    '+1:word.lower()': word1.lower(),
                    '+1:word.istitle()': word1.istitle(),
                    '+1:word.isupper()': word1.isupper(),
                    '+1:postag': postag1,
                    '+1:postag[:2]': postag1[:2],
                }
            ELSE
                SET features EOS is boolean True
            ENDIF
ENDFUNCTION
```

kemudian dilakukan penerapan ke dalam *dataset*. Dimana *Extraction Feature* memberikan output bilangan ril walaupun hanya 0 atau 1. *Pseudocode* 4.13 proses ekstraksi fitur ke dalam dataset.

Pseudocode 4.13 *Extraction Feature* Lanjutan

Ektraction Feature 2

```
Algorithm:
SET X_train as array with [
    FOR each in train_sents do
        CALL sent2features to each
    ENDFOR
]
SET y_train as array with [
    FOR each in train_sents do
        CALL sent2labels to each
```

```
    ENDFOR
]
SET X_test as array with [
    FOR each in test_sents do
        CALL sent2features to each
    ENDFOR
]
SET y_test as array with [
    FOR each in test_sents do
        CALL sent2labels to each
    ENDFOR
]
```

4.5.2 *Training* Model

Training model *Conditional Random Fields* menggunakan *kfold* = 5. Skenario dilakukan dengan proporsi *dataset* sebanyak 80% untuk *training* model yaitu sejumlah 1568 data dan 20% *testing* model sejumlah 392 data.

Pada *wrapper sklearn crf* terlebih dahulu akan melakukan *training* data dan mengirimkan hasil *training* data dengan label yang sesuai. Penelitian telah membangun model CRFs dengan konfigurasi *algorithm lbfgs*, parameter *c1* dan *c2* masing – masing sebesar 0.1. Kemudian adanya *dataset* yang kecil maksimal iterasi yang dilakukan sebanyak 100 kali dengan *possible transition boolean true* dan *verbose boolean true*. Dengan konfigurasi tersebut training model diselesaikan dengan *kfold* sebanyak 5 kali pengulangan. Sehingga dapat menyelesaikan training model dengan beberapa detik. Baris *pseudocode* 4.14 yang dijalankan sebagai berikut.

Pseudocode 4. 14 *Training Model* CRFs

Training model
Algorithm: *FOR each in 5 fold dataset do* *SET X_train, y_train for trainer* *CALL model.fit(X_train,y_train) to training dataset X_train, y_train* *ENDFOR*

Penggunaan algoritma lbfgs berfungsi sebagai algotitma pilihan untuk melakukan *Fitting* model CRFs dengan nilai parameter c1 dan c2. Inisiasi penimbang akan memberikan progress proses training saat model dilatih terhadap data training yang disediakan.

4.5.3 Hasil Pengujian Model

Tujuan dari percobaan penelitian ini adalah untuk menemukan model NER deteksi peristiwa banjir. Evaluasi model ner dengan skenario pengujian *precision, recall* dan *f-measure* pada data testing hasil Kfold sebanyak 5 *fold* dan terdiri dari 392 data training. *Pseudocode* 4.15 proses pengujian Model CRFs

Pseudocode 4. 15 *Testing Model*

```
Testing
Algorithm:
FOR index in 5 fold do
CALL model.predict (X_test) for testing OUTPUT as
pred_value
THEN calculate Precision, Recall, F-measure (y_true,
pred_value)
THEN Calculate Mean of Precision, Recall, F-measure
ENDFOR
```

Tabel 4.12 menunjukkan matrik hasil pengujian berbasis *named entity* per fold partisi. Laporan tersebut menunjukkan presisi metrik klasifikasi *precision, recall* dan *f-measure* pada basis entitas. Metrik dihitung dengan menggunakan positif benar dan salah, negatif benar dan salah. Positif dan negatif dalam hal ini adalah nama entitas untuk kelas yang diprediksi.

Tabel 4.16 Hasil Evaluasi Entitas

Nomor *Fold*	***Entitas***	***Precision***	***Recall***	***F-Measure***
K1	*B-event*	0.978	0.998	0.988
	I-event	0.988	0.914	0.950
	B-location	0.955	0.719	0.820
	I-location	0.905	0.600	0.721
K2	*B-event*	0.968	0.997	0.983
	I-event	0.991	0.884	0.934
	B-location	0.980	0.829	0.890
	I-location	0.986	0.960	0.973
K3	*B-event*	0.995	1	0.997
	I-event	1	0.991	0.995
	B-location	0.984	0.917	0.949
	I-location	0.982	0.902	0.940
K4	*B-event*	0.995	1	0.997
	I-event	1	0.990	0.994
	B-location	0.983	0.963	0.973
	I-location	0.942	0.852	0.894
K5	*B-event*	0.985	1	0.992
	I-event	1	0.932	0.965
	B-location	0.978	0.762	0.857
	I-location	0.976	0.677	0.800

Berdasarkan Tabel 4.16 diketahui evalusi nilai *precision* pada semua *entity* diprediksi dengan nilai yang tinggi diatas 90%. Sedangkan nilai *recall* pada entitas B-location dan I-location memiliki nilai daya ingat yang cukup baik meskipun pada nomor fold K1 entity *B-location* dan *I-location* masing – masing memiliki nilai 0.719 dan 0.600. Sedangkan nilai *f-measure* adalah rata-rata harmonik tertimbang dari precision dan recall sehingga skor terbaik adalah 1,0 dan yang terburuk adalah 0,0. Maka berdasarkan Tabel 4.11 nilai *f-measure* setiap fold memiliki nilai yang mendekati angka 1. Dari hasil Tabel 4.11 matrik per *named entitys* didapatkan hasil rata – rata nilai *precision, recall,* dan *f-measure* pada Tabel 4.12. Berdasarkan rata – rata, model mempunyai nilai *Precision* sebesar 0.981, *Recall* sebesar 0.926 dan *f-measure* sebesar 0.950.

Tabel 4.17 Hasil Rata - Rata Evaluasi Model *Precision, Recall* dan *F-Measure*

Nomor Fold	***Precision***	***Recall***	***F-Measure***
K1	0.969	0.894	0.925
K2	0.975	0.922	0.946
K3	0.990	0.956	0.972
K4	0.985	0.969	0.977
K5	0.983	0.899	0.929
Rata - Rata	0.981	0.926	0.950

Dengan rata – rata *f-measure* sebesar 0.950 maka dapat diketahui model dapat bekerja dengan baik. Hal ini juga menunjukan nilai parameter *c1* = 0.1 dan *c2* = 0.1 telah memberikan nilai *f-measure* yang stabil.

4.5.4 Hasil Deteksi Banjir

Dengan hasil pengujian tersebut, model dapat mendeteksi entity *event* dengan *B-event* dan *I-event* dan entitas lokasi dengan label *B-location* dan *I-location* kejadian banjir. Tabel 4.18 memaparkan sampel hasil label deteksi banjir dengan format *list tuple*.

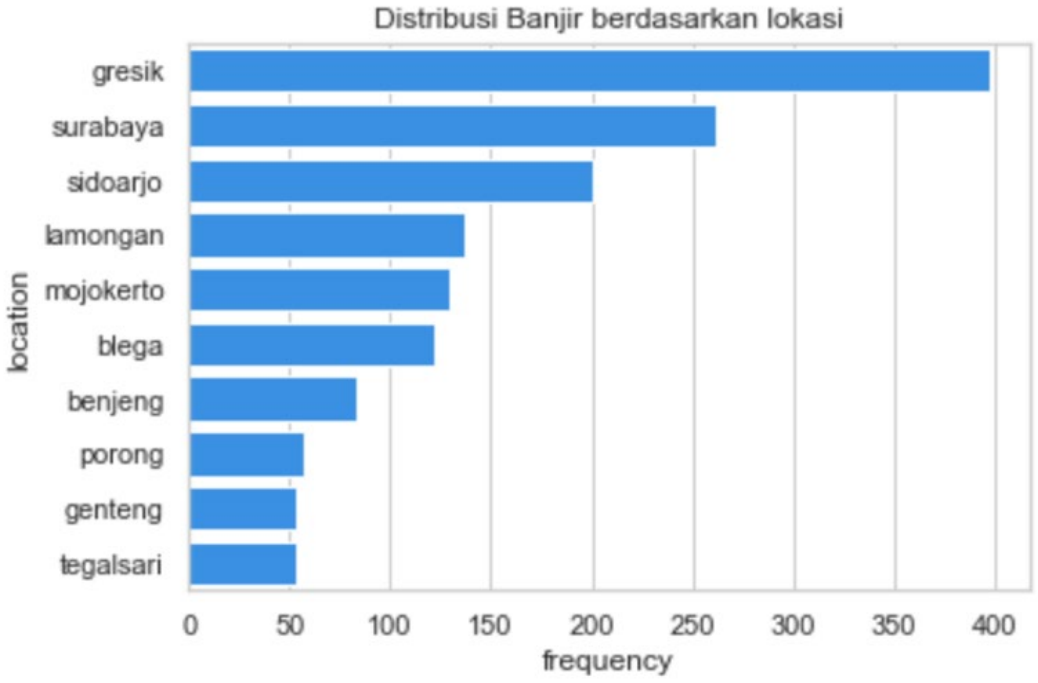
Tabel 4.18 Hasil Deteksi Banjir

Tweet ke-	**Hasil Deteksi**
4	[('genang', 'B-event'), ('hampir', 'O'), ('ruas', 'O'), ('jalan', 'O'), ('jalan', 'O'), ('dari', 'O'), ('wahidin', 'O'), ('sudirohusodo', 'O'), ('jalan', 'O'), ('utama', 'O'), ('gkb', 'O'), ('pasar', 'O'), ('gresik', 'B-location'), ('empat', 'O'), ('nippon', 'O'), ('paint', 'O'), ('banjir', 'B-event'), ('capai', 'O'), ('mata', 'O'), ('kaki', 'O'), ('arus', 'O'), ('deras', 'B-event'), ('arah', 'O'), ('jalan', 'O'), ('veteran', 'O'), ('hati', 'O'), ('hati', 'O')]
110	[('jalur', 'O'), ('utama', 'O'), ('mojokerto', 'B-location'), ('surabaya', 'I-location'), ('macet', 'O'), ('parah', 'O'), ('hambat', 'O'), ('banjir', 'B-event'), ('trosobo', 'O')])]

Tabel 4.19 Hasil Deteksi Banjir Lanjutan

Tweet ke-	Hasil Deteksi
270	[('banjir', 'B-event'), ('terjang', 'O'), ('gresik', 'B-location'), ('rumah', 'O'), ('sekolah', 'O'), ('rendam', 'O')]
600	[('curah', 'O'), ('hujan', 'B-event'), ('jalan', 'O'), ('raya', 'O'), ('porong', 'B-location'), ('sidoarjo', 'I-location'), ('jawa', 'O'), ('timur', 'O'), ('rendam', 'O'), ('banjir', 'B-event'), ('meter', 'O')]
1011	[('infosurabaya', 'O'), ('nuryanto', 'O'), ('infosurabaya', 'O'), ('jalan', 'O'), ('biliton', 'O'), ('gubeng', 'B-location'), ('banjir', 'I-event'), ('mata', 'O'), ('kaki', 'O')]
1711	[('tujuh', 'O'), ('desa', 'O'), ('cerme', 'B-location'), ('gresik', 'I-location'), ('genang', 'I-event'), ('banjir', 'I-event')]

Berdasarkan hasil Tabel 4.18 maka Gambar 4.4 menggambarkan distribusi lokasi kejadian banjir di Kawasan Gerbang Kertosusila. Hasil Gambar 4.4 didapatkan dengan menghitung frekuensi kejadian banjir berdasarkan *filtering tweet* yang mempunyai label *B-location* dan *I-location*. Distribusi frekuensi banjir diketahui lokasi Gresik menjadi lokasi terjadinya banjir tertinggi sebanyak 398, kemudian Surabaya sebanyak 262 dan sidoarjo sebanyak 201. Sehingga dengan model ini dapat diterapkan pada tweet banjir untuk mengetahui lokasi kabupaten dan kecamatan terjadinya peristiwa banjir.



Gambar 4.4 Distribusi Lokasi Banjir

4.6 Pembahasan

Khodra mengemukakan riset untuk mengenali entitas *name, place, time, info* dan *other*. Model dibangun menambahkan tahapan *filter module* berbasis *rule based* dan *extraction feature module*. Kombinasi pada *extraction feature module* terdiri dari metode tokenisasi multi token, POS *tag* dan set fitur pada pengaturan CRFs (Muhammad dan Khodra, 2015). Hasil *Accuracy* yang didapatkan adalah 75%. Berbeda dengan Khodra, Yuda Munarko menerapkan riset *Named Entity Recognition* (NER) bahasa indonesia dari dataset twitter menjadi tiga kategori yaitu informal, formal dan *mixed*. *Preprocessing* yang digunakan pada riset tersebut yaitu *lower case* dan *tokenizing* untuk mengenali entitas *person, location, organization* dan *other*. Riset tersebut mendapatkan nilai *precision* dan *recall* masing - masing 87% dan 62% untuk formal *tweet*, 90% dan 36% informal *tweet*, dan 86% dan 60% untuk *mixed tweet* (Y Munarko dkk., 2018)

Penelitian ini menggunakan skenario Kfold = 5 pada *testing* dan memberikan nilai *f-measure* dengan rata - rata mencapai 0.950. *Training* CRFs dilakukan dengan algoritma lbgsf, dan kombinasi nilai c1 = 0.1, dan nilai c2 = 0.1. Berbeda dengan riset sebelumnya yang menggunakan *preprocessing* standar (Y Munarko dkk., 2018), riset ini menambahkan *handle slang word* untuk mengatasi bahasa informal atau kata gaul pada tweet. Pada proses *data labelling* dilakukan dengan BIO format pada *named entity*. Adapun *feature* yang ditambahkan adalah *pos tag*. Model didapatkan dengan rasio skenario *testing* 80% data *training* dan 20% data testing memberikan nilai *precision* sebesar 0.981 *recall* sebesar 0.926, dan *f-measure* sebesar 0.950.

Dari riset ini ditemukan *insight* baru bahwa adanya kombinasi parameter nilai c1 = 0.1 dan c2 = 0.1 memberikan nilai *f-measure* yang stabil. Dan juga menunjukkan pemilihan komposisi pada tahap *preprocessing* dan *feature extraction* mempengaruhi performa *precision, recall* dan *f-measure* model CRF. Ternyata penanganan *slang word* mampu meningkat nilai *precision* sebesar 12% dan meningkatkan nilai *recall* sebesar 33% jika dibandingkan dengan penelitian (Y Munarko dkk., 2018).

Walaupun nilai *precision* yang cukup tinggi untuk semua *testing* pada riset ini atau riset (Muhammad dan Khodra, 2015; Y Munarko dkk., 2018). Namun, memilih komposisi dan kombinasi model yang tepat terhadap data *testing* akan menghasilkan performa yang lebih baik. Selain itu pengujian silang menggunakan Kfold dapat meningkatkan performa model karena memberikan dataset *testing* dengan proporsi yang seimbang.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan evaluasi yang dilakukan. Berikut kesimpulan dari penelitian ini.

1. Implementasi metode *Conditional Random Fields* untuk deteksi peristiwa banjir berhasil mengenali *entity event* dan *location* berformat BIO (B-*location*, B-*event*, I-*location* dan I-*event*). Dengan model yang telah dibangun diketahui distribusi terjadinya banjir di kawasan Gerbang Ketosusila. Adapun wilayah Gresik menjadi wilayah yang paling sering terjadi banjir sebanyak 398, dilanjutkan dengan Surabaya sebanyak 262, Sidoarjo sebanyak 201 dan seterusnya.
2. Evaluasi dari skenario *kfold* = 5, model telah memberikan rata - rata di atas 90%. Dengan rasio 20% data *testing* dan 80% data training. Skenario tersebut memberikan nilai rata – rata *precision* 0.981, *recall* 0.926, dan *f-measure* sebesar 0.950. Artinya model yang dibangun sudah sangat baik dengan nilai akurasi yang tinggi.

5.2 Saran

1. Data pada penelitian terbilang cukup terbatas karena masih mencakup kejadian banjir di Kawasan Gerbang Kertosusila berdasarkan data Twitter. sehingga diperlukan perluasan wilayah riset untuk memberikan lingkup data yang lebih besar. Seperti menggabungkan berbagai sumber data media sosial atau menambahkan cakupan wilayah menjadi seluruh Indonesia.
2. Implementasi *Conditional Random Fields* baik digunakan dalam mendeteksi *entity*. Namun belum berhasil untuk mengenali konteks ataupun kategorisasi kejadian banjir. sehingga masih diperlukan penelitian lanjutan dengan menggunakan algoritma lain yang mampu mengatasi konteks seperti BERT.

DAFTAR PUSTAKA

Ahmed, W., Bath, P. A., dan Demartini, G. (2017): USING TWITTER AS A DATA SOURCE: AN OVERVIEW OF ETHICAL, LEGAL, AND METHODOLOGICAL CHALLENGES, *Emerald Publishing Limited*, **2**, 79–107. https://doi.org/https://doi.org/10.1108/S2398-601820180000002004

Amershi, S. (2009): Combining Unsupervised and Supervised Classification to Build User Models for Exploratory Learning Environments, **1**(1), 1–54.

Ann Copestake (2004): Natural Language Processing 2004, 8, 2003–2004.

Awalludin, M. H., Teknik, F., Indonesia, U. K., dan Bandung, J. D. (2018): EVENT DETECTION PADA MICROBLOGGING TWITTER DENGAN METODE DENCLUE UNTUK PEMETAAN LOKASI BENCANA LONGSOR, *JBPTUNIKOMPP*, diperoleh melalui situs internet: https://repository.unikom.ac.id/id/eprint/58405.

Azarine, I. S., Bijaksana, M. A., dan Asror, I. (2019): Named entity recognition on Indonesian tweets using hidden markov model, *2019 7th International Conference on Information and Communication Technology, ICoICT 2019*, 1–5. https://doi.org/10.1109/ICoICT.2019.8835277

Baranowski, D. B., Flatau, M. K., Flatau, P. J., Karnawati, D., Barabasz, K., Labuz, M., Latos, B., Schmidt, J. M., Paski, J. A. I., dan Marzuki (2020): Social-media and newspaper reports reveal large-scale meteorological drivers of floods on Sumatra, *Nature Communications*, **11**(1), 1–10. https://doi.org/10.1038/s41467-020-16171-2

Béchet, F., dan Mohit, B. (2011): Named Entity Recognition, *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, 257–290. https://doi.org/10.1002/9781119992691.ch10

BNPB (2021): Kejadian Banjir di Indonesia, diperoleh 11 Agustus 2022, melalui situs internet: https://bnpb.go.id/infografis/kejadian-bencana-tahun-2020.

Charoenpong, J., Pimpunchat, B., Amornsamankul, S., dan Triampo, W. (2019): A Comparison of Machine Learning Algorithms and their Applications, 1–17. https://doi.org/10.5013/IJSSST.a.20.04.08

CNN Indonesia (2021): Jatim Alami 65 Bencana Hidrometeorologi di Januari, Banjir 49, , diperoleh 7 Maret 2021, melalui situs internet: https://www.cnnindonesia.com/nasional/20210206140442-20-603103/jatim-alami-65-bencana-hidrometeorologi-di-januari-banjir-49.

Dinakaramani, A., Rashel, F., Luthfi, A., dan Manurung, R. (2014): Designing an Indonesian part of speech tagset and manually tagged Indonesian corpus, *Proceedings of the International Conference on Asian Language Processing 2014, IALP 2014*, 66–69. https://doi.org/10.1109/IALP.2014.6973519

Ermawati, M., dan Buliali, J. L. (2018): Text Based Approach For Similar Traffic Incident Detection from Twitter, *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, **9**(2), 63. https://doi.org/10.24843/lkjiti.2018.v09.i02.p01

Grimley, B. N. (2016): What is NLP ? The development of a grounded theory of Neuro-Linguistic Programming , (NLP), within an action research journey . Implications for the use of NLP in coaching psychology, **11**(2).

Jaariyah, N., dan Rainarli, E. (2017): Conditional Random Fields Untuk Pengenalan Entitas Bernama Pada Teks Bahasa Indonesia, *Komputa : Jurnal Ilmiah Komputer dan Informatika*, **6**(1), 29–34. https://doi.org/10.34010/komputa.v6i1.2474

Kapetanios, E., Tatar, D., dan Sacarea, C. (2013): Named Entity Recognition, *Natural Language Processing*, **8**(2), 309–322. https://doi.org/10.1201/b15472-19

Klinger, R. (2007): Classical Probabilistic Models and Conditional Random Fields, *Entropy*, diperoleh melalui situs internet: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.140.4527&rep=rep1

&type=pdf, **51**(December), 282–289.
L.Sumathy, K., dan Chidambaram, M. (2013): Text Mining: Concepts, Applications, Tools and Issues An Overview, *International Journal of Computer Applications*, **80**(4), 29–32. https://doi.org/10.5120/13851-1685
Muhammad, F., dan Khodra, M. L. (2015): Event information extraction from Indonesian tweets using conditional random field, *ICAICTA 2015 - 2015 International Conference on Advanced Informatics: Concepts, Theory and Applications*, 0–5. https://doi.org/10.1109/ICAICTA.2015.7335383
Munarko, Y, Sutrisno, M. S., Mahardika, W. A. I., Nuryasin, I., dan Azhar, Y. (2018): Named entity recognition model for Indonesian tweet using CRF classifier, *IOP Conference Series: Materials Science and Engineering PAPER*. https://doi.org/10.1088/1757-899X/403/1/012067
Munarko, Yuda, Malang, U. M., dan Munarko, Y. (2015): Ekstraksi Nama Lokasi Dari Tweets Informasi, *Seminar Teknologi dan Rekayasa (SENTRA)*, 978–979.
Mutawalli, L., Taufan, M., Zaen, A., Tantoni, A., dan Kunci, K. (2020): Pemodelan Resiko Bencana Banjir Dengan Menggunakan Algoritma Self-Organizing Map, *PROSIDING SEMINAR NASIONAL PLANOEARTH*, diperoleh melalui situs internet: https://journal.ummat.ac.id/index.php/PRPE/article/view/3982, **2**, 1–5.
Okazaki, N. (2007): a fast implementation of Conditional Random Fields, diperoleh melalui situs internet: http://www.chokkan.org/software/crfsuite/.
Owen, L. (2022): Indonesian Stopword Combined, diperoleh melalui situs internet: https://github.com/louisowen6/NLP_bahasa_resources/blob/master/combined_stop_words.txt.
Patil, N., Patil, A., dan Pawar, B. V. (2020): Named Entity Recognition using Conditional Random Fields, *Procedia Computer Science*, **167**(2019), 1181–1188. https://doi.org/10.1016/j.procs.2020.03.431
Riny Sulistyowati, Hari Agus Sujono, A. K. M. (2015): Sistem Pendeteksi Banjir Berbasis Sensor Ultrasonik Dan Mikrokontroler, *Seminar Nasional Sains dan Teknologi Terapan*, (January), 49–58.
Salsabila, N. A., Ardhito, Y., Ali, W., Septiandri, A., dan Jamal, A. (2018): Colloquial Indonesian Lexicon, *2018 International Conference on Asian Language Processing (IALP)*, 226–229.
Sastrawi · GitHub. (n.d.): , diperoleh 22 Juni 2022, melalui situs internet: https://github.com/sastrawi.
Sifataru (2019): SIFATARU - Gerbangkertosusila, , diperoleh 11 Maret 2021, melalui situs internet: https://sifataru.atrbpn.go.id/kawasan/Gerbangkertosusila.
Siska, B., Astuti, F., Firdausanti, N. A., dan Purnami, S. W. (2018): Model Evaluation for Logistic Regression and Support Vector Machines in Diabetes Problem, **1**(December).
Sun, P., Yang, X., Zhao, X., dan Wang, Z. (2019): An Overview of Named Entity Recognition, *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*, 273–278. https://doi.org/10.1109/IALP.2018.8629225
Sutton, C., dan McCallum, A. (2011): An introduction to conditional random fields, *Foundations and Trends in Machine Learning*, **4**(4), 267–373. https://doi.org/10.1561/2200000013
Utami, I., dan Marzuki, M. (2020): Analisis sistem informasi banjir berbasis media twitter, *Jurnal Fisika Unand*, diperoleh melalui situs internet: http://jfu.fmipa.unand.ac.id/index.php/jfu/article/view/454, **9**(1), 67–72.
Van Rossum, G., dan Muller, R. P. (2009): Introduction to Python Heavily based on presentations by Matt Huenerfauth (Penn State), diperoleh 1 Desember 2021melalui situs internet: http://www.python.org/doc/.
Wallach, H. M. (2004): ScholarlyCommons Conditional Random Fields : An Introduction Conditional Random Fields : An Introduction, (February).

Wijaya, D. T. (2021): IndoCollex : A Testbed for Morphological Transformation of Indonesian Colloquial Words, (2017), 3170–3183.

Ye, W., Li, B., Xie, R., Sheng, Z., Chen, L., dan Zhang, S. (2020): Exploiting entity BIO tag embeddings and multi-task learning for relation extraction with imbalanced data, *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 1351–1360. https://doi.org/10.18653/v1/p19-1130

Yudi Wibisono (2018): POS Tagger Bahasa Indonesia dengan Python – Blog Yudi Wibisono, diperoleh 22 Juni 2022, melalui situs internet: https://yudiwbs.wordpress.com/2018/02/20/pos-tagger-bahasa-indonesia-dengan-pytho/.