**LAPORAN TUGAS KECIL 4**

**EKSTRAKSI INFORMASI DARI ARTIKEL BERITA DENGAN ALGORITMA PENCOCOKAN STRING**

**IF2211 Strategi Algoritma**

**­­**

Oleh:

**Daffa Pratama Putra 13518033**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2020**

# **BAB I ALGORITMA PENCOCOKAN STRING**

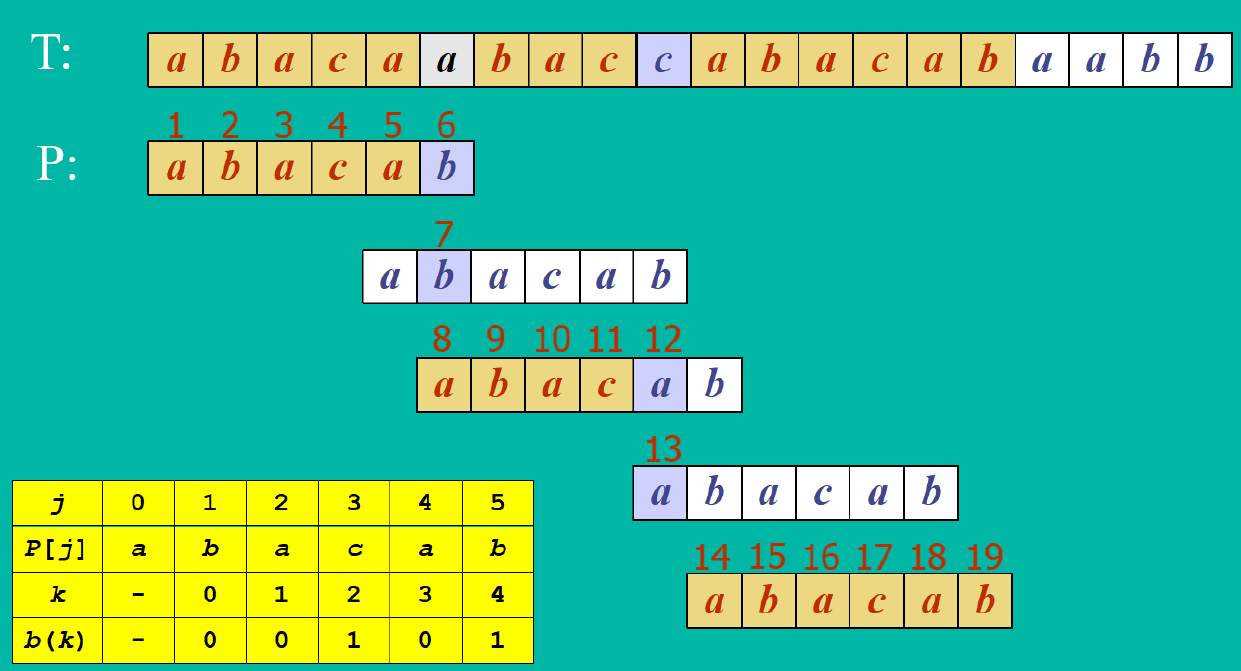
1. **Algoritma Knuth-Morris-Pratt (KMP)**

Algoritma Knuth-Morris-Pratt adalah salah satu algoritma pencocokan *string* yang dinilai lebih baik dari algoritma Brute Force. Algoritma KMP ini dikembangkan oleh salah satu *computer scientist* dari Standford University, Donald E. Knuth pada tahun 1967. Setahun sebelumnya, James H. Morris dan Vaughan R. Pratt juga mengembangkan algoritma yang sama, oleh karena itu nama dari algoritma ini diambil dari nama ketiga penciptanya, yakni Knuth, Morris, dan Pratt.

Pada dasarnya, algoritma ini mirip seperti algoritma Brute Force dalam pencocokan *string*, yaitu melakukan pencocokan dari kiri ke kanan *string*. Namun ada hal yang berbeda. Pada algoritma Brute Force, pergeseran dalam pencocokan *string* dilakukan satu persatu dari kiri ke kanan, baik karakter yang dicocokkan sama atau berbeda. Pergeseran satu persatu dilakukan hingga pencocokan karakter sudah mencapai akhir atau ditemukan *string* yang dicari oleh *pattern* pada *text*. Berbeda dengan algoritma Brute Force, algoritma KMP melakukan pergeseran dalam pencocokan *string* dengan mempertimbangkan *pattern* yang telah dibandingkan. Sehingga pada algoritma KMP dibutuhkan *pre-processing* untuk menentukan jumlah pergeseran yang dilakukan untuk setiap indeks pada *pattern* atau yang disebut dengan fungsi pinggiran.

Fungsi pinggiran atau *failure function* pada algoritma KMP adalah fungsi yang menghitung jumlah pergeseran yang dilakukan dengan menghitung panjang terbesar prefiks dari *pattern,* P[0..k], yang merupakan sufiks dari pattern tersebut, P[1..k], dengan k = (indeks *mismatch* pada pattern) - 1. Dengan demikian, proses pergeseran akan menjadi lebih efisien karena karakter yang sudah dicocokkan pada prefiks tidak dicocokkan kembali. Algoritma KMP menghasilkan kompleksitas waktu O(m+n) dengan O(m) adalah kompleksitas menghitung fungsi pinggiran dan O(n) adalah kompleksitas pencarian *string*.

Berikut adalah contoh pencarian menggunakan algoritma KMP :



Gambar 1. Pencocokan string dengan algoritma KMP (diambil dari Bahan Kuliah IF2211 Strategi Algoritma, Pencocokan String (String/Pattern Matching)

1. **Algoritma Boyer-Moore (BM)**

Algoritma Boyer-Moore

1. **Regex**

if (card instanceof CharacterCard) {  
 colorCard = "F3D06F"; //kuning  
} else if (card instanceof LandCard) {  
 colorCard = "01BAEB"; //biru  
} else if (card instanceof AuraCard) {  
 colorCard = "D13539"; //merah  
} else if (card instanceof DestroyCard) {  
 colorCard = "65C387"; //ijo  
} else if (card instanceof PowerUpCard) {  
 colorCard = "A57FBB"; //ungu  
}

# **BAB II KODE PROGRAM**

1. **Kode Program**

# **BAB III UJI COBA**