**Algoritma *Zhu Takaoka***

Algoritma *Zhu-Takaoka* merupakan salah satu algoritma pencocokan *string* (*String Matching*) yang merupakan pengembangan dari algoritma *BoyerMoore*. Algoritma ini dipublikasikan oleh Zhu Rui Feng dan Tadao Takaoka pada tahun 1986 Dalam penelitiannya, Zhu dan Takaoka menyebut algoritma pencocokan *string* ini sebagai BM" Algorithm (*BoyerMoore* Algorithm) karena merupakan modifikasi dari algoritma pencocokan *string Boyer-Moore,* yaitu algoritma yang dibuat oleh Boyer R.S. dan Moore J.S. Algoritma *Zhu-Takaoka* dan algoritma *Boyer-Moore* memiliki ciri yang sama dalam proses pencarian *string,* yaitu terdapat tahap *Preprocessing, Right-to-left scan, Bad character rule* dan *Good-suffix rule*. Sementara itu, perbedaan dari kedua algoritma tersebut terletak pada tahap penentuan *Bad character rule.* Dalam *Boyer-Moore, bad character* hanya terdiri dari *array* satu dimensi, sedangkan dalam *Zhu-Takaoka array* dimodifikasi menjadi dua dimensi.

Adapun tahap pencarian pattern menggunakan algoritma *zhu-takaoka* adalah sebagai berikut :

1. *Preprocessing*

   Prepocessing dalam algoritma Zhu-Takaoka meliputi pencarian nilai pergeseran karakter (goodsuffix shift) dan pergeseran karakter jika karakter tidak cocok (bad-character shift). Nilai good-suffix shift ditetukan dalam good-suffix prepocessing sedangkan nilai bad-character shift ditentukan dalam bad- characterprepocessing. Prepocessing dilakukan sebelum proses inti dari pencarian pattern dalam suatu text. (Pratama, 2008)
2. *Right-to-Left Scan Rule*

   Proses inti pencarian Algoritma Zhu-Takaoka yaitu dilakukan dengan teknik Right-to-left scan rule. Teknik ini yaitu melakukan perbandingan antara pattern yang dicari dengan target text secara terbalik yaitu bergerak dari kanan ke kiri. Perbandingan pattern dengan target text dimulai dengan membandingkan karakter terakhir dari pattern (karakter paling kanan) dengan target text paling kanan. Apabila ada kecocokan maka perbandingan akan dilanjutkan dengan bergerak ke kiri sampai karakter pertama dari pattern. Sedangkan apabila terjadi ketidakcocokan maka akan dilakukan pergeseran, besarnya pergesaran yang dilakukan ditentukan oleh dua fungsi pergeseran yaitu bad-character shift dan goodsuffix shift.
3. *Bad-Charcter Shift Rule*

   Aturan bad-character shift dibutuhkan untuk menghindari pengulangan perbandingan yang gagal dari suatu karakter dalam target text dengan pattern. Besarnya pergeseran yang dilakukan dalam aturan bad-character shift disimpan dalam bentuk tabel array dua dimensi, tabel ini terdiri dari beberapa kolom yaitu kolom karakter dan kolom shift yang menunjukkan besarnya pergeseran yang harus dilakukan.
4. *Good Suffix Shift Rule*

   Aturan good-suffix shift dibuat untuk menangani kasus dimana terdapat pengulangan karakter pada pattern. Contoh dibawah ini akan menjelaskan bagaimana aturan bad-character shift gagal dalam menangani adanya perulangan bagian dalam
   pattern.