**Sistem Ekstraksi Informasi *Event* dari Twitter**

**Laporan Tugas Akhir**

**Disusun sebagai syarat kelulusan mata kuliah**

**IF4091/Tugas Akhir II dan Seminar**

**Oleh**

**Fawwaz Muhammad**

**NIM : 13511083**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO & INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**DESEMBER 2014**

**Sistem Ekstraksi Informasi *Event* dari Twitter**

**Laporan Tugas Akhir I**

**Oleh**

**Fawwaz Muhammad**

**NIM : 13511083**

**Program Studi Teknik Informatika**

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Bandung, 29 Desember 2014

Mengetahui,

Pembimbing,

Dr. Masayu Leylia Khodra, ST., MT.

NIP. 197604292008122001

DAFTAR ISI

[DAFTAR ISI iii](#_Toc416648791)

[DAFTAR GAMBAR v](#_Toc416648792)

[DAFTAR TABEL vi](#_Toc416648793)

[BAB I PENDAHULUAN 1](#_Toc416648794)

[I.1 Latar Belakang 1](#_Toc416648795)

[I.2 Rumusan Masalah 3](#_Toc416648796)

[I.3 Tujuan 4](#_Toc416648797)

[I.4 Batasan Masalah 4](#_Toc416648798)

[I.5 Metodologi 4](#_Toc416648799)

[I.6 Jadwal Pelaksanaan Tugas Akhir 6](#_Toc416648800)

[BAB II STUDI LITERATUR 7](#_Toc416648801)

[II.1 Twitter 7](#_Toc416648802)

[II.2 Ekstraksi Informasi 8](#_Toc416648803)

[II.2.1 Tahapan Ekstraksi Informasi 10](#_Toc416648804)

[II.2.2 Arsitektur Sistem Ekstraksi Informasi 18](#_Toc416648805)

[II.2.3 Evaluasi Sistem Ekstraksi Informasi 19](#_Toc416648806)

[II.3 Penelitian – Penelitian Terkait 20](#_Toc416648807)

[BAB III Ekstraksi informasi event 25](#_Toc416648808)

[III.1 Analisis Permasalahan 25](#_Toc416648809)

[*III.1.1* Analisis Informasi *event* pada *tweet* 25](#_Toc416648810)

[III.1.2 Ekstraksi Informasi 30](#_Toc416648811)

[III.2 Deskripsi Solusi 34](#_Toc416648812)

[III.2.1 Analisis Keterkaitan dengan Penelitian Lain 35](#_Toc416648813)

[III.2.2 Rancangan Arsitektur Sistem 35](#_Toc416648814)

[BAB IV PENGUJIAN DAN ANALISIS 46](#_Toc416648815)

[IV.1 Tujuan Pengujian 46](#_Toc416648816)

[IV.2 Deskripsi Data Uji 46](#_Toc416648817)

[IV.3 Skenario Pengujian 48](#_Toc416648818)

[IV.3.1 Skenario Pengujian Modul Filter 48](#_Toc416648819)

[IV.3.2 Skenario Pengujian Modul NER 48](#_Toc416648820)

[IV.4 Hasil Pengujian dan Analisis 49](#_Toc416648821)

[IV.4.1 Hasil Pengujian dan Analisis Modul Filter 49](#_Toc416648822)

[IV.4.2 Hasil Pengujian dan Analisis Modul NER 50](#_Toc416648823)

[DAFTAR PUSTAKA 51](#_Toc416648824)

DAFTAR GAMBAR

[Gambar I‑1 Contoh potongan rubrik dalam harian Media Indonesia 1](#_Toc416648836)

[Gambar I‑2 Salah satu contoh agregasi informasi *event* di Twitter 2](#_Toc416648837)

[Gambar II‑1 Contoh teks tak terstruktur mengenai pengumuman suatu seminar. 8](#_Toc416648838)

[Gambar II‑2 Contoh teks semi-terstruktur entitas sudah dipisahkan secara visual. 8](#_Toc416648839)

[Gambar II‑3 Contoh teks terstruktur. Setiap entitas memiliki kelas entitas. 9](#_Toc416648840)

[Gambar II‑4 Komponen-komponen dalam sistem ekstraksi informasi (Feldman & Sanger, 2007). 18](#_Toc416648841)

[Gambar III‑1 Contoh *tweet* noise milik akun twitter @Bdg*Event*. 26](#_Toc416648842)

[Gambar III‑2 Contoh *tweet* dengan informasi tak lengkap 28](#_Toc416648843)

[Gambar III‑3 Contoh *tweet* yang hanya mengandung kombinasi nama entitas 29](#_Toc416648844)

[Gambar III‑4 Contoh *tweet* yang hanya menyebtukan entitas dengan urutan berbeda 30](#_Toc416648845)

[Gambar III‑5 Arsitektur sistem ekstraksi informasi event bahasa indonesia 37](#_Toc416648846)

DAFTAR TABEL

[Tabel II‑1 Contoh *Named Entity* dalam beberapa kalimat 10](#_Toc408161160)

[Tabel II‑2 Contoh ungkapan leksikal yang menandai ekspresi temporal 15](#_Toc408161161)

[Tabel II‑3 Contoh *event* dalam berbagai kelas kata 16](#_Toc408161162)

[Tabel II‑4 Ringkasan penelitian terkait 22](#_Toc408161163)

[Tabel III‑1 Label-label yang digunakan dalam NER 32](#_Toc408161164)

[Tabel III‑2 Daftar akun twitter dan *hashtag* yang dijadikan *training dataset*. **Error! Bookmark not defined.**](#_Toc408161165)

# PENDAHULUAN

## Latar Belakang

Saat ini agregasi informasi mengenai *event* masih dilakukan secara manual. Hal ini bisa dilihat dari beberapa website yang berusaha menghimpun publikasi-publikasi mengenai *event* seperti e*event*.com, *event*brite.com, info*event*.com. Tidak hanya dilakukan dengan *platform* website, agregasi informasi *event* juga dilakukan oleh media cetak. Harian Pikiran Rakyat dan Media Indonesia (Gambar I‑1) , bahkan memiliki rubrik khusus satu halaman yang merangkum *events* apa saja yang akan dilaksanakan dalam seminggu ke depan.



Gambar I‑1 Contoh potongan rubrik dalam harian Media Indonesia

Media sosial saat ini sudah dianggap sebagai salah satu sumber informasi. Para pengguna media sosial membagikan informasi seputar opini dan apa saja yang menarik bagi mereka. Salah satu hal yang sering dibagikan di sosial media adalah informasi seputar *event*. Berdasarkan laporan yang dikeluarkan oleh Amiando, 84% *event organizer* menggunakan Facebook untuk mempromosikan *event*, 61% *event organizer* menggunakan Twitter dan 42% *event organizer* menggunakan Youtube (Spless & Alphen-Schrade, 2013).

Meskipun Twitter menempati urutan kedua dalam media promosi *event*, tetapi Twitter memiliki pengguna aktif cukup tinggi di Indonesia. Berdasarkan penelitan yang dilakukan oleh TechinAsia, Indonesia berada pada posisi ke-5 dalam urutan jumlah pengguna Twitter terbanyak sedunia (Lukman, 2013). Selain itu, Twitter menyediakan API yang lebih mudah dalam hal pencarian *tweet*. Alasan-alasan inilah yang menunjukkan bahwa Twitter memiliki potensi menjadi sumber media informasi suatu *event*.



Gambar I‑2 Salah satu contoh agregasi informasi *event* di Twitter

Twitter dapat dianggap sebagai salah satu media informasi suatu *event*. Hal ini dibuktikan dengan bermunculanya akun-akun Twitter yang berusaha melakukan agregasi informasi *event* seperti @infobdg*event*, @acarajakarta (Gambar I‑2) dan sebagainya. Sayangnya agregasi pada akun Twitter tersebut masih dilakukan secara manual. Selain itu, struktur informasi pada akun Twitter tersebut tidak tersusun secara rapih. Dibutuhkan suatu sistem yang otomatis yang mengagregasi informasi *event*.

Salah satu tahapan dalam agregasi informasi adalah ekstraksi informasi. Ada beberapa penelitian yang mencoba melakukan ekstraksi *event* dari Twitter. Sebagian peneliti mencoba melakukan deteksi *event* di Twitter seperti deteksi gempa bumi dari Twitter (Sakaki dkk.,2010), penelitian tentang deteksi tindakan kriminal dan bencana (Li dkk., 2012), sebagian lagi melakukan penelitian seputar ekstraksi informasi *event* dari Twitter. Sayangnya penelitian-penelitian tersebut dilakukan dalam *tweet* berbahasa Inggris. Sejauh ini ekstraksi informasi dari *tweet* berbahasa Indonesia masih dimanfaatkan untuk pencarian kemacetan lalu lintas (Hasby & Khodra, 2013; Endarnato dkk., 2011). Belum ada penelitian mengenai ekstraksi informasi *event* dari Twitter dalam bahasa Indonesia. Berangkat dari fakta ini, perlu dilakukan penelitian mengenai ekstraksi informasi *event* dari Twitter berbahasa Indonesia.

Informasi yang terdapat pada Twitter cenderung tidak terstruktur karena pembatasan panjang huruf dalam satu *tweet*. Metode ekstraksi informasi di Twitter menggunakan aturan (*rule-based*) cenderung tidak efektif. Selain itu, informasi di Twitter memiliki potensi duplikasi informasi. Oleh karena itu, dibutuhkan suatu metode yang mampu melakukan ekstraksi informasi dari sumber tak terstruktur serta mampu melakukan resolusi terhadap sekelompok *event* yang merujuk pada *event* yang sama.

## Rumusan Masalah

Berdasarkan latar belakang masalah tersebut, masalah yang akan dipecahkan dalam tugas Akhir ini adalah konfigurasi apa saja yang harus diadaptasi dari metode ekstraksi *event* pada penelitan sebelumnya untuk domain berbahasa Indonesia sehingga memberikan kinerja yang cukup baik.

## Tujuan

Tujuan dari Tugas Akhir ini adalah membuat sistem yang dapat melakukan ekstraksi informasi *event* dari Twitter dan menentukan konfigurasi metode ekstraksi *event* yang memuaskan untuk domain bahasa Indonesia. Untuk mencapai tujuan tersebut terdapat sub-tujuan yang harus dicapai yaitu:

1. Membangun sistem yang mampu melakukan ekstraksi informasi *event* dari Twitter menggunakan pendekatan berbasis statistik.
2. Menentukan konfigurasi yang mampu menghasilkan kinerja yang memuaskan

## Batasan Masalah

Batasan masalah yang diperhatikan dalam tugas akhir ini adalah sebagai berikut:

1. Sistem yang dibangun hanya menerima input *tweet* pengguna twitter di Indonesia.
2. *Event* yang diekstrak adalah *event* yang spesifik dan direncanakan oleh manusia. *Event* tersebut berupa *event* lokal yang dibatasi oleh tempat dan waktu tertentu. *Event* global seperti Pemilu Presiden RI atau *event* yang tidak direncanakan manusia seperti bencana alam diluar ruang lingkup tugas akhir.

## Metodologi

Metodologi yang digunakan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Analisis Persoalan

Persoalan ekstraksi informasi *event* dari Twitter adalah hal yang unik. Dibutuhkan analisis khusus untuk mencari hal-hal yang spesifik dalam persoalan ini. Output dari tahap ini adalah alternatif-alternatif pendekatan yang dapat dilakukan untuk menyelesaikan persoalan.

1. Desain Solusi

Setelah melakukan analisis, dipilih salah satu alternatif yang paling dapat dilakukan. Alternatif ini kemudian diperjelas pada tahap desain solusi.

1. Pembangunan Prototipe Sistem

Setelah langkah yang harus diambil sudah rinci, tahap selanjutnya adalah membangun prototipe sistem. Pada tahap ini desain yang sudah dibuat diimplementasikan.

1. Konstruksi *Dataset*

*Dataset* dibutuhkan untuk melatih prototipe. Pada tahapan ini dilakukan *labelling* terhadap sejumlah data untuk proses pelatihan.

1. Eksperimen

Eksperimen dilakukan untuk melatih sistem. Pada tahap eksperimen dilakukan beberapa kombinasi parameter untuk mendapatkan hasil yang terbaik.

1. Pengujian

Pengujian dilakukan untuk mengukur kinerja dari prototipe sistem yang sudah dibuat. Sistem diimplementasikan dalam lingkungan riil.

1. Analisis Akhir

Kinerja sistem setelah melalui tahap pengujian kemudian dibandingkan dengan kinerja yang diharapkan. Jika dibutuhkan, dilakukan analisis akhir seandainya hasil yang didapat berbeda dengan harapan.

## Jadwal Pelaksanaan Tugas Akhir

Berikut adalah rencana jadwal pelaksanaan tugas akhir:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task | Bulan | | | | | | | | |
| 10 | 11 | 12 | 1 | 2 | 3 | 4 | 5 | 6 |
| Analisis Persoalan |  |  |  |  |  |  |  |  |  |
| Desain Solusi |  |  |  |  |  |  |  |  |  |
| Pembangunan Prototipe |  |  |  |  |  |  |  |  |  |
| Konstruksi *Dataset* |  |  |  |  |  |  |  |  |  |
| Eksperimen |  |  |  |  |  |  |  |  |  |
| Pengujian |  |  |  |  |  |  |  |  |  |
| Analisis Akhir |  |  |  |  |  |  |  |  |  |

# STUDI LITERATUR

## Twitter

Twitter adalah salah satu media sosial. Berbeda dengan media sosial Facebook, bentuk interaksi yang terbentuk di Twitter dapat berupa interaksi asimetri. Artinya satu *user* dapat melakukan *follow* terhadap *user* lainya tetapi belum tentu *user* yang di-*follow* melakukan *follow* balik. *Tweets* yang dimiliki oleh akun tanpa proteksi dapat diakses publik. Tidak perlu melakukan *follow* terhadap akun tersebut. Hal inilah yang membuat *crawling* informasi dari Twitter jauh lebih mudah dibandingkan media sosial lainya.

Ada beberapa istilah spesifik yang terdapat di dalam Twitter. *Mention* adalah suatu kata didahului dengan token ‘@’ diikuti dengan nama pengguna Twitter. *Tweet* yang mengandung *mention* akan muncul pada bagian ‘*Mention*’ milik pengguna yang bersesuaian. *Hashtag* adalah suatu kata yang didahului dengan token ‘#’. *Hashtag* digunakan untuk mengelompokan beberapa *tweet* dalam satu kelompok sehingga memudahkan pencarian ulang terhadap *tweet* itu di lain waktu.

Twitter memiliki kebijakan khusus dalam hal panjang karakter sebuah *tweet*. Satu *tweet* di twitter maksimal memiliki panjang 140 karakter. Twitter menggunakan *character encoding* UTF-8. Dalam hal perhitungan karakter, Twitter menggunakan *Normalization Form C*.

Twitter menyediakan akses terhadap *tweet* melalui API dalam bentuk REST. Twitter melakukan transfer data dalam bentuk JSON. Twitter menyediakan *metadata* dalam objek *tweet* seperti informasi *geotag*, media (dapat berupa gambar atau video) dan URL (Twitter,2014).

## Ekstraksi Informasi

Ekstraksi informasi adalah usaha untuk mencari informasi terstruktur dari suatu teks tak terstruktur atau semi-terstruktur (Aggarwal & Zhai, 2012). Informasi tak terstruktur didefinisikan sebagai teks yang maknanya bergantung seluruhnya terhadap analisis lingguistik dan pemahaman bahasa alami. Teks semi-terstruktur didefinisikan sebagai teks yang dapat diinterpretasikan berdasarkan layout fisik yang muncul pada dokumen. Adapun informasi terstruktur didefinisikan sebagai teks yang memiliki semantik informasi yang telah ditentukan oleh suatu organisasi (Herbich & Graepel, 2010). Contohnya dapat dilihat dalam Gambar II‑1 , Gambar II‑2 dan Gambar II‑3.

|  |
| --- |
| Professor John Skvoretz, U. of South Carolina, Columbia, will present a seminar entitled “Embedded Commitment,” on Thursday, May 4th from 4-5.30 in PH 223D |

Gambar II‑1 Contoh teks tak terstruktur mengenai pengumuman suatu seminar.

|  |
| --- |
| Laura Petitte Departement of Physocolgy  McGill University  Thursday, May 4, 1995 12:00 pm Baker Hall 355 |

Gambar II‑2 Contoh teks semi-terstruktur entitas sudah dipisahkan secara visual.

|  |
| --- |
| Name : Dr. Jeffrey D. Hermes Affiliation : Department of AutoImmune Diseases Research & Biophysical Chemistry Merck Research Laboratories  Title : “MHC Class II: A Target for Specific Immunomodulation of the Immune Response” Host/e-mail : Robert Murphy, [murph@a.crf.cmu.edu](mailto:murph@a.crf.cmu.edu) Date : Wednesday, May 3, 1995 Time : 3:30 p.m. Place : Mellon Institute Conference Room Sponsor: MERCK RESEARCH LABORATORIES |

Gambar II‑3 Contoh teks terstruktur. Setiap entitas memiliki kelas entitas.

Ekstraksi informasi adalah proses memindai teks untuk suatu informasi yang relevan pada suatu topik termasuk melakukan ekstraksi entitas dan relasi. Salah satu aplikasi ekstraksi informasi yang cukup sulit adalah ekstraksi informasi mengenai *event* (siapa melakukan apa terhadap apa, kapan, dan dimana). Ekstraksi informasi tidak hanya sekedar mencari potongan kata dari suatu teks, melainkan proses yang membutuhkan analisis yang mendalam. (Herbich & Graepel, 2010).

Ekstraksi informasi bisa gagal pada teks yang membutuhkan pemahaman yang mendalam. Misalnya saat informasi pada teks yang harus dibaca seluruhnya untuk memahami maksud dari penulis (Herbich & Graepel, 2010) Ekstraksi informasi hanya berguna jika dilakukan pada teks yang mengandung kriteria :

1. Informasi yang ingin diekstrak ditentukan secara eksplisit, tanpa membutuhkan proses inferensi.
2. Terdapat sejumlah *template* yang sudah cukup mampu meringkas bagian yang berhubungan pada suatu dokumen.
3. Seluruh informasi yang dibutuhkan dinyatakan relatif lokal terdapat dalam dokumen. (Feldman & Sanger, 2007)

Dalam proses ekstraksi informasi, setiap dokumen diolah untuk mendapatkan entitas-entitas, atribut entitas dan hubungan antar entitas yang ada di dalam dokumen tersebut. Hubungan antar entitas akan membentuk fakta atau *event* yang melibatkan beberapa entitas. Fakta adalah hubungan yang cenderung statis dan biasanya tidak berubah adapun *event* lebih bersifat dinamis dan memiliki keterkaitan terhadap waktu (Feldman & Sanger, 2007).

### Tahapan Ekstraksi Informasi

Menurut Jurafsky (Jurafsky & Martin, 2008), dalam proses ekstraksi informasi terdapat beberapa tahapan:

1. *Named Entity Recognition*
2. *Relation Detection and Classification*
3. *Event Detection and Classification*
4. *Temporal Expression Analysis*
5. *Template Filling*

#### *Named Entity Recognition (NER)*

*Named Entity* didefinisikan sebagai unit informasi yang dapat diacu oleh suatu nama konkret (Jurafsky & Martin, 2008). *Named Entity Recognition* adalah proses identifikasi seluruh nama entitas yang mungkin ada pada teks. Proses *Named Entity Recognition* perlu dilakukan karena mustahil seluruh kemungkinan nama suatu entitas disimpan dalam sebuah kamus.

Tabel II‑1 Contoh *Named Entity* dalam beberapa kalimat

|  |  |
| --- | --- |
| Jenis Entitas | Contoh kalimat |
| Nama Orang | ***Udin*** pergi ke pasar; ***Ahmad*** belajar di sekolah. |
| Nama Organisasi | Udin anggota ***APB*** (Asosiasi Pedagang Bandung); Ahmad anggota ***OSIS*** |
| Waktu | Udin pergi ke pasar ***setiap hari pukul 1 dini hari***; Ahmad berangkat ke sekolah ***jam 7***. |
| Nama Tempat | Dia pergi ke ***Pasar Wastukencana***; Sekolah Ahmad ada di ***Jalan Belitung***. |

Dalam Tabel II‑1 dapat dilihat beberapa contoh *Named Entity* dalam beberapa kalimat. Kata ‘Udin’ dan ‘Ahmad’ mengacu pada suatu nama orang. Kata ‘APB’ dan ‘OSIS’ mengacu pada nama organisasi. Frase ‘setiap hari pukul 1 dini hari’ dan kata ‘jam 7’ mengacu pada waktu. Kata ‘Pasar Wastukencana’ dan ‘Jalan Belitung’ mengacu pada nama tempat. Kata-kata tersebut (nama orang, nama organisasi, waktu atau nama tempat) tidak mungkin disimpan dalam suatu kamus karena ada tidak terhingga kemungkinan penamaan entitas-entitas tersebut.

##### Pendekatan Klasik

Sistem ekstraksi informasi yang menggunakan pendekatan klasik memanfaatkan aturan-aturan spesifik (pola) yang didefinisikan secara manual atau secara otomatis dipelajari. Teks kemudian dibandingkan dengan aturan-aturan tersebut. Sistem akan mengenali *Named Entity* jika ada aturan yang cocok dengan sebagaian teks.

Aturan terdiri dari pola kata dan suatu aksi. Suatu pola biasanya berupa *Regular Expression* yang didefnisikan berdasarkan sejumlah fitur dari *token*. Jika pola ini cocok dengan suatu urutan *token*, maka suatu aksi spesifik yang sudah didefinisikan akan dibangkitkan dan dilaksanakan.

Akan sangat mungkin suatu urutan *token* memenuhi beberapa aturan sekaligus dalam satu waktu. Untuk menangani konflik ini, maka perlu aturan tambahan lain yang mengatur prioritas bagaimana suatu aksi aturan dilaksanakan. Proses pembuatan aturan-aturan ini membutuhkan pakar (Aggarwal & Zhai, 2012).

Sebagai contoh, misalkan pakar membuat sebuah aturan *regular expression* berupa

PT {Kata diawali huruf kapital} -> Perusahaan

Aturan tersebut dapat diartikan sebagai, untuk semua kata yang diawali dengan kata PT, kemudian diikuti dengan suatu nama dengan huruf kapital pada huruf pertamanya, maka nama itu akan dianggap sebagai nama perusahaan. Aturan tersebut akan memenuhi kata-kata seperti PT Inti, PT Telkom, PT Divusi. Dari aturan ini akan didapatkan potongan kata yang mewakili entitas nama suatu perusahaan yaitu Inti, Telkom, Divusi.

##### Pendekatan Statistik

Dalam pendekatan statistik persoalan *Named Entity Recognition* dapat dianggap sebagai persoalan klasifikasi *sequence labelling* (Hasby & Khodra, 2013). Persoalan diformulasikan sebagai berikut. Diberikan suatu urutan observasi ***x****=*(*x1, x2,.., xn-1, xn*). Setiap nilai observasi dinyatakan dalam vektor fitur. Tujuanya adalah bagaimana kita memberikan label yi untuk setiap observasi xi. Akan tetapi label yi tidak hanya bergantung kepada xi melainkan juga bergantung pada observasi label lain yang bertetanggaan pada urutan tersebut. Dalam konteks *Named Entity Recognition*, nilai observasi dapat dianggap sebagai setiap kata pada suatu kalimat. Label kelas harus mampu mengindikasikan tipe dan batas *Named Entity* dalam suatu kalimat. Notasi BIO digunakan untuk memenuhi kebutuhan ini. Dengan notasi ini *token* yang diberi label B-T dapat diartikan sebagai *token* yang awal dari *Named Entity* dengan tipe T. *Token* yang diberi label I-T diartikan sebagai *token* yang termsauk dalam *Named Entity* tipe T namun bukan *token* pertama. Adapun *token* yang diberi label O-T adalah *token* yang tidak berkaitan dalam *named entity*. (Aggarwal & Zhai, 2012). Pendekatan berbasis statistik ini memiliki beberapa algoritma diantaranya *Maximum Entropy*, HMM (*Hidden Markov Model*), CRF(*Conditional Random Field*), SVM (*Support Vector Machine*), k-NN(K-*Nearest Neighbour*) dan lain-lain (Hasby & Khodra, 2013). Diantara algoritma tersebut, yang paling umum digunakan dalam sequence labelling adalah HMM dan CRF. Berikut penjelasan mengenai kedua algoritma tersebut.

###### Algoritma Hidden Markov Model

Hidden Markov Model termasuk model generatif. Model ini didasari pada Markov *Chain*. Markov *chain* adalah bentuk perluasan dari *Finite State Automata* (FSA). Markov chain menambahkan probabilitas transisi antara satu *state* ke *state* lainya (Jurafsky, 2008) . Markov *chain* didefinisikan secara formal dengan 3 buah komponen :

1. Q = q1, q2, ..., qn yaitu N buah jumlah *state*.
2. A = a01, a02, ... an1, ... *ann*yaitu matriks probabilitas transisi A. Setiap elemen matriks A (aij) merepresentasikan probabilitas transisi dari state ke-i untuk berpindah ke state ke-j. Untuk setiap *state*, jumlah probabilitas yang keluar dari *state* ke-i = 1.
3. Q0, Qf yaitu state khusus representasi *state* awal dan *state* akhir yang tidak diasosiasikan dengan observasi apapun.

Markov *chain* menyederhanakan persoalan dengan asumsi, bahwa probabilitas sebuah *state* hanya bergantung dari probabilitas *state* sebelumnya. Dalam Hidden Markov Model, terdapat komponen tambahan untuk Markov *chain* yaitu,

1. O = o1,o2,... ot yaitu urutan t buah observasi, masing-masing diambil dari himpunan variabel V= {v1,v2..vv}
2. B = bi(ot) yaitu probabilitas emisi. Menyatakan seberapa besar probabilitas ot akan dihasilkan pada *state* ke-i.

Selain penambahan komponen, Hidden Markov Model menambahkan asumsi bahwa probabilitas emisi independen terhadap faktor probabilitas yang lainya. Perhitungan probabilitas hidden markov model mengikuti perhitungan *naive bayes*.

Prinsip kerja Hidden Markov Model adalah menghitung probabilitas gabungan kejadian diamati input O dan menghasilkan output V . Dalam notasi formal, Hidden Markov Model memaksimalkan P(**V**,**O**).

Hidden Markov Model menggunakan algoritma Baum-Welch untuk melakukan pembelajaran. Algoritma ini mencoba melakukan penyesuaian agar memaksimalkan model yang didapat berdasarkan rangkaian pengamatan. Sementara dalam proses klasifikasi, Hidden Markov Model menggunakan algoritma Viterbi, salah satu jenis dari algoritma pemrograman dinamis. Algoritma ini mencari rangkaian observasi yang memberikan nilai P(**V**,**O**) paling maksimum.

###### Algoritma Conditional Random Field (CRF)

Conditional Random Field termasuk model diskriminatif. Model ini bentuk perluasan dari regresi logistik. Mirip dengan HMM, CRF memiliki sejumlah state. Namun CRF sedikit berbeda dengan model HMM, model ini berusaha menghampiri secara langsung probablitas P(**V**|**O**).

#### *Relation Extraction*

Setelah melakukan ekstraksi entitas, langkah selanjutnya yang harus dilakukan adalah menentukan hubungan antar entitas. *Relation Extraction* dilakukan untuk mendapatkan hubungan semantik antar entitas. Sebagai contoh dalam kalimat

‘Ahmad anggota OSIS’

Dapat diekstrak hubungan semantik yaitu:

Anggota(Ahmad,OSIS)

Terdapat beberapa pendekatan dalam melakukan *Relation Extractoin* : Pertama menggunakan pendekatan klasik; Kedua menggunakan pendekatan Statistik. (Aggarwal & Zhai, 2012). Pada pendekatan klasik, *Relation Extraction* dilakukan dengan membentuk aturan tertentu(*rule-based*). Adapun dalam pendekatan statistik digunakan algoritma-algoritma *learning* umum seperti SVM. Fitur-fitur yang digunakan antara lain fitur entitas, fitur konteks leksikal, fitur konteks sintaksis dan *background knowledge*.

#### *Temporal Resolution*

Ada beberapa jenis ungkapan terkait dengan waktu (Jurafsky & Martin, 2008).

1. Batas – batas waktu

Ungkapan mengenai batas-batas waktu dapat dibagi menjadi dua kategori:

* Batas waktu absolut, batas waktu ini mutlak, tidak tergantung pada waktu saat diucapkan. Contohnya dalam kalimat “Ahmad pergi ke sekolah jam 7 pagi”.
* Batas waktu relatif, batas waktu ini tergantung pada waktu saat diucapkan. Contohnya dalam kalimat “Ahmad akan pulang dari sekolah 1 jam lagi”.

1. Rentang Waktu

Contohnya dalam kalimat “Ahmad belajar di sekolah selama 8 jam”

Secara sintaksis terdapat beberapa leksikal-leksikal yang menunjukkan suatu ekspresi temporal. Leksikal-leksikal tersebut dapat berupa Noun, Proper Noun, Adjective atau Adverb) sebagaimana yang terlihat dalam Tabel II‑2:

Tabel II‑2 Contoh ungkapan leksikal yang menandai ekspresi temporal

|  |  |
| --- | --- |
| Kategori | Contoh |
| *Noun* | Pagi, Siang, Sore, Malam |
| *Proper Noun* | Senin, Selasa, Januari, Bulan Ramadhan, |
| *Adjective* | Terakhir, Tahunan, |
| *Adverb* | Setiap pekan, Setiap hari |

Untuk mengenali ungkapan temporal dapat dilakukan melalui beberapa metode yaitu (Jurafsky & Martin, 2008):

1. Pendekatan *rule-based*

Pendekatan ini menggunakan suatu FSA(*Finite State Automata*) bertingkat untuk mengenali pola ekspresi temporal yang kompleks.

1. Pendekatan statistik

Pendekatan statistik menganggap ekspresi temporal adalah persoalan *sequence labelling*. Pendekatan statistik menggunakan tag B-I-O (Begin, In, Outside) untuk menentukan ekspresi temporal.

#### *Event Detection and Classification*

Task ini memiliki tujuan untuk mencari kemunculan seluruh *event* dalam dokumen. Kebanyakan *event* dikaitkan dengan kata kerja. Tapi terkadang tidak semua kata kerja menunjukkan *event*. Ada juga beberapa *event* yang dikaitkan dengan frase kata benda seperti yang terlihat dalam Tabel II‑3.

Tabel II‑3 Contoh *event* dalam berbagai kelas kata

|  |  |
| --- | --- |
| Kelas Kata | Contoh Kalimat |
| Kata Kerja | Bunga Citra Lestari ***menyanyikan*** lagu di Sabuga**.** |
| Kata Benda | ***Konser*** Bunga Citra Lestari ada di Sabuga. |

#### Task ini dapat digunakan untuk membuat urutan parsial (*partial order*) dari *event* dan ungkapan temporal yang disebutkan di dalam teks. Untuk melakukan pencarian keterurutan suatu *event*, digunakan metode ekstraksi relasi (Jurafsky & Manning, 2008).

#### *Template Filling*

Pada tahapan *template filling*, setiap entitas yang sudah diekstrak dipetakan ke *slot* template yang sudah ditentukan sebelumnya. MUC menamakan task ini sebagai *Scenario Template Task*. Menurut Li dkk. (2014), ada 4 jenis *slot* yang terdapat dalam template:

1. *Set Fill*

Dalam kategori *set fill*, *slot* diisi berupa seleksi dari sebuah daftar yang sudah didefinisikan sebelumnya. Daftar tersebut harus memenuhi aturan sesuai dengan domain sistem ekstraksi informasi. Sebagai contoh, dalam slot template `jenis kelamin` isi slotnya hanya dapat berupa salah satu dari `Pria` atau `Wanita`.

1. *String Fill*

Dalam kategori *String Fill*, *slot* diisi dengan salinan yang sama persis seperti yang terdapat dalam teks yang diekstrak informasinya. Sebagai contoh, misal terdapat suatu teks:

“Bunga Citra Lestari akan tampil di Jakarta Night Festival”

Untuk suatu template dengan slot ‘nama penyanyi’ akan diisi sama persis dengan entitas yang ditemukan dalam teks. Pada kasus ini slot tersebut akan diisi dengan ‘Bunga Citra Lestari’

1. *Normalized Fill*

Dalam kategori *Normalized Fill, slot* diisi dengan bentuk yang sudah dinormalkan dari teks yang ditemukan. Bentuk normal ini sudah ditentukan sesuai dengan kebutuhan sistem. Sebagai contoh misal terdapat suatu teks :

“BCL akan tampil pada malam tahun baru, 31/12/14”

Untuk suatu *template* dengan *slot* ‘tanggal’ mungkin akan diisi dengan ‘31 Desember 2014’

1. *Index Fill*

Dalam kategori *Index Fill, slot* diisi dengan suatu *pointer* terhadap himpunan suatu objek yang sudah didefinisikan. Misal tedapat suatu teks:

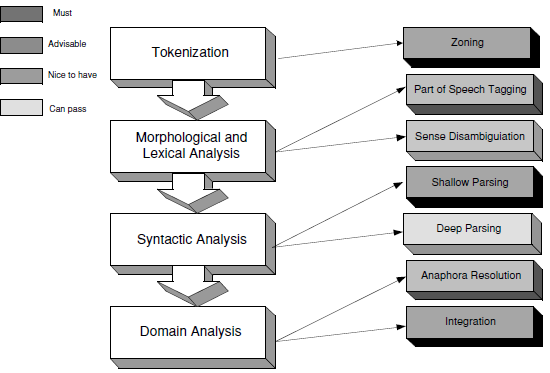
“Mahasiswa tersebut bernama Arief Rahman”

Untuk suatu template dengan slot ‘nim’, slot tersebut akan diisi dengan ‘13511020’.

Menurut Jurafsky & Martin (2008), pendekatan paling efektif untuk melakukan task template filling adalah pendekatan sequence labelling. Ada beberapa pendekatan untuk melakukan hal ini. Pertama, dengan melatih beberapa *classifier* untuk mengenali setiap slot secara terpisah. Kedua, dengan menggunakan satu buah *classifier* besar, misal HMM(*Hidden Markov Model*), dan memberikan label untuk setiap slot dari classifier ini. Meskipun demikian, dalam beberapa kasus bentuk *template filling* menggunakan *rule-based* sudah memberikan hasil yang sangat baik.

### Arsitektur Sistem Ekstraksi Informasi

Secara umum arsitektur sistem ektraksi informasi minimal 4 komponen seperti yang terlihat dalam Gambar II‑4 (Feldman & Sanger, 2007):



Gambar II‑4 Komponen-komponen dalam sistem ekstraksi informasi (Feldman & Sanger, 2007).

1. Tokenisasi

Pada tahap ini, input teks dokumen dipecah menjadi unit atomis terkecil. Biasanya unit tersebut berupa kata-kata atau kalimat atau paragraf.

1. Analisis Morfologi dan Leksikal

Modul ini bertanggung jawab tehadap aktivitas-aktivitas seperti POS *Tag*, atau pembentukan frase kata benda / farse kata kerja (*noun phrase* atau *verb phrase*) dan proses memutuskan makna kata yang mengandung ambiguitas.

1. Analisis Sintaktik

Komponen ini bertanggung jawab terhadap pembentukan hubungan antar bagian-bagian dari kalimat. Hal ini dilakukan dengan menggunakan *shallow parsing* atau *deep parsing*.

1. Analisis domain

Komponen ini akan menggabungkan semua informasi yang berhasil dikumpulkan dari komponen-komponen sebelumnya dan membentuk kerangka utuh yang menjelaskan hubungan antar entitas.

### Evaluasi Sistem Ekstraksi Informasi

Ada banyak parameter yang dapat digunakan untuk melakukan evaluasi kinerja sistem ekstraksi informasi. ACE (*Automatic Content Extraction*) memiliki parameter yang dikembangkan sendiri untuk mengukur kinerja sistem ekstraksi informasi. Selain ACE, TREC (*Text Retrieval Conference*) juga mengembangkan parameter evaluasi sistem ekstraksi informasi. Kebanyakan mengacu pada MUC (*Message Understanding Conference)* (Buckley & Voorhees, 2000).

MUC menetepakan beberapa parameter untuk mengevaluasi sistem ekstraksi informasi :

* + 1. *Recall*

*Recall* pada suatu model didefinisikan sebagai perbandingan antara jumlah data yang berhasil diklasifikasikan dengan benar dari suatu sumber data dengan jumlah data yang benar pada sumber tersebut

* + 1. *Precision*

*Precission* suatu model didefinisikan sebagai perbandingan antara jumlah data yang berhasil diklasifikasikan dengan benar dengan jumlah data yang diambil dari sumber data.

* + 1. *F-Measure*

Dalam mengukur kinerja sistem ekstraksi informasi yang berada pada lingkungan *imbalance* *dataset*, pengukuran kinerja menggunakan recall dan precission tidak tepat. Oleh karena itu ditetapkan parameter F-Measure yang didefinisikan sebagai:

|  |  |  |
| --- | --- | --- |
|  |  | (2.1) |

## Penelitian – Penelitian Terkait

Terdapat beberapa penelitian terkait ekstraksi informasi *event* di Twitter. Penelitian tersebut terbagi dalam dua kategori: Pertama, deteksi kemungkinan keberadaan suatu *event*. Kedua, ekstraksi komponen informasi *event* dari *tweet*.

Penelitian-penelitian seputar deteksi *event* dari *tweet* sebagianya melakukan ekstraksi *event* pada kejadian-kejadian yang tidak diduga misalnya kejadian gempa (Sakaki dkk., 2010), tindak kriminal dan bencana (Li dkk., 2012). Sebagian lagi melakukan penelitian pada kejadian yang memang direncanakan seperti *event* pertandingan olahraga (Sugitani dkk., 2013; Boettcher & Lee, 2012; Cui dkk., 2012).

Sugitani dkk. (2013) menggunakan algoritma *Cluster*ing untuk mendeteksi *event* dengan memanfaatkan fitur metadata *tweet* berupa koordinat lintang dan bujur dan waktu, saat *tweet* tersebut dibuat. *Cluster* yang didefinisikan dibatasi oleh suatu batas radius tertentu,sejumlah orang yang melakukan *tweet* pada batas radius tersebut, dan rentang waktu tertentu. Boettcher & Lee (2012), menambahkan sistem deteksi *event* yang dilakukan oleh Sugitani. *Cluster* yang dibentuk oleh Boettcher & Lee (2012) tidak ditentukan oleh suatu konstanta batas radius area tertentu. *Cluster* tersebut dibentuk dari kesamaan leksikal dalam daftar suatu daftar kamus kategori *event*. Adapun Cui dkk. (2012), melakukan deteksi *event* berdasarkan *hashtag* saja.

Penelitian-penelitan seputar ekstraksi informasi *event* dari Twitter tidak terlalu banyak. Ritter dkk. (2012) melakukan ekstraksi informasi acara, klasifikasi jenis acara, dan melakukan ranking *event* yang cukup penting. Ritter dkk. menggunakan algoritma CRF untuk menentukan komponen-komponen pembentuk *event* seperti entitas yang terlibat, frase *event* dan tanggal *event*. Ritter dkk. menggunakan fitur konteks *tweet*, kamus, fitur ortografis, POS Tag khusus yang diadaptasi untuk domain Twitter, dan kamus terminology *event* yang dikumpulkan dari WordNet. Ritter menggunakan TempEx untuk melakukan resolusi frase temporal dan LinkLDA untuk mengkategorikan *event* yang diekstrak. Terakhir, urutan penting tidaknya suatu acara diurtukan berdasarkan perhitungan G2 yang melibatkan entitas yang terlibat dan tanggal *event*.

Penelitian Ritter dkk. kemudian dikembangkan oleh Khurdiya dkk. (Khurdiya dkk., 2012). Khurdiya dkk. menambahkan modul *tweet* indexer berbasis SOLR[[1]](#footnote-1) untuk mengelompokan *tweet*-*tweet*, modul *event* resolution dan *event* compiler. Khurdiya dkk. menggunakan CRF untuk mendapatkan komponen-komponen pembentuk *event* berupa subjek, aksi, objek, waktu, lokasi dan konteks dari *event*. Khurdiya dkk. memanfaatkan fitur berupa fitur kata yang sudah dilakukan lemmatisasi, fitur orthogonal, fitur spesifik Twitter (*hashtag*, *mention*, *retweet*), POS (*Part Of Speech*) untuk 5 kata yang bertetanggaan, dan tag *Named Entity* kata yang bersebelahan. Kemudian modul *event resolution* akan memanfaatkan entitas-entitas yang sudah diekstrak menggunakan algoritma CRF kembali untuk menghasilkan suatu kalimat lengkap nama *event*. Modul *event resolution* milik Khurdiya dkk. akan membandingkan struktur kalimat lengkap hasil proses CRF terakhir. Jika ditemukan kemiripan, modul *event resolution* Khurdiya dkk. akan menganggap *events* tersebut sebagai satu buah *event* tunggal. Dengan demikian didapatkan hasil ekstraksi *event* yang unik dan tidak mengandung duplikasi *event*.

Selain penelitian-penelitian tentang ekstraksi informasi *event*, ada juga penelitian-penelitian tentang ekstraksi informasi dari Twitter untuk penerapan yang berbeda. Sakaki dkk. (2012) melakukan ekstraksi informasi kemacetan lalulintas menggunakan dua modul: Pertama modul untuk mendeteksi *tweet* yang berpotensi mengandung informasi kemacetan lalu lintas; Kedua, modul untuk melakukan ekstraksi informasi lokasi kemacetan. Sakaki dkk. menggunakan SVM untuk memisahkan *tweet* yang mengandung potensi informasi kemacetan lalu lintas dengan *tweet* yang tidak mengandung informasi kemacetan lalu lintas. Hasil *tweet* yang mengandung potensi informasi kemacetan kemudian diproses untuk diambil informasi mengenai lokasi kemacetan lalu lintas. Sakaki dkk. hanya menggunakan gazetteer dari Wikipedia dan Hatena *Keywords*, meskipun metode yang digunakan sederhana hasil pengukuran F-measure metode ini cukup baik yaitu 0,8.

Selain penelitian Sakaki dkk., terdapat juga penelitian ekstraksi informasi kemacetan untuk domain *tweet* berbahasa Indonesia. Hasby & Khodra (2013) menggunakan algoritma SMO (*Sequential Minimal Optimization*) untuk melakukan ekstraksi entitas pada *tweet* yang sudah dinormalisasi. Hasby & Khodra menggunakan fitur leksikal kata (urutan kata dalam kalimat), POS *Tag* 2 kata bertetangga, fitur ortografis, *Tag* *Name Entity* dan *Gazeteer* nama jalan di Bandung.(Hasby & Khodra, 2013)

Untuk penelitian pada bidang *Named Entity Recognition* pada domain Twitter Bontcheva membuat penelitian khusus (Bontcheva dkk., 2013). Bontcheva dkk. memodifikasi GATE[[2]](#footnote-2) dengan menambahkan modul khusus bernama *Tweet*IE. *Tweet*IE melakukan normalisasi dan melakukan adaptasi terhadap Stanford *Parser* yang dilatih dalam *tweets* PennTreeBank dengan tambahan fitur-fitur spesifik Twitter (*mention ,hashtag ,retweet*) (Bontcheva dkk., 2013). Hasil NER milik Bontcheva dkk. memiliki F-Measure yang cukup baik yaitu 0,8, jauh lebih baik dari NER milik Ritter.

Tabel II‑4 meringkas hasil dari penelitian-penelitian terkait

Tabel II‑4 Ringkasan penelitian terkait

|  |  |  |  |
| --- | --- | --- | --- |
| **Deteksi *Event* dari Twitter** | | | |
| Judul Penelitian | Fitur | Algoritma | Hasil |
| Earthquake shakes Twitter users: Real time *Event* Detection by Social Sensors (Sakaki dkk.) | * Jumlah Kemunculan suatu kata dalam *tweet* * Kata yang terdapat dalam *tweet* | Kombinasi dengan algoritma pribadi & Filter Kalman | Precission:  87.5 %  Recall:  63.64%  F-Measure:  73.69% |
| *Event* Radar : A Real Time Local *Event* Detection Scheme Using Twitter Stream | * Meta Data Geolocation * MetaData Waktu | Klustering | Precission:  4.65%  Recall :  62% |
| Detecting Local *Event*s by Analyzing Spatiotemporal Locality of *Tweet*s  (Sugitani dkk.) | * Meta Data Geolocation * Meta Data waktu *tweet* dibuat * Kata dalam *Tweet* | Klustering | Precission :  50%  Recall :  40% |
| **Ekstraksi *Event* dari Twitter** | | | |
| Judul Penelitian | Fitur | Algoritma | Hasil |
| Open Domain *Event* Extraction from Twitter (Ritter dkk.) | * POS *Tag* khusus milik Ritter (baca bagian NER) * Konteks * Kamus ortografis * Kamus | CRF | Precission:  56%  Recall:  74%  F-Measure:  64% |
| Extraction and Compilation of *Event*s and Sub-*event*s from Twitter  (Khurdiya dkk.) | * POS *Tag* khusus * Konteks * Kamus ortografis * Gazeteer | CRF | Precission:  70%  Recall :  59%  F-Measure:  64% |
| **Ekstraksi Informasi dari Twitter (Domain Berbeda)** | | | |
| Judul Penelitian | Fitur | Algoritma | Hasil |
| Online Path Finding based on Traffic Information Extraction from Twitter  (Hasby & Khodra) | * Fitur Leksikal * Fitur POS *Tag* * Informasi Ortografis * Gazeteer * Element *Tweet* * Simbol Khusus | SMO (Sequential Minimal Optimization) | F-Measure : 93.1% |
| Real Time *Event* Extraction for Driving Information from Social Sensor (Sakaki dkk.) | * Gazeteer * Leksikal | SVM | Precission :  85%  Recall:  37&  F-Measure:  47% |
| **Named Entity Recognition pada Domain Twitter** | | | |
| Judul Penelitian | Fitur | Algoritma | Hasil |
| Named Entity Recognition in *Tweet*s: Experimental Study (Ritter dkk.) | * POS *Tag* * Fitur Ortografis * Shallow Parsing | Labelled LDA | Precission :  73%  Recall:  61%  F-Measure:  67% |
| TwitIE: An Open-Source Information Extraction Pipeline  for Microblog Text  (Bontcheva dkk.) | * Elemen spesifik *Tweet* * Fitur-fitur lain yan dipakai dalam POS *Tag* Stanford | GATE | Precission:  77%  Recall:  83%  F-Measure:  80% |

# Ekstraksi informasi event

## Analisis Permasalahan

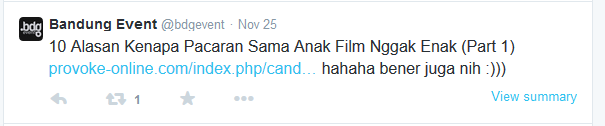
Ekstraksi informasi *event* dari Twitter dapat berguna karena memenuhi syarat-syarat kriteria ekstraksi informasi *event* (Herbich & Graepel, 2010). Dalam ekstraksi informasi *event* dari Twitter, terdapat sejumlah *template* yang mampu meringkas informasi yaitu pelaku *event*, nama *event*, waktu, lokasi *event* dan pelengkap *event*. Informasi-informasi tersebut didapatkan secara tersurat tanpa melalui proses inferensi.

Ada beberapa karakteristik yang perlu diperhatikan dalam merancang solusi ekstraksi informasi *event* dari Twitter. Pertama, Tata bahasa di Twitter tidak baku, selalu ada kemungkinan kata-kata baru muncul (*slang words*). Kedua, terdapat redundansi data.

### Analisis Informasi *event* pada *tweet*

Ekstraksi informasi dilakukan pada sekumpulan *tweet* dari Twitter. Sistem ekstraksi informasi akan menerima input sekumpulan *tweet* hasil *search* dengan *keyword* tertentu atau *hashtag* tertentu.

Meskipun informasi yang dikandung dalam hasil search *keyword* atau *hashtags* tersebut kebanyakan membahas tentang informasi *event*, tidak menutup kemungkinan masih terdapat *noise*. *Noise* didefinisikan sebagai *tweets* yang tidak relevan dan tidak mengandung informasi *event*. *Noise* yang terdapat pada *tweets* tersebut bisa berupa interaksi pemilik akun dengan *follower*-nya, atau *retweet* mengenai promo sesuatu. *Tweet* *noise* sengaja dibuat oleh admin akun-akun tersebut untuk membentuk interaksi antara akun tersebut dengan *follower*-nya agar terlihat hidup seperti pada Gambar III‑1.



Gambar III‑1 Contoh *tweet* noise milik akun twitter @Bdg*Event*.

Selain mengandung *noise*, informasi *event* di Twitter rawan terhadap duplikasi informasi. Duplikasi informasi terjadi saat suatu akun melakukan re*tweet* terhadap akun lainya, saat melakukan *retweet* terhadap followernya, atau pada *tweet* yang memang dipromosikan oleh akun Twitter tersebut. *Retweet* diindikasikan dengan keberadaan literal ‘RT’ di depan *tweet*.

Ada karakteristik unik dalam teks pada *tweet*. Teks pada *tweet* di Twitter berbeda dengan teks pada dokumen konvensional karena Twitter membatasi jumlah karakter dalam satu *tweet*, yaitu sebanyak 140 karakter. Dalam perhitungan jumlah karakter, Twitter memiliki kebijakan untuk menggunakan bentuk normalisasi C (*normalized form C*) dari *tweet*. Pembatasan jumlah karakter ini membuat pengguna twitter harus kreatif dalam menyampaikan pesan.

Twitter mendukung penggunaan *encoding character* UTF-8. Penggunaan *encoding character* tersebut mengakibatkan pengguna Twitter dapat menggunakan blok karakter emoji atau blok karakter ornamen di *character set* UTF-8. Hal ini mengakibatkan pengguna twitter bisa lebih bebas dalam mengungkapkan ekspresi di twitter sekaligus menjadi tantangan tambahan dalam proses ekstraksi informasi *tweet*.

Salah satu hal unik yang ditemukan dalam *tweets* adalah penggunaan *emoticon*. Ada dua jenis *emoticon* yang digunakan di twitter yaitu *emoticon* konvensional dan *emoticon* spesial. *Emoticon* konvensional dibentuk dari kombinasi beberapa tanda baca sementara *emoticon* spesial dibentuk dari karakter unicode pada blok emoji.

Salah satu contoh *emoticon* konvensional adalah ‘:-)’. *Emoticon* tersebut dibentuk dari 3 buah tanda baca yaitu simbol titik dua, simbol dash dan tanda kurung. Twitter menghitung *emoticon* ‘:-)’ sebagai literal dengan panjang 3 karakter.

Adapun contoh *emoticon* spesial adalah ‘☺’. *Emoticon* tersebut dibentuk dari 1 buah karakter pada blok unicode emoji dengan *codepoint* U+263A. Pada kasus ini Twitter menghitung *emoticon* ‘☺’ sebagai literal dengan panjang 1 karakter.

Tantangan lainya terdapat pada bentuk distribusi *tweet* terhadap waktu. Menurut *report* yang dikeluarkan oleh Amiando, jumlah *tweet* tidak terdistribusi secara merata. Hanya ada 18 % *tweet* mengenai suatu *event* yang muncul sebelum *event*, 60 % *tweet* muncul saat selama *event* berlangsung dan 22% *tweet* muncul pasca-*event* (Ferenczy dkk., 2010).

*Tweets* yang mengandung informasi seputar *event* biasanya mengandung komponen-komponen informasi berupa :

1. Nama *Event*

Nama *event* adalah informasi utama yang diambil pada task ekstraksi informasi. Nama *event* adalah entitas utama yang mewakili suatu *event*.

1. Waktu *Event*

Waktu *event* adalah waktu saat berlangsungnya *event*. Ada tiga kemungkinan nilai waktu *event*. Waktu *event* yang sudah berlalu, waktu *event* sedang berlangsung dan waktu yang akan datang. Fokus tugas akhir ini ada pada event yang akan datang.

1. Lokasi *Event*

Lokasi *event* adalah tempat *event* akan berlangsung. Menurut Boettcher (2012), berdasarkan lokasi kejadian *event, event* dapat dikategorikan menjadi dua jenis yaitu *event* lokal dan *event* global. *Event* lokal didefinisikan sebagai *event* yang dibatasi koordinat geografis tertentu sementara *event* global adalah *event* yang tidak dibatasi oleh koordinat geografis. Sebagai contoh, *Event* “Perayaan 17 Agustus HUT RI” berbeda dengan *event* “lomba makan kerupuk 17 Agustus di alun-alun kota bandung”. *Event* “Perayaan 17 Agustus HUT RI” dianggap sebagai event global karena kejadianya tidak dibatasi oleh koordinat geografis tertentu. Adapun *event* “lomba makan kerupuk 17 Agustus di alun-alun kota bandung” dianggap sebagai event lokal karena keberadaanya dibatasi oleh lokasi geografis yaitu alun-alun kota bandung. Fokus pengerjaan tugas akhir ini ada pada event lokal.

1. Pelengkap *Event*

Pelengkap *event* dapat berupa Nomor handphone *Contact Person*, Gambar, atau link menuju suatu halaman web. Berbeda dengan konten promosi atau ajakan untuk mengikuti acara, informasi yang terdapat pada bagian ini masih dapat melengkapi informasi nama waktu dan lokasi event.

*Tweets* yang mengandung keempat poin di atas biasanya menunjukkan informasi mengenai *event*. Akan tetapi suatu *tweet* mengenai informasi *event* belum tentu mengandung semua komponen di atas. Salah satu contohnya *tweet* pada gambar Gambar III‑2.



Gambar III‑2 Contoh *tweet* dengan informasi tak lengkap

Konten *tweet* tersebut meskipun memiliki judul (WUBIFEST) tetapi tidak memiliki informasi eksplisit mengenai tempat dan waktu acara. Informasi tersebut didapatkan dari bagian pelengkap *event* dalam kasus ini berupa URL sebuah gambar poster promosi acara tersebut.

*Tweets* yang mengandung informasi *event* kebanyakan tidak memiliki tata bahasa baku. Karena jumlah karakter dalam 1 *tweet* dibatasi sejumlah 140 karakter, sering ditemukan *tweets* yang hanya berupa gabungan nama entitas-entitas saja. Sebagaimana yang ditunjukan dalam gambar Gambar III‑3



Gambar III‑3 Contoh *tweet* yang hanya mengandung kombinasi nama entitas

Pada Gambar III‑3 tidak ditemukan kata kerja, informasi antar entitas dipisahkan oleh *token* ‘|’. Kalimat dalam *tweet* tidak memenuhi tata bahasa baku karena tidak memiliki subjek atau predikat yang eksplisit.

*Tweets* di twitter tidak terdapat urutan penyebutan entitas yang konsisten. Sebagai contoh, *tweet* pada Gambar III‑3 dan Gambar III‑4. Urutan penyebutan entitas pada Gambar III‑3 berbeda dengan urutan penyebutan entitas pada Gambar III‑4. Pada Gambar III‑3 urutan entitas yang disebut berturut-turut adalah tanggal *event*, nama *event*, subjek-subjek *event*, lokasi *event*, dan informasi pelengkap. Adapun pada Gambar III‑4, urutan entitas yang disebut adalah nama *event*, subjek-subjek *event*,waktu *event*, lokasi *event* dan informasi pelengkap.



Gambar III‑4 Contoh *tweet* yang hanya menyebtukan entitas   
dengan urutan berbeda

### Ekstraksi Informasi

Dalam hal ekstraksi informasi *event*, Sakaki dkk. (2012) membagi proses ekstraksi informasi *event* dalam dua bagian: Pertama, deteksi *tweet* yang berpeluang besar mengandung informasi mengenai *event*. Kedua, ekstraksi komponen penyusun informasi *event* dari *tweet*s yang berpeluang besar mengandung informasi mengenai *event*. Pada tugas akhir ini juga akan dilakukan pendekatan yang sama. Tugas akhir ini akan memecah proses ekstraksi informasi event dalam dua bagian: Pertama, mendeteksi kandidat *tweets* yang mengandung *informasi event*; Kedua melakukan ekstraksi komponen penyusun informasi event.

Sebelumnya pada bab 2 dibahas tentang penelitian-penelitian terkait. Kebanyakan penelitian memiliki fokus pada bagian deteksi *tweet* yang berpeluang besar mengandung informasi mengenai *event* (Li dkk., 2012; Sugitani dkk., 2012). Berbeda dengan penelitian-penelitian tersebut, tugas akhir ini memiliki fokus pada bagian ekstraksi komponen penyusun informasi *event*. Tugas akhir ini tidak berfokus mengenai persoalan bagaimana suatu *tweet* dianggap berpeluang besar mengandung informasi *event*.

Bagian pertama dalam tugas akhir ini adalah persoalan bagaimana memisahkan *tweet* *noise* dengan *tweet* yang bukan *noise* sebelum diproses di bagian kedua. Bagian ini dibutuhkan untuk mengurangi pengaruh *tweet* *noise* terhadap kinerja bagian yang melakukan ekstraksi komponen penyusun informasi event. Dengan demikian diharapkan model yang dihasilkan pada bagian kedua akan memiliki kinerja yang lebih baik dibanding tanpa menggunakan bagian pertama.

Bagian kedua dalam tugas akhir ini adalah persoalan bagaimana menentukan kata dalam *tweet* yang mewakili salah satu komponen penyusun informasi event. Pada bagian ini dilakukan proses Named Entity Recognition (NER). Bagian ini adalah bagian utama dalam sistem ekstraksi informasi *event*

Menurut Jurafsky & Martin (2008), ekstraksi informasi dapat dianggap sebagai persoalan *sequence labelling*. Ada banyak algoritma yang dapat digunakan untuk melakukan *sequence labelling*. Berdasarkan penelitian yang dilakukan oleh Ritter (2012), algoritma CRF memberikan akurasi yang baik dalam persoalan NER. Oleh karena itu pada penelitian ini juga akang menggunakan algoritma CRF sebagai algoritma pembelajaran.

CRF termasuk algoritma pembelajaran jenis *supervised learning* dengan kategori model diskriminatif. Algoritma *supervised learning* membutuhkan *dataset* sebagai data latih. Pada kasus ini digunakan *dataset* berupa himpunan sejumlah *tweet* yang telah diberi label untuk melatih bagian yang melakukan ekstraksi komponen penyusun informasi event.

Untuk memenuhi kebutuhan *labelling* NER digunakan beberapa *label* sesuai dengan entitas-entitas pembentuk informasi *event*. Entitas-entitas tersebut adalah nama *event*, waktu *event,* lokasi *event*, dan pelengkap *event*. Karena kategori CRF adalah model diskriminatif, tidak perlu digunakan notasi BIO. Notasi BIO hanya dibutuhkan oleh algoritma generatif. Dalam tugas akhir ini digunakan beberapa label seperti yang terlihat pada Tabel III‑1 .

Tabel III‑1 Label-label yang digunakan dalam NER

|  |  |
| --- | --- |
| **Label** | **Deskripsi** |
| I-Name | Label untuk kata-kata yang menyatakan nama event |
| I-Time | Label untuk kata-kata yang menyatakan waktu event |
| I-Place | Label untuk kata-kata yang menyatakan lokasi event |
| I-Info | Label untuk kata-kata yang menyatakan informasi pelengkap event. Beberapa jenis informasi pelengkap tersebut diantaranya adalah harga tiket, nomor telpon *contact person*, gambar atau tautan ke halaman website. |
| Other | Label untuk kata-kata selain diluar jenis informasi yang ingin diamati (Nama, Waktu, Lokasi, dan informasi pelengkap) |

Pada tugas akhir ini, bagian NER akan memanfaatkan beberapa fitur. Fitur-fitur yang digunakan diantaranya adalah,

1. *Part of speech* *Tag* (POS Tag)

Menurut Ritter dkk.(2011), penambahan komponen POS *Tagger* dapat meningkatkan akurasi ekstraksi informasi. Oleh karena itu pada tugas akhir ini digunakan pula komponen POS *Tagger* yang diadaptasi untuk bahasa Indonesia. Himpunan POS Tag (*tagset*) yang digunakan pada lingkup tugas akhir ini adalah tagset Dionysius Thrax.

1. *Named Entity Tag*.

*Named entity Tag* yang digunakan adalah *Named Entity Tag* untuk kata ke-n, kata ke-n-1, kata ke-n-2, kata ke-n+1 dan kata ke-n+2. Fitur ini digunakan karena *Named Entity* pada suatu *token* kadang bergantung pada *Named Entity* *token* sebelumnya / setelahnya. Untuk membedakan entitas satu dengan entitas yang lainya terkadang tidak cukup hanya memanfaatkan tag *named entity* kata-kata yang muncul sebelum kata yang sedang diobservasi, melainkan harus melibatkan juga tag named entity kata-kata yang muncul setelah kata yang sedang diobservasi. Dalam beberapa kasus saat satu *tweet* hanya mengandung entitas-entitas saja (seperti pada Gambar III‑3), batas antar entitas menjadi tidak jelas. Dalam *tweet* pada Gambar III‑3, literal ‘9’dianggap mewakili entitas waktu karena literal ‘Nov’(Singkatan dari November) yang muncul setelahnya diklasifikasikan sebagai literal yang mewakili entitas waktu.

1. Fitur leksikal.

Fitur leksikal yang digunakan adalah kata ke-n-1,kata ke-n dan kata ke-n+1. Fitur ini digunakan karena dalam entitas terdapat hubungan yang kuat dengan *token*-*token* tertentu. Contohnya, entitas lokasi biasa dimulai dengan *token* “di”, entitas waktu biasanya mengandung *token* “jam”.

1. *Gazeteer*

*Gazeteer*[[3]](#footnote-3) digunakan untuk memudahkan proses pencarian entitas lokasi. Karena dalam Tugas akhir ini penelitian dibatasi pada ekstraksi informasi *event* di kota bandung, maka *Gazeteer* yang dikumpulkan adalah *Gazeteer* lokasi-lokasi di bandung. Untuk menyederhanakan persoalan digunakan Gazeteer berupa informasi tempat-tempat publik yang biasa diselenggarakan *event*. Misal Balai Kota, mall PVJ, dst.

1. Elemen *Tweet*

Elemen *tweet* yang digunakan adalah *hashtag*, *mention*, atau tautan URL. Fitur ini digunakan karena dalam beberapa kasus, suatu *event* memiliki akun Twitter khusus yang sama dengan nama *event* tersebut. Sebagai contoh adalah akun Twitter Braga Culinary Night (@bragaculinary).

1. Simbol Khusus

Sebagaimana yang terlihat pada gambar Gambar III‑3, terkadang antar entitas dipisahkan oleh simbol-simbol tertentu. Untuk kasus Gambar III‑3, antar entitas dipisahkan dengan simbol ‘|’.

Informasi ortografis diabaikan dalam fitur karena tidak terdapat konsistensi penggunaan ortografis dalam *tweet*. Sering ditemukan *tweet* yang kontenya seluruhnya dalam huruf kapital. Kadang ditemukan juga *tweet* yang fitur ortografisnya saling bercampur. Hal ini dilakukan oleh pengguna twitter untuk memancing perhatian *follower* mereka. Oleh karena itu pada bagian NER tidak memanfaatkan fitur ortografis.

## Deskripsi Solusi

Dalam proses ekstraksi informasi *event*, digunakan acuan tahapan proses ekstraksi informasi umum yang dijelaskan oleh Jurafsky (Jurafsky & Manning, 2008). Akan tetapi karena proses ekstraksi informasi *event* dilakukan pada domain teks Twitter, perlu dilakukan adaptasi terhadap tahapan-tahapan ekstraksi informasi.

Ada tak hingga kemungkinan penamaan entitas-entitas yang mewakili komponen pembentuk *event*. Oleh karena itu tahapan *Named Entity Recognition* sangat dibutuhkan.

Di dalam domain Twitter, satu dokumen ekivalen dengan satu buah *tweet*. Karena jumlah karakter yang dibatasi, hanya sedikit kombinasi informasi yang mungkin ada pada satu dokumen (*tweet*). Karenanya, tahapan *relation extraction* & *event detection and classification* kurang relevan dengan kebutuhan sistem ini sehingga dapat dihilangkan.

Setiap pengguna di twitter memiliki cara unik tersendiri untuk mengekspresikan informasi mengenai kapan suatu *event* akan berlangsung. Tidak ada standar baku yang menjamin akan ada format tertentu mengenai bagaimana pengguna twitter menyebut informasi temporal suatu *event*. Oleh karena itu dibutuhkan tahapan *Temporal Analysis* dan *Template Filling*.

### Analisis Keterkaitan dengan Penelitian Lain

Sistem yang akan dibangun sebenarnya memiliki kemiripan dengan beberapa penelitian yang lain. Penelitian-penelitian yang cukup mirip dengan tugas akhir ini adalah penelitian tentang ekstraksi informasi kemacetan lalu lintas dari twitter pada domain bahasa indonesia.

Dalam penelitian tersebut dijelaskan bahwa arsitektur sistem bagian ekstraksi informasi terdiri dari tokenisasi, normalisasi, NER, Template Element Task, Relation Extraction, Information filling. Ritter dkk. (2012) menggunakan POS *tag*, *temporal resolver*, *named entity recognition*, *event tagger* dalam proses melakukan ekstraksi informasi.

Dari hasil pengamatan, ternyata kerangka ekstraksi informasi yang dilakukan oleh ritter untuk kasus penelitian tugas akhir ini terlalu berlebihan. Sebagai contoh, dalam kerangka sistem ekstraksi yang diusulkan Ritter dibutuhkan modul *relation extraction*. Modul ini melakukan ekstraksi hubungan entitas yang ada di dalam *tweet*. *Tweet* mengenai event di dalam bahasa indonesia tidak memiliki banyak variasi dalam penyebutan event. Rata-rata *tweet* event bahasa indonesia memiliki POS *tag* *Noun*(kata benda). Sedikit *tweet* mengenai event di dalam bahasa indonesia yang memiliki POS *adjective* (kata sifat) atau verb (kata kerja). Oleh karena itu tidak seluruh modul dalam arsitektur yang diusulkan Ritter dkk.

### Rancangan Arsitektur Sistem

Proses ekstraksi informasi memiliki beberapa tahapan. Pertama proses tokenisasi, kedua proses analisis leksikal dan morfologi, ketiga proses analisis sintaksis dan terakhir proses analisis domain (Jurafsky, 2008). Dalam tugas akhir ini hanya digunakan proses tokenisasi, analisis leksikal dan analisis morfologi. Hal ini disebabkan bentuk informasi teks di twitter tidak memiliki struktur tata bahasa baku sehingga sangat sulit dilakukan proses analisis sintaksis. Karena panjang karakter dalam *tweet* dibatasi hanya 140 karakter, kemungkinan penggunaan anafora akan sangat kecil.

Sistem akan memanfaatkan akun-akun Twitter yang sudah melakukan agregasi informasi *event* secara manual sebagai *training dataset*. Selain *tweet* yang didapat dari akun tersebut, sistem juga akan menerima input dari *hashtags* yang mengindikasikan suatu *tweet* memiliki peluang mengandung informasi *event*. Untuk menyederhanakan persoalan, penelitian ini akan menerima *tweet* dari akun Twitter yang menghimpun acara-acara di sekitar bandung.

Dalam tugas akhir ini digunakan beberapa sumber korpus:

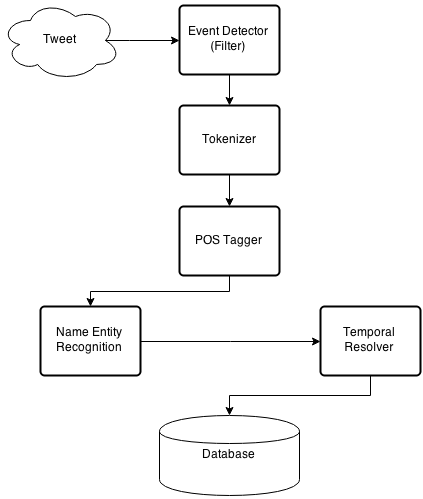
1. *Tweet* milik akun pribadi selain @infobdgevent yang mengandung *mention* terhadap @infobdgevent selain *retweet*.
2. *Tweet* milik akun pribadi selain @bdgevent yang mengandung *mention* terhadap @bdgevent selain *retweet*.
3. *Tweet* milik @infobdgevent termasuk *retweet*.
4. *Tweet* milik @bdgevent termasuk *retweet*.
5. Hasil pencarian *hashtag* #eventbdg
6. Hasil pencarian *hashtag* #eventbandung

Akun-akun atau *hashtags* tersebut dipilih karena kedua akun tersebut adalah akun twitter yang paling memiliki pengaruh di bandung. Kedua akun tersebut memiliki jumlah *tweet* paling banyak dibanding akun-akun twitter sejenis di bandung. Selain itu, kedua akun tersebut adalah akun agregator informasi dengan jumlah *follower* tertinggi se-bandung.

*Tweet*s yang mengandung *mention* terhadap @infobdgevent/@bdgevent dianggap sebagai upaya pengguna twitter mengumpulkan informasi *event*. Para pengguna twitter mengumpulkan informasi *event* dengan cara memberikan sugesti *tweet* kepada akun twitter @infobdgevent atau @bdgevent. Para pemilik akun pribadi ingin mengumpulkan informasi event kepada @infobdgevent/@bdgevent dan berharap *tweet* mereka di-*retweet* oleh akun @infobdgevent/@bdgevent. Meskipun demikian, kebanyakan *tweet* yang mengandung *mention* kepada @infobdgevent atau @bdgevent berupa *retweet* oleh akun lainya. Seandainya re*tweet* tidak diabaikan, akan banyak ditemukan redundansi informasi dan mengganggu model yang dibentuk. Oleh karena itu re*tweet* yang mengandung *mention* terhadap @infobdgevent harus diabaikan.

Karena akun @infobdgevent / @bdgevent didedikasikan untuk melakukan agregasi event, *tweets* dari kedua akun tersebut dapat langsung dijadikan sumber korpus. Berbeda dengan sumber korpus ke-1& ke-2, sumber korpus ke-3 & ke-4 tidak mengabaikan *retweet*. Hal ini didasari bahwa kedua akun tersebut mungkin saja melakukan *retweet* terhadap sugesti informasi event yang diberikan oleh akun lain. Oleh karena itu *retweet* yang dilakukan oleh akun @infobdgevent / @bdgevent tidak boleh diabaikan.

Pengguna twitter juga sering mengumpulkan informasi event dengan bantuan *hashtag*. Menurut Twitter(2014), fitur *hashtag* dibuat agar *tweets* yang memiliki kesamaan topik lebih mudah dikelompokan. Dalam kasus informasi event, pengguna twitter di bandung biasa menggunakan *hashtag* #eventbandung atau #eventbdg.



Gambar III‑5 Arsitektur sistem ekstraksi informasi event bahasa indonesia

Untuk memenuhi kebutuhan pada bagian analisis permasalahan, bentuk arsitektur yang digunakan seperti yang terdapat dalam Gambar III‑5**Error! Reference source not found.**. Komponen arsitektur terdiri dari:

1. *Event Detector* / *Filter*

Filter dibutuhkan untuk memisahkan *tweets* yang berpotensi besar mengandung informasi suatu *event* dengan *tweets* yang hanya mengandung noise. Dalam hasil search suatu keyword tertentu terkadang dihasilkan *tweets* pribadi yang kurang relevan terhadap sistem ekstraksi. Modul ini akan mengeliminasi *tweets* yang tidak relevan sehingga hanya *tweets* yang relevan yang akan dimasukan ke dalam sistem. Sebagai contoh, *tweet* seperti ‘@InfoBdgEvent dimana ka?’ yang tidak memiliki informasi event sama sekali akan diabaikan dan tidak masuk ke proses selanjutnya.

Pada tugas akhir ini, *tweet noise* didefinisikan sebagai *tweets* yang tidak memiliki informasi sama sekali mengenai *event*. Satu *tweet* dikatakan bukan *tweet noise* jika memiliki salah satu diantara 4 komponen penyusun informasi event. Misal, dalam *tweet* ‘Ke Dago Tea House y 8/3! Akan ada Pagelaran 7 Etnis Sumatera Utara #HorasParahyangan ! More info? @HorasBandung <http://t.co/eCEu29WNVs>’ Terdapat 4 jenis informasi penyusun event: Lokasi event yaitu Dago Tea House; Waktu event yaitu tanggal ‘8/3’(8 maret); Nama event yaitu ‘Pagelaran 7 Etnis Sumatera Utara #HorasParahyangan’; dan Informasi pelengkap berupa tautan poster event yaitu ‘http://t.co/eCEu29WNVs’. *Tweet* ini tidak akan diabaikan dan diteruskan oleh sistem untuk proses selanjutnya.

Salah satu fitur yang disediakan oleh Twitter adalah kemampuan untuk menambahkan entitas-entitas selain teks. Entitas-entitas tersebut diantaranya adalah url, *mention* dan media. Dalam beberapa kasus, entitas-entitas ini dapat membentuk salah satu dari 4 penyusun informasi event. Sebagai contoh, dalam *tweet* ‘THE EXECUTIVE SUPER SALE up to 80% off Last 2 Days!THE EXECUTIVE boutiques @istanaplaza @InfoBdgEvent <http://t.co/tQyZRRjO26>’ lokasi *event* diwakili oleh sebuah *mention* yaitu @istanaplaza. Dengan demikian entitas-entitas seperti url, *mention* dan media berpotensi mengidentifikasi keberadaan salah satu diantara 4 komponen penyusun informasi event.

Kebijakan pembatasan jumlah karakter dalam sebuah *tweet* di Twitter mengakibatkan pengguna Twitter harus hemat dalam melakukan *tweeting*. Di sisi lain, seandainya jumlah karakter *tweet* terlalu sedikit, bisa jadi komponen penyusun informasi *event* tidak dapat dicantumkan dalam *tweet*. Dari analisis ini disimpulkan bahwa panjang karakter *tweet* berpotensi menjadi salah satu penentu apakah *tweet* dikatakan relevan atau tidak. Oleh karena itu pada tugas akhir ini identifikasi terhadap *tweet* yang *relevan* / tidak menggunakan pendekatan berbasis aturan.

Entitas-entitas di Twitter seperti url, *mention* dan media memiliki panjang karakter yang khas. Twitter menentukan bahwa setiap url yang ada di *tweet* akan dipotong menjadi tautan singkat dengan penyingkat tautan <http://t.co> milik Twitter. Saat ini, panjang tautan setelah dipotong oleh <http://t.co> akan selalu 23 karakter. Berkaitan dengan entitas *mention*, twitter memiliki kebijakan bahwa panjang maksimal karakter *username* twitter ada 15 karakter tanpa simbol ‘@’. Sementara itu, dalam hal panjang karakter entitas media, saat ini twitter membatasi jumlah media di dalam *tweet* maksimal ada 1 buah media. Twitter memperlakukan entitas media sebagaimana entitas url. Untuk setiap url menuju media, twitter akan secara otomatis memotong panjang url media dengan penyingkat tautan <http://t.co>. Dengan kata lain, panjang karakter media dalam *tweet* akan selalu 23 karakter.

Twitter memang tidak membatasi jumlah maksimal entitas url dan *mention* yang ada di dalam *tweet*. Meskipun demikian, pembatasan jumlah 140 karakter *tweet* akan secara tidak langsung membatasi jumlah entitas url dan *mention* yang ada di dalam *tweet*. Untuk entitas url, jumlah maksimum url yang ada di *tweet* adalah floor(140 karakter / 23 karakter) = 6 buah url. Untuk entitas *mention*, jumlah maksimum *mention* yang ada di dalam *tweet* adalah floor(140 karakter / 15 karakter) = 9 *mention*. Tentu jumlah *mention* yang dikandung dalam 1 *tweet* bisa lebih banyak jika terdapat *username* dengan karakter yang lebih pendek dari 15 karakter. Sekedar informasi tambahan, menurut penelitian yang dilakukan pada tahun 2012, rata-rata panjang karakter *username* twitter ada 11 karakter. Jika perhitungan dilakukan menggunakan standar ini, setidaknya maksimum jumlah *mention* adalah floor (140 karakter /11 karakter) = 12 *mention*.

Dari uraian sebelumnya, komponen penyusun informasi mungkin diwakili oleh entitas-entitas ini (url, *mention* atau media). Informasi pelengkap *event* biasa diwakili oleh url atau media. Informasi nama *event* kadang diwakili oleh *mention* terhadap akun resmi nama *event* yang berkaitan. Sebagai contoh, dalam *tweet* ‘Enjoy @MalamEP2015! Live music, pom-pom boys & @THEPAPSPAPS | HTM 20K. CP: 08997127861 | Info: @malam\_ep15 <http://t.co/XgBZkyaIQM>’ komponen informasi nama *event* diwakili dengan *mention* terhadap ‘@MalamEP2015’ dan komponen informasi pelengkap *event* diwakili salah satunya oleh url poster yaitu ‘ ‘http://t.co/XgBZkyaIQM’.

Dengan asumsi bahwa setiap pengguna ingin selalu memaksimalkan penggunaan jumlah karakter di twitter (140 karakter), panjang karakter minimum *tweets* yang relevan dapat ditentukan. Sebagai contoh, *tweet* dengan informasi pelengkap *event* yang diwakili oleh entitas url atau media akan memiliki panjang minimal 140 karakter – 23 karakter url= 117 karakter. Sementara *tweet* dengan informasi nama *event* yang diwakili oleh entitas *mention* akan memiliki panjang minimal 140 karakter – 15 karakter *mention* = 125 karakter. Batas panjang karakter minimum dalam *tweet* ini mampu membedakan antara *tweet* yang tidak relevan dan yang relevan.

Berdasarkan data pengamatan, panjang karakter rata-rata *tweet* yang relevan ada 96 karakter sementara panjang karater rata-rata *tweet* yang tidak relevan ada 68 karakter. Nilai tengah keduanya adalah 82 karakter. Dengan asumsi data terdistribusi homogen, maka dibuat aturan jika panjang karakter setelah dilakukan eliminasi *mention*, url dan hashtag melebihi 82 karakter, maka *tweet* diklasifikasikan sebagai relevan. Begitu juga sebaliknya, jika panjang karakter *tweet* setelah dilakukan eliminasi *mention*, url dan hashtag kurang dari 82 karakter, maka *tweet* diklasifikasikan sebagai tak relevan. Aturan inilah yang digunakan dalam komponen filter.

Selain menggunakan aturan perhitungan panjang karakter, alternatif aturan lain adalah menggunakan aturan berbasis literal. Berdasarkan pengamatan didapatkan bahwa kebanyakan *tweet* yang tidak relevan termasuk dalam kategori iklan atau berupa pertanyaan. Dalam hal ini *tweet* kategori iklan biasanya memiliki literal tertentu yang menunjukan *tweet* tersebut adalah *tweet* iklan seperti literal ‘dijual’ atau ‘disewakan’. Sementara *tweet* pertanyaan memiliki token unik seperti simbol tanda tanya (?) atau kata-kata pertanyaan (‘siapa’,‘berapa’,‘kapan’).

1. *Tokenizer*

Sesuai dengan arsitektur umum yang sudah dijelaskan dalam bab 2 (Gambar II‑4), arsitektur harus mengandung *tokenizer*. *Tokenizer* berfungsi memecah teks menjadi unit terkecil dalam pemrosesan bahasa yang disebut sebagai token. Dalam konteks pemrosesan teks di media sosial, unit terkecil teks adalah potongan kata.

Pada tugas akhir ini digunakan modul *Tokenizer* milik Ark[[4]](#footnote-4). Modul tersebut bekerja dengan melakukan string matching dengan suatu pattern regular expression tertentu. Modul tersebut dimodifikasi dengan menambahkan beberapa aturan agar sesuai dengan kebutuhan bahasa indonesia.

Ada beberapa pattern yang ditambahkan dari modul twokenizer bawaaan Ark. Pattern-pattern tersebut diadptasi sesuai dengan data yang masuk. Berikut pattern yang digunakan dalam tokenizer:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Pattern** | **Pattern Regular Expression** | **Contoh *matching words*** | **Catatan** |
| Nama Hari | (?>senin|selasa|rabu|kamis|jumat|sabtu|minggu|ahad|sunday|monday|tuesday|wednesday|thursday|friday|saturday|sun|mon|tues|wed|thu|fri|sat|snn|sls|rb|kms|jmt|sab|ming) | Senin; selasa; rabu; dst. |  |
| Bulan dalam angka | (?:1[012]|0?+[1-9]) | 0; 1; 02; 2 dst. | Bulan dalam kalender masehi hanya dari 0-12, tidak ada bulan ke-13 |
| Nama Bulan dalam bahasa indonesia | (?i)(?>januari|februari|maret|april|mei|juni|juli|agustus|september|oktober|november|desember) | Januari februari dst. |  |
| singkatan nama bulan dalam bahasa indonesia | (?i)(?>jan|feb|mar|apr|mei|jun|jul|agt|sep|okt|nov|des) | feb, mar dst. |  |
| Nama bulan dalam bahasa inggris | (?i)(?>January|february|march|april|may|june|july|august|september|october|november|december) | january, february, |  |
| Singkatan nama bulan dalam bahasa inggris | (?i)(?>jan|feb|mar|apr|may|jun|jul|aug|sept|oct|nov|dec) | Jan, feb, dst.. |  |
| **Nama Pattern** | **Pattern Regular Expression** | **Contoh *matching words*** | **Catatan** |
| Tahun | ((?:19|20)?\\d\\d) | 1999; 2099 | Untuk menyederhanakan persoalan, dianggap tahun valid antara 1900-2099 |
| Indikator waktu pada jam | (?i)(?>jam|pk|pukul|pk\\.) | Pukul, pk. Dst.. |  |
| Jam | (2[0-3]|1[0-9]|0?+[1-9]) | 0; 1; 23 | Jam hanya valid jika rangenya diantara 0-23 |
| Menit | ([0-5][0-9]|60) | 01; 02; dst.. | Menit hanya valid jika rangenya diantara 0-60 |
| Separator jam-menit | ([:.]) | Simbol titik pada 23.59 atau simbol titik dua pada 00:00 |  |
| Nomor Handphone | ((?:0|\\+62)\\d{9,11}) | 085794936041; +628155070251 | Mendukung dua jenis nomor handphone, dengan awalan kode negara (+62) dan tanpa awalan kode negara. |
| Pin BB | ([a-f0-9]{8}) | 12AE92 |  |
| Separator tanggal-bulan | [/- ] | Simbol pemisah informasi nama bulan atau nama tanggal |  |

Pada tahap tokenisasi, *tweet* ‘Acara JAKARTA : Gadget Festival 2015 | 9-11 Januari | Marketing Kantor Golf Island Pantai Indah Kapuk.’

Akan dipecah menjadi {Acara, JAKARTA, :, Gadget, Festival, 2015, |, 9, - , 11, Januari, |, Marketing, Kantor, Golf, Island, Pantai, Indah, Kapuk}

1. *Part of Speech* (POS *Tagger)*

*Part of speech* dalam bahasa Indonesia disebut sebagai Kelas Kata. Kelas kata adalah golongan kata dalam satuan bahasa berdasarkan kategori bentuk, fungsi dan makna dalam sistem tata bahasa (Widjono, 2007).

Ada beberapa macam POS Tag, namun yang paling umum digunakan adalah POS Tag Dionysius Thrax. Himpunan POS Tag Dionysisus Thrax terdiri dari 8 kategori yaitu: Noun (Kata benda), Verb (Kata Kerja), Pronoun (Kata Ganti), Preposition(Kata Depan), Adverb (Kata sifat untuk kata kerja), Conjunction (Kata hubung), Particle (Partikel), dan Article (Artikel).

Hasil *tagging* dari modul POS *Tagger* dimanfaatkan oleh modul NER. POS *tag* yang digunakan adalah POS *tag* untuk kata ke-*n*, ke-*n-1* dan ke-*n+1*. Fitur ini digunakan karena dalam beberapa kasus POS Tag suatu kata dapat menunjukan jenis komponen informasi event dalam *tweet*. Sebagai contoh, kata ‘hubungi’ biasa terdapat pada komponen informasi pelengkap event. Kata ‘hubungi’ memiliki kelas kata *verb*(kata kerja).

Menurut Keraf (1984), imbuhan dalam bahasa indonesia dapat menunjukan kelas kata. Sebagai contoh, kata ‘pendaftaran’ memiliki kelas kata *noun* (kata benda) karena memiliki imbuhan ‘pe-an’. Dalam tugas akhir ini POS Tag kata pada *tweet* ditentukan berdasarkan imbuhan.

Setelah melalui proses ini, setiap token akan diberi POS Tag. Menggunakan contoh kasus diatas, hasil pada proses ini adalah sbb:

{Acara/**N**, JAKARTA/**N**, :, Gadget/**N**, Festival/**N**, 2015, |, 9, - , 11, Januari, |, Marketing/**N**, Kantor/**N**, Golf/**N**, Island/**N**, Pantai/**N**, Indah/**Adj**, Kapuk/**N**}

1. *Named Entity Recognition* (NER)

Komponen NER adalah komponen utama dalam arsitektur ini. Modul NER menggunakan algoritma CRF untuk mempelajari named entity dalam *tweet*. Modul ini menggunakan kakas Mallet[[5]](#footnote-5). Pada modul ini digunakan konfigurasi default CRF bawaan Mallet. Struktur model CRF yang digunakan adalah graf *fully conected*.

Dalam contoh di atas, dihasilkan NER berupa,

{Acara/**O**, JAKARTA/**O**, :, Gadget/**I-Name**, Festival/**I-Name**, 2015/**I-Name**, |/**O**, 9/**I-Time**, -/**I-Time** , 11**/I-I-Time**, Januari/**I-Time**, |/**O**, Marketing/**I-Place**, Kantor/**I-Place**, Golf/**I-Place**, Island/**I-Place**, Pantai/**I-Place**, Indah/**I-Place**, Kapuk/**I-Place** }

1. *Temporal resolver.*

Modul *temporal resolver* termasuk dalam tahapan analisis domain seperti penjelasan pada Gambar II‑4. Modul ini dibutuhkan untuk mewakili tahapan *temporal expression analysis*.

# PENGUJIAN DAN ANALISIS

## Tujuan Pengujian

Pada sistem ini dilakukan 2 pengujian untuk setiap modul utama dalam arsitektur sistem ekstraksi informasi. Pengujian pertama untuk modul filter yang memisahkan *tweet* relevan dan yang tidak relevan. Pengujian modul filter dilakukan dengan tujuan untuk mengukur akurasi aturan yang dibuat sehingga dapat ditingkatkan lagi. Pengujian kedua untuk modul *Named Entity Recognition* (NER). Pengujian modul NER bertujuan untuk mengevaluasi pemilihan fitur dan parameter yang tepat untuk melakukan ekstraksi informasi event.

Pengujian untuk modul filter dilakukan pada dataset mentah yang sudah dilabeli data mana yang dianggap relevan data mana yang dianggap tidak relevan. Data yang sudah dilabeli kemudian dicoba dipisahkan dengan aturan dan didapatkan data-data yang dianggap relevan/tidak relevan menurut aturan yang dibuat. Saat ini, perbedaan hasil antara data asli dengan data hasil aplikasi aturan dihitung dengan menggunakan akurasi (Recall).

Pengujian untuk modul NER dilakukan pada dataset yang setiap tokenya sudah dilabeli dengan komponen penyusun informasi event. Setelah dataset tersebut dilabeli dilakukan proses *training* dan evaluasi pada dataset tersebut dengan percentage split sebesar 60%. Metode ini digunakan agar pada pembentukan model data latih tidak tercampur dengan data uji. Evaluasi sistem ekstraksi informasi saat ini menggunakan akurasi (Recall).

## Deskripsi Data Uji

Sebagaimana yang sudah dijelaskan sebelumnya, dataset yang dilakukan pada pengujian ini adalah *tweet* yang sudah dilabeli. Kedua pengujian menggunakan sumber *tweet*  yang sama. Berikut akan dijelaskan mengenai dataset yang digunakan untuk modul filter dan modul NER :

1. Modul Filter

Dataset yang digunakan pada modul filter berasal dari 180 *tweet* yang didapatkan dari hasil crawling pada tanggal tanggal 7 maret 2015 pukul 20:25:26 hingga tanggal 28 maret 2015 pukul 12:13:50. Setiap *tweet* kemudian dilabeli apakah *tweet* tersebut relevan / tidak relevan.

1. Modul NER

Dataset yang digunakan pada modul NER adalah dataset yang dianggap relevan oleh hasil aplikasi aturan filter. Terdapat 111 *tweet* dari 180 *tweet* yang dianggap relevan oleh aturan filter. Meskipun demikian masih teradapat kesalahan klasifikasi relevan/tidak relevan dari 111 *tweet* tersebut sehingga menyisakan 90 *tweet* yang benar-benar relevan.

Kemudian, *tweet* dipecah menjadi token-token dan dilabeli untuk setiap token. Proses tokenisasi menghasilkan 1125 token unik. Label diberikan sesuai dengan *named entity*-nya. Pada sistem ini terdapat 4 named-entity, masing-masing named entity memiliki 1 label. 1 label tambahan digunakan untuk melabeli kata diluar named entity yaitu Other. Contoh pelabelan adalah sbb:

|  |
| --- |
| Acara **O**  JAKARTA **O** : **O**  Gadget **I-Name**  Festival **I-Name**  2015 **I-Name**  | **O**  9 **I-Time**  - **I-Time**  11 **I-I-Time**  Januari **I-Time**  | **O**  Marketing **I-Place**  Kantor **I-Place**  Golf **I-Place**  Island **I-Place**  Pantai **I-Place**  Indah **I-Place**  Kapuk **I-Place** |

## Skenario Pengujian

Secara umum skenario pengujian terdiri dari dua skenario. Skenario pengujian untuk modul filter dan skenario pengujian untuk modul NER. Berikut ini akan dijelaskan mengenai skenario pengujian yang dilakukan

### Skenario Pengujian Modul Filter

Pengujian yang dilakukan untuk modul filter dilakukan dengan beberapa kombinasi rule. Ada beberapa kombinasi rule yang dilakukan pada skenario pengujian filter:

1. Perhitungan karakter setelah eliminasi *mention* dan url.
2. Filter berbasis leksikal (*bag of words*).

Pengujian kinerja aturan filter dilakukan dengan membandingkan hasil aplikasi aturan dari ketiga jenis aturan.

### Skenario Pengujian Modul NER

Pengujian yang dilakukan untuk modul NER dilakukan dengan beberapa kombinasi percobaan. Saat ini kombinasi yang dilakukan baru meliputi variasi praproses dan metode tokenisasi. Kombinasi yang dilakukan diantaranya,

1. Berdasarkan metode tokenisasi
   1. Tokenisasi komponen penunjuk waktu sebagai 1 keutuhan misal jika ditemukan *tweet* ‘Konser Tulus 7maret2015’, sistem menganggap literal ‘7maret2015’ tersebut sebagai 1 token.
   2. Tokenisasi komponen penunjuk waktu sebagai beberapa token, misal pada contoh kasus diatas literal ‘7maret2015’ akan dianggap sebagai 3 buah token, yaitu ‘7’ , ‘maret’ dan ‘2015’.
2. Berdasarkan metode praproses
   1. Tanpa praproses, token yang digunakan adalah token mentah hasil *crawling*. Sebagai contoh, misal token ‘7 maret 2015’ akan dibiarkan sebagaimana aslinya yaitu ‘7 maret 2015’.
   2. Praproses berupa penyederhanaan token waktu. Sebagai contoh misal token ‘7 maret 2015’ akan diubah sebagai token ‘date’.
   3. Praproses berupa penyederhanaan token waktu dan penambahan fitur khusus yang menunjukan variabel waktu.

Pengujian performa dari beberapa set fitur dilakukan dengan membandingkan kombinasi input dari ketiga jenis pengujian. Pada tahap selanjutnya akan dilakukan variasi kombinasi pada parameter yang digunakan untuk melakukan optimasi CRF.

## Hasil Pengujian dan Analisis

Berdasarkan skenario pengujian yang telah dijelaskan sebelumnya, berikut adalah hasil pengujian untuk modul filter dan modul NER.

### Hasil Pengujian dan Analisis Modul Filter

Setelah mengaplikasikan aturan filter pertama didapatkan *confussion* *matrix* seperti berikut:

Tabel IV‑1 *Confussion matrix* setelah dilakukan filter aturan pertama

|  |  |  |
| --- | --- | --- |
| Relevan | Irrelevan |  |
| 85 | 26 | Relevan |
| 28 | 40 | Irrelevan |

Kolom vertikal (warna lebih gelap) menyatakan jumlah *tweet* yang diklasifikasikan relevan/tidak relevan oleh aturan filter. Kolom horizontal (warna putih) menyatakan jumlah *tweet* asli yang relevan. Dari confussion matrix ini didapatkan akurasi sebesar 85 / (85+26) = 76%.

Akurasi yang didapat masih dapat ditingkatkan lagi dengan melakukan iterasi pembuatan rule selanjuntya.

### Hasil Pengujian dan Analisis Modul NER

Setelah dilakukan pengujian NER dengan percentage split 60% didapatkan hasil berupa :

|  |  |
| --- | --- |
| **Jenis pengujian** | **Hasil akurasi** |
| Token bawaan dari *tweet* + fitur | 75% |

akurasi sebesar 75%. Hal ini disebabkan karena jumlah data yang masuk baru 90 *Tweet*. Dibutuhkan lebih banyak *tweet* untuk membuat model yang lebih baik.

DAFTAR PUSTAKA

Aggarwal, C. C. & Zhai, C., 2012. *Mining Text Data.* New York: Springer.

Aggarwal, C. C. & Zhai, C., 2012. *Mining Text Data.* London: Springer.

Axon, S., 2011. *Sprout Social.* [Online]   
Available at: http://sproutsocial.com/insights/twitter-term-definitions/  
[Accessed 12 12 2014].

Boettcher, A. & Lee, D., 2012. *EventRadar : A Real-Time Local Event Detection Scheme Using Twitter Stream.* Besancon, IEEE, pp. 358-367.

Bontcheva, K. et al., 2013. *TwitIE: An Open-Source Information Extraction Pipeline.* Hissar, ACL.

Brenner, J. & Smith, A., 2013. *72% of Online Adults are Social Networking Site Users,* Washington DC: Pew Research Center.

Buckley, C. & Voorhees, E. M., 2000. *Evaluating Evaluation Measure Stability.* New York, s.n.

Clark, A., Fox, C. & Lappin, S., 2010. *The Handbook of Computational Lingguistic and Natural Language Processing.* 1st ed. New Jersey: John Willey.

Cui, A. et al., 2012. *Discover Breaking Events with Popular Hashtags in Twitter.* New York, ACM.

Endarnato, S. K., Pradipta, S., Nugroho, A. S. & Purnama, J., 2011. *Traffic Condition Information Extraction & Visualization from Social Media Twitter for Android Mobile Application.* Bandung, IEEE, pp. 1-4.

Feldman, R. & Sanger, J., 2007. *The Text Mining Handbook.* Cambridge: Cambridge University Press.

Feldman, R. & Sanger, J., 2007. *The Text Mining Handbook.* Cambridge: Cambridge University Press.

Ferenczy, D. v., Spiess, S. & Staudt, A., 2010. *Social Media: How To Get The Most Out Of Twitter To Make Your Event A Success.,* Munich: Amiando.

Hasby, M. & Khodra, M. L., 2013. *Optimal Path Finding Based on Traffic Information Extraction from Twitter.* Jakarta, IEEE, pp. 1-5.

Herbich, R. & Graepel, T., 2010. *Handbook of Natural langguagge processing.* Cambridge: CRC Press.

Jurafsky, D. & Martin, J. H., 2008. *Speech and Language Processing, An Introduction to Natural Language Processing,.* 2nd penyunt. New Jersey: Pearson Prentice Hall.

Khurdiya, A., Dey, L., Mahajan, D. & Verma, I., 2012. *Extraction and Compilation of Events and Sub-events from Twitter.* Macau, IEEE.

Li, F., 2014. *Internet-based Information Extraction Technologies.* [Online]   
Available at: http://www.cs.sjtu.edu.cn/~li-fang/Lecture%206-7%20*Event*%20IE.pdf  
[Accessed 30 December 2014].

Li, R., Lei, K. H., Khadiwala, R. & Chang, K. C.-C., 2012. *TEDAS : a Twitter-based Event Detection and Analysis System.* Washington, DC, IEEE, pp. 1273-1276.

Lukman, E., 2013. *Tech In Asia.* [Online]   
Available at: https://www.techinasia.com/Indonesia-social-jakarta-infographic/  
[Accessed 25 11 2014].

Mani, I. & Wilson, G., 2000. *Robust Temporal Processing of News.* s.l., ACL.

Ritter, A., Clark, S., Mausam & Etzioni, O., 2011. *Named Entity Recognition int tweets : an experimental study.* Stroudsburg, ACL.

Ritter, A., Mausam, Etzioni, O. & Clark, S., 2012. *18th ACM SIGKDD international conference on Knowledge discovery and data mining.* Beijing, ACM.

Sakaki, T. et al., 2012. *Real-time Event Extraction for Driving Information from social sensors.* Bangkok, IEEE.

Sakaki, T., Okazaki, M. & Matsuo, Y., 2010. *Earthquake shakes Twitter users: real-time event detection by social sensors.* New York, USA, ACM, pp. 851-860.

Spless, S. & Alphen-Schrade, M. v., 2013. *Social Media & Events Report,* Munich: Amiando.

Sugitani, T., Shirakawa, M., Hara, T. & Nishio, S., 2013. *Detecting Local Events by Analyzing Spatiotemporal Locality of Tweets.* Barcelona, IEEE, pp. 191-196.

1. SOLR adalah platform pencarian konten berbasis Apache Lucene. [↑](#footnote-ref-1)
2. GATE : General Architecture for Text Engineering, sebuah arsitektur pemrosesan teks yang dikembangkan oleh University of Sheffield. [↑](#footnote-ref-2)
3. Gazeteer adalah kamus geografis berisi informasi suatu lokasi. [↑](#footnote-ref-3)
4. Ark adalah software milik Carneigge Mellon University. Ark dapat diakses pada tautan <Http://ark.cs.cmu.edu> [↑](#footnote-ref-4)
5. Mallet adalah kakas milik Umass. Mallet dapat diunduh pada tautan <http://mallet.cs.umass.edu> [↑](#footnote-ref-5)