

Penerapan Algoritma Pencocokan String dalam Translator Bahasa Pascal ke C

Randi Chilyon Alfianto - 13513087
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
randi_chilz@yahoo.com

Abstract—Translator merupakan solusi otomatis untuk menerjemahkan suatu kode program dari suatu Bahasa pemrograman ke Bahasa pemrograman lainnya. Translator Bahasa Pascal ke Bahasa C bekerja dengan melakukan parsing terhadap kode program, kemudian diterapkan aturan penerjemahan. Dengan menggunakan translator ini, pengguna dapat menghemat waktu yang diperlukan untuk membuat suatu program yang sama ke dalam dua Bahasa pemrograman yang berbeda. Untuk menerapkan aturan penerjemahan yang ada, diperlukan pencocokan dari Bahasa Pascal ke Bahasa C. Dengan menggunakan algoritma pencocokan string, dimungkinkan dilakukan proses penerjemahan dari Bahasa Pascal ke Bahasa C. Karena terdapat kesamaan antara Bahasa Pascal dan Bahasa C, maka kedua Bahasa tersebut dimungkinkan dilakukan penerjemahan secara akurat. Bahasa C didisain untuk digunakan oleh pemula.

Keywords—translator, pascal, c, string matching, programming, source code.

I. PENDAHULUAN

Proses penerjemahan suatu kode pemrograman dari satu Bahasa pemrograman ke Bahasa lain seringkali menjadi kendala yang dihadapi oleh programmer. Beberapa alasan yang melatarbelakangi hal ini antara lain : kurangnya penguasaan terhadap Bahasa yang diinginkan, waktu yang dibutuhkan untuk melakukan penerjemahan, kurangnya referensi yang dibutuhkan, dan lain-lain. Kebutuhan akan proses penerjemahan kode pemrograman sering terjadi karena alasan bisnis, integrasi, regulasi perusahaan, serta adaptasi terhadap teknologi yang terus berkembang seiring berjalannya waktu.

Proses penerjemahan suatu kode program ke Bahasa pemrograman lain bukanlah hal yang mudah, terlebih jika kode sumber yang ingin diterjemahkan merupakan suatu program yang kompleks dan besar. Apabila terjadi hal demikian, proses penerjemahan tersebut akan memerlukan waktu yang panjang dan juga tenaga ahli yang tidak sedikit serta rentan terhadap kesalahan. Oleh karena itu, diperlukan suatu program yang dapat digunakan untuk melakukan hal tersebut. Program tersebut diharapkan dapat menerjemahkan suatu kode program ke Bahasa

pemrograman lain yang dikehendaki. Program tersebut dikenal sebagai Translator.

Saat ini, masih banyak kegiatan akademik di bidang Informatika yang masih menggunakan Bahasa Pascal sebagai Bahasa pemrograman pertama yang diajarkan (salah satunya adalah STEI ITB). Pascal dipilih karena keluwesan sintaks serta kelebihanannya yang mirip dengan Bahasa manusia sehingga lebih mudah dipahami. Selain Bahasa Pascal, Bahasa pemrograman lain yang juga sering digunakan adalah Bahasa C. Dalam Bahasa C, lebih banyak digunakan simbol-simbol dibandingkan dengan Bahasa Pascal. Bahasa C mendorong para pengguna untuk lebih kreatif dalam menyusun algoritma, karena Bahasa C memiliki kompleksitas yang lebih rumit apabila dibandingkan dengan Bahasa Pascal.

Bahasa Pascal dan Bahasa C memiliki banyak kesamaan, di antaranya : menganut paradigma procedural, merupakan seri lanjutan dari Bahasa ALGOL, waktu munculnya hampir bersamaan, dan lain-lain. Bahasa C memiliki sifat robust, fleksibel, efisien, ekspresif, dan permisif, sedangkan Bahasa Pascal memiliki sintaks yang jelas dan mudah dipahami.

Penerjemahan pada beberapa aspek Bahasa Pascal ke Bahasa C tidak dapat dilakukan secara straight-forward (kata per kata), namun perlu dilakukan proses pencocokan sintaks untuk menyesuaikan lingkungan dari Bahasa Pascal dengan Bahasa C sehingga dapat menghasilkan hasil terjemahan yang baik. Oleh karena itu, diperlukan suatu algoritma yang dapat mencocokkan sintaks tersebut dengan baik. Dalam hal ini, salah satu algoritma pencocokan string, yaitu algoritma Boyer-Moore merupakan salah satu pilihan yang diharapkan dapat menghasilkan hasil terjemahan yang baik dan maksimal.

II. LANDASAN TEORI

A. Algoritma

Algoritma adalah urutan langkah-langkah untuk memecahkan suatu masalah. Definisi lain dari algoritma adalah deretan langkah-langkah komputasi yang mentransformasikan data masukan menjadi keluaran. Algoritma dapat dituliskan dalam berbagai notasi,

misalnya dalam notasi kalimat-kalimat deskriptif. Sebagai contoh, misalkan kita akan menuliskan algoritma untuk mencari elemen terbesar dari sebuah senarai S yang beranggotakan n buah bilangan bulat. Bilangan-bilangan bulat di dalam senarai dinyatakan sebagai S_1, S_2, \dots, S_n . Elemen terbesar akan disimpan di dalam peubah (variable) yang bernama maks.

Berikut adalah contoh algoritma untuk menentukan Elemen terbesar dari suatu senarai :

1. Asumsikan S_1 sebagai elemen terbesar sementara. Simpan S_1 ke dalam maks.
2. Bandingkan nilai maks dengan elemen S_2 . Jika elemen S_2 lebih besar daripada maks, maka nilai maks diganti dengan S_2 .
3. Ulangi langkah 2 untuk elemen-elemen berikutnya.
4. Berhenti jika tidak ada lagi elemen yang tersisa di dalam senarai. Dalam hal ini, maks berisi nilai dari elemen terbesar.

Selain notasi deskriptif, algoritma juga dapat digambarkan dalam notasi Bahasa pemrograman, misalnya Pascal, C, C++, dan lain-lain. Setiap Bahasa pemrograman memiliki aturan sintaks yang rumit yang mengakibatkan algoritma tersebut menjadi lebih sulit untuk dipahami oleh manusia. Padahal, suatu algoritma pada hakekatnya berisi abstraksi dari model penyelesaian masalah, sehingga algoritma seharusnya dibebaskan dari hal-hal teknis yang tidak perlu.

Para ilmuwan komputer lebih menyukai menuliskan algoritma dalam notasi yang lebih praktis, yaitu notasi *pseudo-code*. *Pseudo* memiliki arti semu atau tidak sebenarnya. *Pseudo-code* adalah notasi yang mirip dengan notasi Bahasa pemrograman tingkat tinggi, seperti Bahasa Pascal. Tidak seperti Bahasa pemrograman yang terikat dengan berbagai sintaks yang merepotkan (seperti : titik koma(;), indeks, format keluaran, kata-kata khusus, dan sebagainya), *pseudo-code* dapat diterima asalkan perintahnya tidak membingungkan pembaca. Keuntungan menggunakan notasi *pseudo-code* adalah kemudahan mengkonversi atau mentranslasi ke Bahasa pemrograman.

B. Bahasa Pemrograman

Bahasa pemrograman (programming language) adalah sebuah instruksi standar untuk memerintah komputer agar mempunyai fungsi tertentu. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan atau diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

Fungsi bahasa pemrograman yaitu memerintah komputer untuk mengolah data sesuai dengan alur berpikir yang kita inginkan. Keluaran dari bahasa pemrograman tersebut berupa program/aplikasi. Contohnya adalah program yang digunakan oleh kasir

di mall-mall atau swalayan, penggunaan lampu lalu lintas di jalan raya, dll.

Bahasa Pemrograman yang kita kenal ada banyak sekali di belahan dunia, tentang ilmu komputer dan teknologi dewasa ini. Perkembangannya mengikuti tingginya inovasi yang dilakukan dalam dunia teknologi. Contoh bahasa pemrograman yang kita kenal antara lain adalah untuk membuat aplikasi game, antivirus, web, dan teknologi lainnya.

Bahasa pemrograman komputer yang kita kenal antara lain adalah Pascal, Java, Visual Basic, C++, C, Cobol, PHP, .Net, dan ratusan bahasa lainnya. Namun tentu saja kebutuhan bahasa ini harus disesuaikan dengan fungsi dan perangkat yang menggunakannya.

Secara umum bahasa pemrograman terbagi menjadi 4 kelompok, yaitu :

1. Object Oriented Language (Visual dBase, Visual FoxPro, Delphi, Visual C)
2. High Level Language (seperti Pascal dan Basic)
3. Middle Level Language (seperti bahasa C)
4. Low Level Language (seperti bahasa Assembly)

C. Bahasa Pascal

Pascal adalah bahasa pemrograman yang pertama kali di buat oleh Profesor Niklaus Wirth, seorang anggota International Federation of Information Processing (IFIP) pada tahun 1971. Dengan mengambil nama dari matematikawan Perancis, Blaise Pascal, yang pertama kali menciptakan mesin penghitung, Profesor Niklaus Wirth membuat bahasa Pascal ini sebagai alat bantu untuk mengajarkan konsep pemrograman komputer kepada mahasiswanya. Selain itu, Profesor Niklaus Wirth membuat Pascal juga untuk melengkapi kekurangan-kekurangan bahasa pemrograman yang ada pada saat itu.

Kelebihan dari bahasa pemrograman Pascal adalah:

1. Tipe Data Standar, tipe-tipe data standar yang telah tersedia pada kebanyakan bahasa pemrograman. Pascal memiliki tipe data standar: boolean, integer, real, char, string,
2. User defined Data Types, programmer dapat membuat tipe data lain yang diturunkan dari tipe data standar.
3. Strongly-typed, programmer harus menentukan tipe data dari suatu variabel, dan variabel tersebut tidak dapat dipergunakan untuk menyimpan tipe data selain dari format yang ditentukan.
4. Terstruktur, memiliki sintaks yang memungkinkan penulisan program dipecah menjadi fungsi-fungsi kecil (procedure dan function) yang dapat dipergunakan berulang-ulang.
5. Sederhana dan Ekspresif, memiliki struktur yang sederhana dan sangat mendekati bahasa manusia (bahasa Inggris) sehingga mudah dipelajari dan dipahami.

D. Bahasa C

Bahasa pemrograman C merupakan salah satu bahasa pemrograman komputer. Dibuat pada tahun

1972 oleh Dennis Ritchie untuk Sistem Operasi Unix di Bell Telephone Laboratories. Meskipun C dibuat untuk memprogram sistem dan jaringan komputer namun bahasa ini juga sering digunakan dalam mengembangkan software aplikasi. C juga banyak dipakai oleh berbagai jenis platform sistem operasi dan arsitektur komputer, bahkan terdapat beberapa compiler yang sangat populer telah tersedia. C secara luar biasa memengaruhi bahasa populer lainnya, terutama C++ yang merupakan ekstensi dari C.

Bahasa C banyak dipakai untuk :

1. Membuat sistem operasi dan program-program sistem.
2. Pemrograman yang berhubungan dengan perangkat keras (misalnya untuk control peralatan).
3. Membuat tool kit.
4. Menulis program aplikasi.

Kelebihan dari Bahasa C, sehingga terpilih untuk aplikasi-aplikasi di atas adalah kemampuannya untuk membuat kode yang compact, efisien tanpa perlu mengorbankan readability-nya.

E. *String Matching*

String matching adalah pencarian sebuah pattern pada sebuah teks (Ronald L. Rivest dkk. 1994). Prinsip kerja algoritma string matching adalah sebagai berikut:

1. Memindai teks dengan bantuan sebuah window yang ukurannya sama dengan panjang pattern.
2. Menempatkan window pada awal teks.
3. Membandingkan karakter pada window dengan karakter dari pattern. Setelah pencocokan (baik hasilnya cocok atau tidak cocok), dilakukan shift ke kanan pada window. Prosedur ini dilakukan berulang-ulang sampai window berada pada akhir teks. Mekanisme ini disebut mekanisme sliding-window.

Algoritma string matching mempunyai tiga komponen utama, yaitu:

1. Pattern, yaitu deretan karakter yang akan dicocokkan dengan teks, dinyatakan dengan $x[0..m-1]$, panjang pattern dinyatakan dengan m .
2. Teks, yaitu tempat pencocokan pattern dilakukan, dinyatakan dengan $y[0..n-1]$, panjang teks dinyatakan dengan n .
3. Alfabet, yang berisi semua simbol yang digunakan oleh bahasa pada teks dan pattern, dinyatakan dengan Σ dengan ukuran dinyatakan dengan $|\Sigma|$.

F. *Algoritma Boyer-Moore*

Algoritma Boyer-Moore termasuk algoritma string matching yang paling efisien dibandingkan algoritma-algoritma string matching lainnya (Ronald L. Rivest dkk. 1994). Karena sifatnya yang efisien, banyak dikembangkan algoritma string matching dengan bertumpu pada konsep algoritma Boyer-Moore, beberapa di antaranya adalah algoritma Turbo BM dan algoritma Quick Search.

Algoritma Boyer-Moore menggunakan metode pencocokan string dari kanan ke kiri yaitu memindai karakter pattern dari kanan ke kiri dimulai dari karakter

paling kanan. Algoritma Boyer-Moore menggunakan dua fungsi shift yaitu good-suffix shift dan bad-character shift untuk mengambil langkah berikutnya setelah terjadi ketidakcocokan antara karakter pattern dan karakter teks yang dicocokkan.

G. *Stack*

Stack adalah suatu bentuk khusus dari linear list (suatu struktur data yang merupakan himpunan terurut yang dapat berkurang atau bertambah setiap saat) di mana operasi penyisipan dan penghapusan atas elemen-elemennya hanya dapat dilakukan pada satu sisi saja yang disebut sebagai "TOP". Ada empat operasi dasar yang didefinisikan pada stack, yaitu:

1. Create, operator ini berfungsi untuk membuat sebuah stack kosong.
2. IsEmpty, operator ini berfungsi untuk menentukan apakah suatu stack adalah stack kosong.
3. Push, operator ini berfungsi untuk menambahkan satu elemen ke dalam stack.
4. Pop, operator ini berfungsi untuk mengeluarkan satu elemen dari dalam stack.

Dalam penggunaannya, untuk menempatkan stack biasanya digunakan sebuah array. Tetapi perlu diingat di sini bahwa stack dan array adalah dua hal yang berbeda. Selain itu penggunaan stack dengan array dirasakan kurang tepat. Penggunaan stack pada pembangunan Translator ini menggunakan record sebagai implementasi stack.

H. *Tree / Pohon*

Tree/pohon merupakan struktur data yang tidak linear/non linear yang digunakan terutama untuk merepresentasikan hubungan data yang bersifat hierarkis antara elemen-elemennya. Kumpulan elemen yang salah satu elemennya disebut dengan root (akar) dan sisa elemen yang lain disebut sebagai simpul (node/vertex) yang terpecah menjadi sejumlah himpunan yang tidak saling berhubungan satu sama lain, yang disebut subtree/cabang.

Sebuah pohon biner T dapat didefinisikan sebagai sekumpulan terbatas dari elemen-elemen yang disebut nodes/simpul di mana :

1. T dikatakan kosong (disebut null tree/pohon null atau empty tree/pohon kosong)
2. T terdiri dari sebuah node khusus yang dipanggil R , disebut root dari T dan node-node T lainnya membentuk sebuah pasangan terurut dari binary tree T_1 dan T_2 yang tidak berhubungan yang kemudian dipanggil subtree kiri dan subtree kanan.

Jika T_1 tidak kosong maka rootnya disebut successor kiri dari R dan jika T_2 tidak kosong, maka rootnya disebut successor dari R . Berikut adalah beberapa jenis dari binary tree :

1. Complete Binary Tree

Suatu binary tree T akan disebut complete/lengkap jika semua levelnya memiliki child 2 buah kecuali untuk level paling akhir. Tetapi pada akhir level setiap leaf/daun muncul terurut dari sebelah kiri.

2. Extended Binary Tree : 2-Tree

Sebuah binary tree dikatakan 2-tree atau extended binary tree jika tiap simpul N memiliki 0 atau 2 anak. Simpul dengan 2 anak disebut dengan simpul internal (internal node), dan simpul dengan 0 anak disebut dengan external node. Kadang-kadang dalam diagram node-node tersebut dibedakan dengan menggunakan tanda lingkaran untuk internal node dan kotak untuk eksternal node.

III. ANALISIS MASALAH

Metodologi penelitian yang digunakan dalam penelitian ini menggunakan metode studi kepustakaan, di mana studi kepustakaan yang dilakukan dalam penelitian ini bertujuan untuk mencari sumber data sekunder yang akan mendukung penelitian dan untuk mengetahui sampai ke mana ilmu yang berhubungan dengan penelitian terkait dengan algoritma string matching telah berkembang, sampai ke mana terdapat kesimpulan dan degener-alisasi yang pernah dibuat.

Langkah yang dilakukan dalam studi kepustakaan yaitu mengidentifikasi teori secara sistematis, penemuan pustaka, dan analisis dokumen yang memuat informasi yang berkaitan dengan Algoritma String Matching.

Idealnya, suatu teks akan dikenali sebagai pernyataan Pascal jika teks tersebut telah sukses melalui proses compile pada Turbo Pascal 7.0. Pernyataan Pascal tidak bisa secara langsung diterjemahkan ke bentuk C. Proses penerjemahan pun harus memiliki kualifikasi yang baik dari sudut lama waktu proses yang dibutuhkan. Maka harus diterapkannya beberapa fungsi pengenalan, diantaranya:

1. Penyesuaian algoritma Boyer-Moore pada setiap kasus pencocokan.
2. Proses pencocokan string terhadap teks, dimaksudkan agar setiap kata dapat dikenali oleh translator dan dapat diterjemahkan.
3. Aturan terjemahan, dimaksudkan sebagai aturan yang diterapkan jika ditemukannya sebuah bentuk pernyataan di dalam teks.
4. Mengenali jenis ekspresi yang dioperasikan dan menemukan bentuk hasil operasi dari Pascal ke C.
5. Proses handling untuk setiap bentuk pernyataan yang tidak memiliki aturan penerjemahan.
6. Metode pemberian pesan kesalahan.

IV. IMPLEMENTASI

Penggunaan stack dengan single link list dimaksudkan sebagai manajemen memori pada runtime Translator. Single link list difungsikan sebagai media penyimpanan data sementara hasil proses algoritma Boyer-Moore. Di mana stack memiliki nilai batas penyimpanan yang dinamis. Sehingga dapat memaksimalkan alokasi memori terhadap kebutuhan Translator yang sedang berjalan. Di Visual Basic 6.0, stack dengan single link list dapat diimplementasikan sebagai collection control.

Representasi metode pemanfaatan collection untuk mendukung kinerja Translator dengan diasumsikan BmBc sebagai class module, Pos dan Char sebagai elemen class module.

Sama halnya dengan implementasi stack, binary tree difungsikan sebagai solusi pemecahan hirarki operator pada sebuah ekspresi. Pada implementasinya, binary tree hanya digunakan sebagai tempat penyimpanan sementara dari potongan-potongan teks yang dikenali sebagai sebuah ekspresi. Berikut algoritma penerapan binary tree pada translator:

1. Lakukan pencarian posisi dengan menggunakan prosedur BM untuk 2 kata kunci pada teks. Kata kunci-1 merupakan kata kunci dengan arti buka (tanda yang menyebutkan sebuah awalan) dan Kata kunci-2 merupakan kata kunci dengan arti tutup (tanda yang menyebutkan sebuah akhiran). Contoh, kata kunci-1 : “(“ dan kata kunci-2 : “)”
2. Jika jumlah posisi kedua kata kunci yang ditemukan tidak sama (tidak sepasang) maka berikan pesan kesalahan.
3. Jika jumlah posisi kedua karakter sama, maka bandingkan antara posisi paling pertama dari kata kunci-1 terhadap posisi paling pertama dari kata kunci-2 pada teks.
4. Jika kata kunci-1 memiliki nilai perbandingan terbesar, maka proses ulang (rekursif) teks dengan menambahkan kata kunci awalan dan akhiran pada kedua ujung teks.
5. Tambahkan setiap nilai posisi yang ditemukan ke dalam binary tree. Binary tree pada posisi ini merupakan binary tree yang tersusun dengan 1 subtree kanan, 0 subtree kiri. Difungsikan untuk memudahkan dalam pengurutan node.
6. Urutkan setiap posisi yang berkesesuaian (pasangan) antara posisi kata kunci-1 dan 2 dengan menggunakan metode pengurutan Pre-Order. Pre-Order dimaksudkan agar pengolahan operasi terjadi dari posisi buka-tutup terluar (hirarki terkecil). Langkah Pre-Order dilakukan dengan mengunjungi root Binary Tree, mengunjungi subtree kiri, lalu subtree kanan secara rekursif.
7. Pada setiap proses pengunjungan, dilakukan perubahan bentuk pada binary tree hingga menjadi sebuah binary tree transversal dengan Pre-Order sebagai tolok ukur struktur tree.
8. Setelah posisi-posisi kata kunci-1 dan 2 telah tersusun sebagaimana semestinya, terjemahkan bagian teks yang berada pada posisi yang berkesesuaian terhadap masing-masing posisi pada binary tree. Proses pengunjungan subtree dilakukan dengan metode Post-Order secara rekursif, dengan tujuan agar teks dapat terpecah dari bagian paling dalam (hirarki terbesar).

Fungsi utama pada Translator terhadap teks yang dibentuk menjadi pernyataan atau serangkaian pernyataan Pascal adalah pencocokan suatu pattern tertentu yang diasumsikan sebagai pemisah rangkaian pembentuk pernyataan Pascal. Pencocokan ini dilakukan untuk memenggal teks berdasarkan posisi pattern terhadap teks, sehingga teks dapat dikenali sebagai pernyataan Pascal

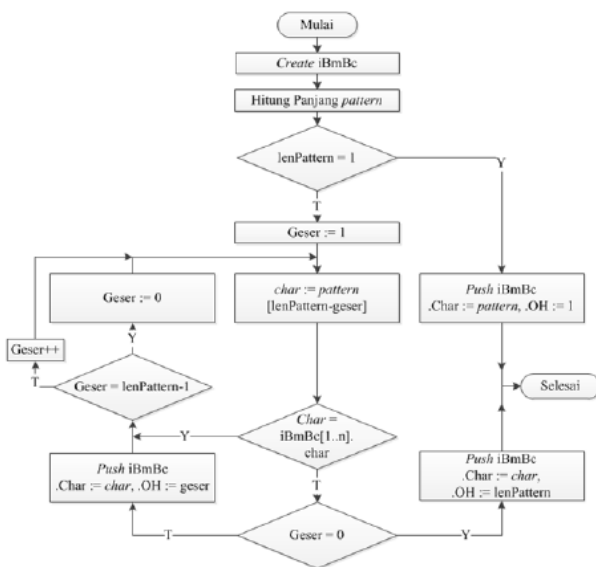
dan diterjemahkan ke pernyataan C.

Kaidah pertama dari algoritma Boyer-Moore ini adalah masing-masing karakter dari pattern pencocokan harus mempunyai kode ASCII yang sama terhadap target karakter yang ada pada teks serta karakter-karakter antara pattern dan target harus terurut sama persis.

Kaidah yang kedua adalah hasil penerjemahan teks berdasarkan pencocokan pattern akan bernilai benar secara utuh jika teks telah sukses melalui proses compile pada compiler Turbo Pascal 7.0. Walaupun demikian, Translator menerapkan fungsi-fungsi yang dapat memastikan kebenaran sintaks pernyataan berdasarkan perolehan posisi pattern terhadap teks dan mengabaikan beberapa kesalahan sederhana pada sintaks sehingga dapat diterjemahkan ke bentuk C. Dengan kata lain, suatu teks memungkinkan diterjemahkan secara utuh, sebagian, atau tidak sama sekali. Untuk itu, dibutuhkan suatu basis pengetahuan dalam pencocokan posisi dan pemenggalan pattern terhadap teks.

Secara Umum, langkah-langkah dari teknik penerapan Boyer-Moore pada pencocokan pattern pada teks dapat direpresentasikan sebagai berikut :

1. Menjalankan prosedur preBmBc dan preBmGs untuk mendapatkan nilai perbandingan pergeseran.
2. Menjalankan prosedur preBmBc. Fungsi dari prosedur ini adalah untuk menentukan berapa besar pergeseran yang dibutuhkan untuk mencapai karakter tertentu pada pattern dari karakter pattern terakhir/terkanan. Hasil dari prosedur preBmBc disimpan pada stack bmBc.

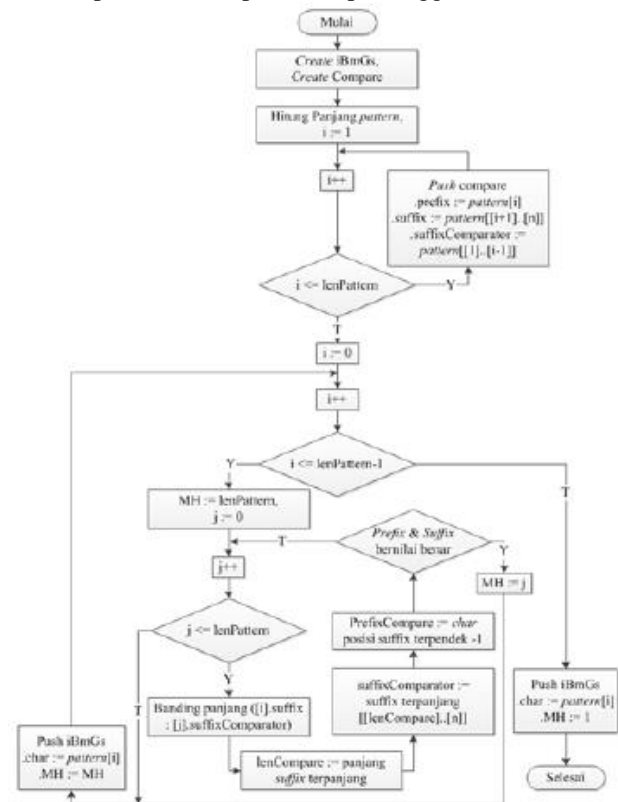


Gambar 1. Flowchart Prosedur preBmBc

3. Menjalankan prosedur preBmGs. Sebelum menjalankan isi prosedur ini, prosedur suffix dijalankan terlebih dulu pada pattern. Fungsi dari prosedur suffix adalah memenggal sejumlah karakter yang dimulai dari karakter terakhir/terkanan dengan sejumlah karakter yang dimulai dari setiap karakter yang lebih kiri. Hasil dari prosedur suffix disimpan pada stack suff. Dengan demikian suff[i] mencatat panjang dari suffix yang cocok dengan segmen dari pattern yang di akhiri karakter ke-i.

4. Dengan prosedur preBmGs, dapat diketahui berapa banyak langkah pada pattern dari sebuah segmen ke segmen lain yang sama, di mana letaknya lebih kiri dengan karakter di sebelah kiri segmen yang berbeda. Prosedur preBmGs menggunakan stack suff untuk mengetahui semua pasangan segmen yang sama.

Menjalankan prosedur BM, proses pencocokan pattern dengan menggunakan hasil dari prosedur preBmBc dan preBmGs yaitu stack bmBc dan bmGs sebagai landasan keputusan pergeseran. Proses pengambilan nilai BmBc dan BmGs pada database dimaksudkan agar proses perhitungan nilai keputusan pergeseran oleh prosedur BmBc dan BmGs dapat menjadi lebih optimal, karena nilai keputusan telah tersimpan untuk beberapa pattern dengan nilai ketetapan. Namun proses filter pada database (recordset) membutuhkan estimasi waktu tertentu. Maka jika dibandingkan proses pencocokan nilai keputusan oleh prosedur preBmBc dan preBmGs untuk pattern yang lebih pendek, proses cek database akan menjadi lebih lama. Untuk itu, keputusan pengecekan database ditentukan dari parameter prosedur BM pada saat pemanggilan.



Gambar 2. Flowchart Algoritma Boyer-Moore

Terdapat pula parameter plusPos yang berfungsi sebagai nilai tambah untuk setiap posisi yang ditemukan. plusPos digunakan pada saat pencocokan string yang membutuhkan posisi akhir, tengah, maupun posisi tertentu dari sebuah karakter yang terdapat pada pattern terhadap teks.

Selain itu, terdapat pula kriteria jumlah pencocokan, kriteria ini difungsikan sebagai parameter jumlah pencocokan yang dibutuhkan. Sebagai contoh, jika terdapat 10 kecocokan, maka pencocokan akan dihentikan

ketika mencapai kecocokan ke-5, di mana 5 adalah kriteria jumlah pencocokan. Sehingga menjadikan algoritma Boyer-Moore berjalan lebih optimal dan sesuai kebutuhan Translator.

Basis pengetahuan algoritma Translator Pascal ke C dibangun oleh 2 dasar sintaks seperti yang telah dijelaskan sebelumnya. Berikut representasi sintaks pemenggalan teks berdasarkan sintaks pernyataan Pascal. Masing-masing pernyataan memiliki alur yang bernilai tetap. Alur-alur tersebut membentuk sebuah pola pembentukan pernyataan yang dapat dimanfaatkan oleh algoritma Translator untuk mengkondisikan kriteria pencocokan pattern algoritma Boyer-Moore terhadap teks. Terlihat pula bahwa suatu pernyataan dapat memiliki badan berupa pernyataan lainnya, dan tentunya hal ini berdasarkan pengkondisian pernyataan induk. Maka dengan mengaitkan per pola variasi pernyataan Pascal terhadap C, didapatkan suatu aturan baru mengenai pemenggalan dan penggabungan hasil dari pencocokan algoritma Boyer-Moore terhadap teks.

Dengan pemanfaatan Database Microsoft Access 2003 sebagai media penyimpanan data, pengkondisian pola sintaks dirancang dan disimpan berdasarkan keterkaitan yang telah dijelaskan pada pembahasan sebelumnya. Sehingga dapat digunakan oleh Translator sebagai basis pengetahuan pemberian pattern pencocokan algoritma Boyer-Moore dan memvalidasi nilai posisi yang dihasilkan. Sehingga didapatkan nilai posisi pemenggalan suatu teks berdasarkan sintaks Pascal, dan penggabungan teks berdasarkan sintaks C.

V. KESIMPULAN

Algoritma pencocokan string dapat diterapkan dengan beberapa jenis algoritma, salah satunya adalah algoritma Boyer-Moore. Algoritma Boyer-Moore terbukti menjadi suatu metode pencocokan string yang efisien. Perbandingan hasil yang didapatkan dan waktu yang dibutuhkan berimbang dan cocok untuk diterapkan pada pencocokan string dengan pattern yang panjang maupun pendek. Algoritma pencocokan string juga dapat digunakan untuk melakukan penerjemahan suatu Bahasa pemrograman.

REFERENCES

- [1] Munir, Rinaldi. 2009. Diktat Kuliah IF3051 Strategi Algoritma. Bandung : Penerbit Informatika.
- [2] Munir, Rinaldi. 2005. Matematika Diskrit Edisi Ketiga. Bandung : Penerbit Informatika.
- [3] Liem, Inggiani. 2003. Catatan Singkat Bahasa C. Bandung : Penerbit Informatika ITB.
- [4] <http://intanstemapa24.blogspot.com/2014/08/pengertian-bahasa-pemograman.html> diakses pada 3 Mei 2015 pukul 13.05
- [5] <https://iotsmaga.wordpress.com/apa-itu-pascal/> diakses pada 3 Mei 2015 pukul 13.10
- [6] <http://molimood.blogspot.com/2013/02/stack-algoritma-dan-struktur-data.html> diakses pada 3 Mei 2015 pukul 13.15
- [7] <http://writerguy.users.btopenworld.com/Pascal/PascalFrameset.html> diakses pada 3 Mei pukul 13.30

- [8] Kadir , Abdul., 2003, Pemrograman Dasar Turbo C Untuk IBM . Andy Offset, Yogyakarta.

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, 5 Mei 2015



Randi Chilyon Alfianto
13513087