

Ekstraksi Intisari Teks *Role Play* dengan *Pattern Matching*

Risalah Widjayanti – 13509028
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13509028@std.stei.itb.ac.id

Abstract—Dalam *role play* berbasis teks, seringkali terdapat bacaan panjang dengan banyak istilah rumit yang membuat pemain lain merasa kesulitan membacanya. Padahal, kebanyakan dari kalimat yang digunakan adalah kalimat tidak perlu yang ditambahkan untuk membuat tulisan menjadi bertele-tele. Untuk kemudahan membaca dan memahaminya, dapat diaplikasikan algoritma *pattern matching* yang dapat mengekstraksi poin penting dalam teks *roleplay*.

Index Terms—*role play, pattern matching, game.*

I. PENDAHULUAN

Role play adalah sejenis permainan gerak yang didalamnya ada tujuan, aturan dan sekaligus melibatkan unsur senang (Jill Hadfield, 1 986). Secara sederhana, *role play* adalah bermain peran. Pemain *role play* akan mengondisikan dirinya sebagai orang lain, baik karakter fiktif rekaannya sendiri, atau meniru tokoh nyata. Lingkup *role play* begitu luas. Misalnya, seni peran (drama) pada dasarnya adalah bentuk *role play*, RPG (*role play game*) yang dimainkan baik *online* maupun *offline* lewat komputer juga merupakan bentuk *role play*. Namun yang akan dibahas dalam makalah ini, adalah *role play* berbasis teks.

Dalam *role play* berbasis teks, setiap pemain melakukan dengan gerakan dengan menuliskan apa yang ia lakukan. Bentuknya seperti cerita bersambung yang setiap orang hanya boleh menggerakkan bagiannya saja. Pemain menuliskan cerita lewat sudut pandang ketiga karakternya lalu pemain lain membalas interaksi dengan menggerakkan karakternya pula membentuk sebuah plot yang utuh. Interaksi saling berbalas lewat cerita itu lah yang menjadi daya tarik *role play* berbasis teks.

Untuk membatasi lingkup, biasanya *role play* memiliki semestanya sendiri. Cerita berlatar belakang apa yang menjadi tema dalam *role play* yang mereka mainkan. Mau tidak mau, cerita yang dituliskan pemain akan bernuansa tema yang mereka sepakati. Biasanya, untuk menentukan tema, diatur oleh forum yang dibuat. Forum *role play* berbasis teks berdiri sendiri-sendiri. Ada yang berbasis cerita populer semacam *Harry Potter* atau *Naruto*, namun bisa juga menggunakan tema original atau tema karangan empunya forum.

Begitu juga dengan perancangan karakter. Karakter

yang dibuat harus memenuhi aturan-aturan tertentu. Ada peraturan yang spesifik untuk setiap forum. Misalnya, penamaan, latar belakang, ras, dan sebagainya. Ada juga aturan umum yang disepakati bersama semisal aturan sebuah karakter tidak boleh menggerakkan karakter milik orang lain atau membaca pikirannya.

Mengenai bahasa yang digunakan, tergantung dari komunitas *role player* itu sendiri. Dalam forum berbahasa Inggris, yang digunakan tentu bahasa Inggris sesuai *grammar* dengan meminimalisasi bahasa *slank*. Demikian juga dengan bahasa Indonesia. Pemain harus menggunakan bahasa Indonesia yang baik dan benar, sesuai dengan EYD (Ejaan Yang Disempurnakan) dan sebisa mungkin menghindari kata asing, bahasa daerah, atau bahasa sehari-hari yang tidak baku. Secara tidak langsung, bermain *role play* membantu meningkatkan kemampuan berbahasa Indonesia.

Isi sebuah teks *role play* sangat beragam. Untuk sebuah postingan yang singkat, biasanya hanya berisi deskripsi gerakan dan dialog yang diutarakan karakternya. Namun, pada perkembangannya, isi yang terdapat dalam sebuah postingan *role play*, dapat terkandung deskripsi berbagai hal. Mulai dari latar tempat, latar waktu, suasana sekeliling, deskripsi isi pikiran, hingga latar belakang cerita yang sebetulnya tidak begitu diperlukan.

Contoh interaksi dalam *roleplay*:

1. Karakter Budi

Pintu ruang kamar kakaknya sedikit terbuka. Budi memberanikan diri menengok ke dalam. Didorongnya sedikit pintu kamar kakaknya dengan teramat pelan. Nyaris tanpa suara. Dengan hati-hati, dijulurkannya kepalanya ke dalam. Jantungnya berdebar keras karena takut ketahuan.

Untungnya, tidak ada siapa-siapa di kamar.

Budi mengembuskan napas lega. Pandangannya mulai menyisir penjuru kamar milik kakaknya. Ada buku-buku yang berserakan di atas lantai dan sebuah botol kosong tergeletak di depan pintu. Ia paham mengapa kakaknya tidak pernah mengizinkan siapa-siapa masuk ke kamarnya.

2. Karakter Susi

Gadis itu melangkah terburu-buru di koridor. Ia lupa mengunci kamarnya karena ada telepon untuknya tadi.

Langkah tergesa-gesanya makin dipercepat begitu melihat seorang lelaki berdiri setengah membungkuk di depan pintu kamarnya yang terbuka.

“BUDIII!” teriaknya keras-keras.

Terang saja ia marah. Ia tidak mengizinkan siapa pun masuk ke kamarnya tapi adiknya begitu lancang mengintip ada apa di dalam. Tangannya menyambar bahu Budi, hendak menariknya kuat-kuat.

3. Karakter Ibu

Perempuan paruh baya itu tengah asyik menyenandungkan lagu Nia Daniati kesukaannya sembari mengelap debu di permukaan cermin. Tiba-tiba didengarnya suara anak perempuannya bergaung dari lantai dua, memekikkan nama adiknya. Dengan tergepoh-gopoh, ia segera menaiki anak tangga untuk melihat apa yang terjadi.

4. Karakter Budi

Bulu tenguknya tiba-tiba saja berdiri. Firasatnya tidak enak. Hanya sepersekian detik setelah itu ia mendengar namanya diteriakkan. Badannya terasa kaku, kepalanya tidak berani ia tolehkan. Bahunya diraih dengan kasar, membuatnya mau tidak mau membalikkan badan.

“K-kak Su-si,” katanya terbata-bata, “a-ampun!”

Karena kemampuan berbahasa seseorang berbeda-beda, tidak semua cerita yang dihasilkan dalam sebuah plot memiliki standar yang sama. Ada perbedaan panjang (jumlah kata), diksi yang digunakan, dan hal teknis lainnya. Hal ini yang kemudian akan menimbulkan permasalahan antara satu pemain dengan pemain lain.

II. PERMASALAHAN

Role play berbasis teks tidak hanya sebagai fasilitator untuk bermain. Secara langsung maupun tidak, fungsinya berubah menjadi fasilitas untuk menunjukkan bakat. Perbedaan kemampuan menulis satu pemain dengan pemain lain dapat terlihat langsung dari bagaimana cara ia menulis cerita.

Pemain yang baru mulai bermain, tak jarang menganggap *role play* tak ubahnya seperti area mengobrol yang mengakibatkan isi postingannya hanya *oneliner* (postingan berisi sebaris). Meskipun hal ini membuat orang lain langsung mengerti apa yang dilakukan karakter miliknya, tetapi hal yang seperti ini adalah hal yang salah kaprah.

Sebaliknya, banyak penulis yang menunjukkan kemampuan menulisnya habis-habisan. Menggunakan kalimat dengan terlalu banyak majas, kalimat dengan bahasa a la karya sastra, diksi yang tidak umum, dan detail lain yang sebetulnya tidak perlu. Tidak jarang membuat sebuah postingan dengan jumlah kata mencapai empat digit karena menganggap kuantitas setara dengan kualitas.

Hal ini menyulitkan pemain yang baru saja mulai belajar. Bahkan, bagi pemain biasa pun hal yang seperti ini kerap menyulitkan. Bukan hanya karena sulit untuk

dimengerti, pemakaian detail yang terlalu melenceng dari poin utama membuat kalimat menjadi tidak efektif. Hasilnya, teks *role play* menjadi sulit untuk dimengerti oleh lawan main.

Padahal, dalam *role play*, interaksi adalah hal yang sangat penting. Seperti yang dicontohkan dalam bab sebelumnya, sebuah plot akan berjalan mulus jika setiap karakter mengerti suasana yang terjadi dan mengerti apa yang dilakukan karakter lain. Bayangkan jika dalam contoh (2) Susi membuat postingan sebagai berikut:

2. Karakter Susi

Gadis enam belas tahun itu mengoordinasikan gerak tungkainya yang jenjang dengan sedemikian rupa sehingga derap langkahnya bersahut-sahutan cepat di udara, sekitar jalanan beralaskan keramik. [...]

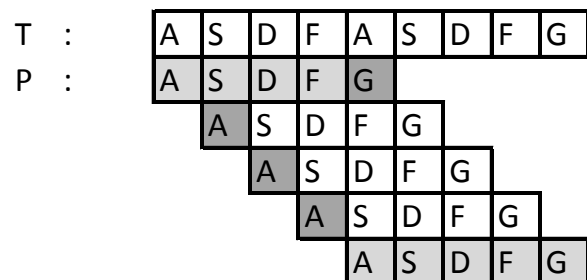
Hal ini menjadi sulit dimengerti padahal bisa langsung dituliskan bahwa ia berjalan dengan terburu-buru di sepanjang koridor. Belum lagi jika seterusnya Susi menggunakan kalimat yang semacam itu. Tentu akan membuat pemain lain kesulitan menemukan inti kalimatnya.

Oleh karena itu, dibutuhkan cara untuk mengekstraksi poin-poin penting dari tesk *role play* yang panjang dan bertele-tele tersebut agar dapat dimengerti.

III. DASAR TEORI

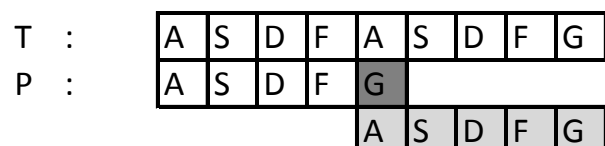
Pattern matching yang diaplikasikan menggunakan algoritma Boyer-Moore.

Algoritma Boyer-Moore adalah algoritma pencocokan string yang memperhatikan susunan dan jumlah karakter. Algoritma ini ditemukan oleh Bob Boyer dan J. Strother Moore dan dipublikasikan di 1977. Boyer-Moore disebut-sebut sebagai algoritma pencarian cepat.



Gambar-1 Pencocokan String dengan Brute Force

Algoritma ini membutuhkan pergeseran sebanyak empat kali sebelum *pattern* yang diinginkan ditemukan dalam teks yang diberikan.



Gambar-2 Pencocokan String dengan Booyer More

Cara kerja algoritma ini adalah sebagai berikut:

1. *Pattern* disejajarkan dengan teks di awal teks.
2. Perbandingan yang dilakukan diawali dari huruf terakhir *pattern*. Misalnya, di contoh, huruf yang dicocokkan adalah huruf A pada teks dengan huruf G pada *pattern*.
3. Karena nilai yang dikembalikan adalah tidak sama, *pattern* harus digeser. Pergeseran *pattern* dengan melihat huruf yang ada pada teks tersebut.
4. Anggap pencocokkan terjadi antara $P[1..n]$ dengan $T[1..m]$. Maka dilihat apakah di antara $T[i]$ hingga $T[m]$ memiliki karakter yang sama dengan yang terkandung pada $P[1]$ sampai dengan $P[n]$. Jika ada, anggaplah berada pada posisi $T[i+k]$ di mana $T[i+k]$ sama dengan $P[1]$. Maka, *pattern* digeser sejauh k supaya $P[1]$ sejajar dengan $T[i+k]$.
5. Jika dalam langkah 4 tersebut menghasilkan tidak adanya karakter yang sama antara $P[1..n]$ dengan $T[1..m]$, pergeseran dilakukan sejauh n .
6. Setelah pergeseran dilakukan, dicocokkan kembali karakter-karakter yang ada, mulai dari $P[n]$ dengan $T[i+k+m]$ lalu bergerak ke kiri.
7. Ulangi langkah tersebut sampai berada di akhir teks dan tidak ditemukan *pattern* atau jika *pattern* ditemukan dalam teks.

Alasan pemilihan algoritma Boyer-Moore untuk diterapkan adalah sebagai berikut:

- Algoritma Boyer-Moore efektif apabila dibandingkan dengan algoritma pencocokan lainnya. *Brute force* tidak mangkus sementara KMP (Knuth-Morris-Pratt) menghasilkan banyak *mismatch* jika ada banyak alfabet. Padahal dalam teks *role play* akan ada banyak alfabet yang dipergunakan.
- Algoritma Boyer-Moore akan berjalan cepat untuk teks dengan jenis karakter yang beragam. Karena yang dibandingkan adalah teks *role play* yang mengandung alfabet beragam dan tanda baca yang juga beragam, sangat cocok untuk menerapkan algoritma Boyer-Moore.

IV. IMPLEMENTASI DAN PENGUJIAN

Implementasi *pattern matching* untuk mengekstraksi teks *role play* tidaklah mudah. Karena luasnya perbendaharaan kata dalam bahasa Indonesia ditambah lagi dengan istilah dalam bahasa Inggris membuat sistem mungkin tidak mengenali kosakata tertentu. Belum lagi, keberadaan majas yang membuat sistem salah mengenali kata yang dimaksud. Hal ini bisa diakali dengan menyaringnya lebih dahulu dan diikutsertakan dalam hasil ekstraksi, namun belum tentu jika diikutsertakan, kalimat yang dihasilkan membentuk sesuatu yang berarti.

Algoritma dalam ekstraksi dapat disederhanakan menjadi berikut ini:

1. Kalimat yang berada di antara tanda petik dua dan diakhiri tanda baca, diambil seluruhnya.
2. Kalimat yang tidak memenuhi kaidah di atas,

dicari pola SPO (subjek, predikat, objek) dalam kalimatnya akan diolah lebih jauh.

3. Kalimat yang tidak memiliki predikat tidak akan diolah lebih jauh.

Untuk pengolahan lebih lanjut, kalimat yang telah hasil dari proses kedua, dicek apa subjeknya dengan algoritma sebagai berikut:

2.1 Jika kata dasar predikatnya adalah kata kerja, kalimat tersebut akan diproses lebih lanjut (kata negasi otomatis diikut sertakan).

2.2 Jika kata dasar predikatnya bukan merupakan kata kerja, kalimat tersebut tidak diproses lagi.

Sekali lagi, kalimat yang lolos dari proses 2.2 di atas, akan mengalami pengolahan berikutnya. Kali ini yang dicek adalah subjek dengan algoritma sebagai berikut:

2.1.1 Kalimat berpredikat pasif yang memiliki objek dan kalimat berpredikat aktif yang berada di awal kalimat, atau yang subjeknya merupakan nama karakternya, kata ganti orang ketiga, referensi yang merujuk pada karakternya, atau anggota badan akan diambil.

2.1.2 Kalimat yang tidak memenuhi kaidah di atas tidak diproses lebih lanjut.

Dengan algoritma seperti ini, sistem memiliki basis data untuk menampung agar sistem mengenali berbagai macam kata. Kata yang telah ada dalam basis data, mampu membedakan apakah kata tersebut kata benda, kata kerja (pasif dan aktif), kata sifat, kata keterangan, atau kata hubung dengan menggunakan prosedur yang mampu mengembalikan tipe kata.

Berdasarkan algoritma di atas, ada beberapa validasi yang diikutsertakan dalam sistem, yakni sebagai berikut:

(1) Validasi SPO. Validasi akan mengembalikan nilai valid jika dalam kalimat terdapat pola seperti berikut ini:

kata benda + predikat + kata benda

kata benda + predikat

predikat + kata benda

Hal ini dikarenakan, dalam bahasa Indonesia, inti dari kalimat adalah SPO. Sementara sisanya adalah pengembangan yang membuat kalimat tersebut menjadi kalimat majemuk atau kalimat dengan keterangan.

(2) Validasi kata predikat. Validasi akan mengembalikan nilai valid untuk predikat yang kata dasar merupakan kata kerja. Untuk menghindari pengikutsertaan kata kerja berawalan ber- yang merupakan keterangan tambahan. Seperti pada kata 'berusia', atau 'berdasar'.

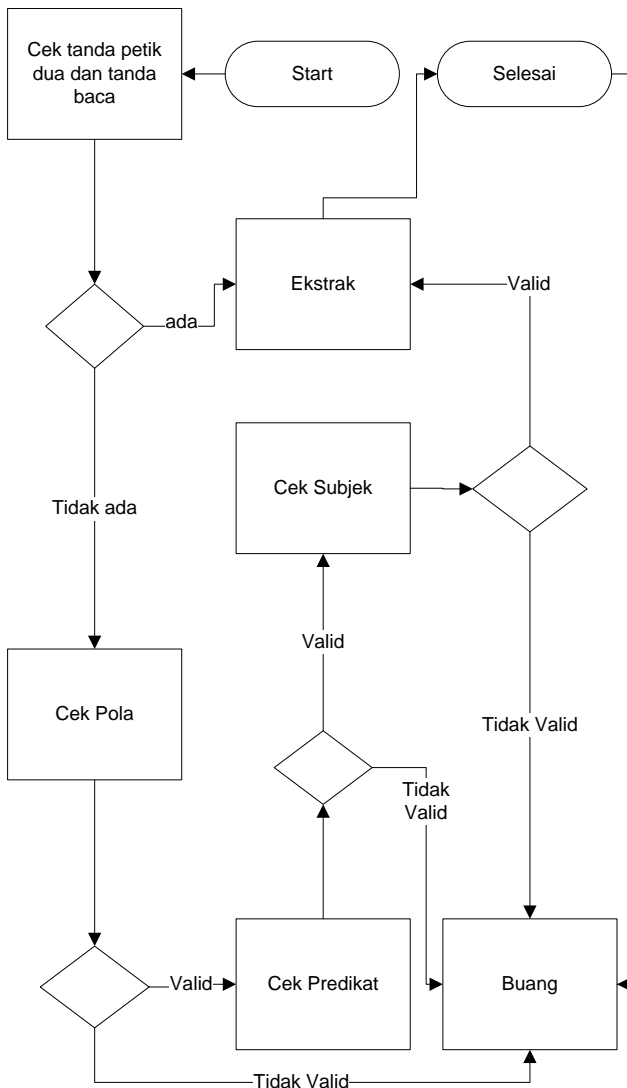
(2) Validasi kata benda. Validasi akan mengembalikan nilai valid jika subjek nama karakternya, kata ganti orang ketiga, referensi yang merujuk pada karakternya, atau anggota badan, atau tidak didahului subjek jika disertai predikat aktif. Predikat pasif juga akan mendapatkan nilai valid jika disertai dengan kata benda. Hal ini untuk menghindari detail sepele yang bukan bagian dari inti teks *roleplay* tersebut.

(4) Validasi tanda baca untuk kalimat dalam tanda petik dua. Validasi akan mengembalikan nilai valid untuk kalimat di antara tanda petik yang diakhiri tanda baca. Hal ini untuk membedakan tanda petik di dalam deskripsi

dengan tanda petik yang menyatakan dialog. Sementara dialog wajib diikutsertakan dalam ekstraksi karena merupakan interaksi yang harus diketahui pemain lain.

melihat pekerjaan rumahnya.

“Apa yang kaulakukan di sini?” tanyanya dengan nada tinggi.



Gambar-3 Diagram Alir Proses Algoritma

Pattern matching yang dilakukan adalah mencocokkan apakah kriteria string yang ada sama dengan string yang tersimpan dalam *pattern*. *Pattern* yang disimpan adalah daftar kata kerja dalam bentuk kata dasar dan kata kerja dasar yang mengalami peluruhan bila diberikan awalan (contoh: kata dasar berawalan k, m, dan t untuk awalan me-). Selain itu *pattern* juga menyimpan kata benda untuk mencocokkan objek dan subjek. Ada *pattern* tersendiri untuk kata benda yang valid bagi objek, yakni kata ganti (ia, dia, dirinya, dan seterusnya), referensi (gadis [itu], lelaki [itu], dan seterusnya), anggota badan (mata[nya], tangan[nya], dan seterusnya), serta nama karakter yang bermain peran.

Contoh teks *role play* yang dimasukkan:

Susi murka sekali atas kejadian ini. Dulu, Brandon menjahatnya dan membuatnya pulang dengan cucuran air mata darah hanya karena Susi tidak mengizinkannya

Dengan menggunakan ekstraksi yang mengimplementasikan *pattern matching*, kalimat tersebut akan diproses sebagai berikut:

1. **Susi murka** sekali atas kejadian ini.

“Susi” valid untuk subjek dan “murka” valid untuk predikat”.

2. Dulu, Brandon menjahatnya dan membuatnya pulang dengan cucuran air mata darah hanya karena Susi tidak mengizinkannya melihat pekerjaan rumahnya.

Kalimat memenuhi SPO namun subjek yang ada adalah Brandon yang tidak ada sangkut pautnya dengan plot utama.

3. **“Apa yang kaulakukan di sini?”** tanyanya dengan nada tinggi.

Kalimat yang berada di antara tanda petik langsung diikutsertakan dalam hasil ekstraksi.

Dari teks *role play* di atas, keluaran yang dihasilkan adalah:

Susi murka.

“Apa yang kaulakukan di sini?”

Dengan hasil ekstraksi yang seperti itu, mudah dimengerti apa yang terjadi pada Susi dan bagaimana interaksinya dengan Budi (jika melanjutkan cerita pada bagian I). Pemain lawan tidak perlu membaca deskripsi bertele-tele dan kilas balik yang tidak dibutuhkan dalam cerita kali ini.

V. ANALISIS

Sistem cukup membantu dalam menyelesaikan masalah menyaring tulisan yang bertele-tele. Namun, pada tipe *role play* lain, yang sering bermain dengan struktur kalimat, tentu akan sulit digunakan dengan baik.

Pada dasarnya, masalah ekstraksi tidak bisa dilakukan tanpa mempertimbangkan makna utuh suatu kalimat. Apa yang dilakukan sistem hanyalah membantu mereka yang kesulitan membaca ratusan bahkan ribuan kata untuk cerita yang bertele-tele. Selain itu, *pattern* harus seluas mungkin kosakatanya karena banyaknya diksi yang bersinonim. Contoh sederhana, untuk subjek pada aturan ‘referensi yang merujuk pada pemain’, kreativitas penulis tidak terbatas pada penggunaan kata ‘gadis’, ‘pemuda’, ‘anak’, dan sebagainya. Bisa jadi kata yang digunakan seperti ‘Si ikal atau berikutnya ‘sang calon insinyur’ dan sebagainya yang juga merujuk pada si pemain.

Selain itu, karena yang disaring hanyalah pola-pola kalimat SPO, keluaran yang dihasilkan masih dalam bentuk kasar. Pengguna perlu memikirkan makna yang mungkin dari hasil keluaran yang terpotong. Solusinya, penambahan kata hubung dan perbaikan pola dengan menambahkan aturan mengenai keterangan.

Sistem masih membutuhkan banyak penyempurnaan mulai dari logika, struktur, tambahan basis data, dan lain-

lain untuk dapat benar-benar digunakan sebagai ekstraksi teks *role play*. Namun, untuk saat ini, ekstraksi yang dilakukan sudah cukup membantu.

VI. KESIMPULAN

1. Untuk algoritma *pattern matching*, salah satu algoritma yang paling baik adalah Boyer-Moore yang mangkus dan efektif.
2. Implementasi algoritma *pattern matching*—dalam hal ini Boyer-Moore—dapat digunakan untuk melakukan ekstraksi teks *role play*.
3. Untuk ekstraksi pada teks naratif berbahasa Indonesia, daftar *pattern* yang disediakan harus cukup besar karena kosakata bahasa Indonesia yang luas dan penggunaan diksi yang bisa bermacam-macam.
4. Struktur bahasa Indonesia sulit untuk dipolakan dan disaring dengan *pattern matching*. Perlu banyak aturan kondisional untuk menyempurnakan agar cocok dengan kaidah bahasa Indonesia.

REFERENCES

- [1] Munir, Rinaldi. 2005. Diktat Kuliah Strategi Algoritmik IF2251 Strategi Algoritmik. Departemen Teknik Informatika ITB
- [2] Slide Pattern Matching Kuliah IF3051 Strategi Algoritma Semester 1 Tahun Pelajaran 2012/2013

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Desember 2012
ttd



Risalah Widjayanti
13509028