

# Aplikasi Algoritma Pencarian String Dalam Sistem Pembayaran Parkir

Andi Kurniawan Dwi P - 13508028  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
If18028@students.if.itb.ac.id

*Sistem pembayaran parkir merupakan sebuah sistem yang menangani pembayaran tarif parkir secara otomatis. Dalam hal ini, sistem dapat mencatat nomor polisi dari setiap kendaraan bermotor yang akan masuk ke tempat parkir, kemudian nomor polisi ini disimpan dalam suatu tabel (database), sistem juga mencatat lama waktu kendaraan bermotor tersebut berada di tempat parkir, ketika kendaraan bermotor tersebut hendak keluar dari tempat parkir sistem melakukan pencarian string terhadap nomor polisi kendaraan bermotor tersebut dalam database dan melakukan perhitungan tarif berdasarkan lama parkirnya.*

*Untuk menentukan besarnya tarif dari kendaraan yang akan keluar dari tempat parkir, sistem harus mampu mencari nomor kendaraan bermotor tersebut dalam database. Dalam hal inilah algoritma pencarian string digunakan. Kumpulan string nomor polisi kendaraan bermotor di dalam database dianggap sebagai suatu Text dan string nomor polisi kendaraan bermotor yang akan keluar dari tempat parkir sebagai Pattern. String Pattern akan dicari dalam untaian string Text dengan menggunakan algoritma pencarian string. Pada makalah ini akan dibahas penggunaan algoritma pencarian string dalam sistem pembayaran parkir.*

*Brute force, pencarian string, sistem pembayaran parkir*

## 1. PENDAHULUAN

Beberapa tahun terakhir ini, jumlah kendaraan bermotor di Indonesia semakin naik drastis. Kenaikan jumlah kendaraan bermotor tampak terasa terutama di kota-kota besar, terbukti dengan banyaknya kemacetan yang terjadi di sepanjang jalan. Dampak lain yang timbul adalah penuhnya tempat-tempat parkir.

Meningkatnya kebutuhan tempat parkir ini tentu membawa keuntungan bagi penyedia layanan tempat parkir. Untuk memaksimalkan keuntungannya, penyedia layanan ini harus mampu mempercepat proses keluar masuk parkir agar tempat parkir nya mampu menampung lebih banyak kendaraan bermotor. Untuk itulah dibutuhkan sebuah sistem pembayaran parkir yang

mampu membantu petugas pembayaran parkir dalam melayani proses pembayaran parkir. Petugas pembayaran kini tidak perlu melakukan pencarian nomor polisi kendaraan bermotor karena hal ini sudah ditangani oleh sistem dan petugas pembayaran hanya tinggal menyebutkan tarif yang ditampilkan oleh sistem dan menerima uang dari pengemudi kendaraan bermotor tersebut. Dengan sistem ini diharapkan proses pembayaran yang biasanya membutuhkan waktu cukup lama dapat dipercepat, sehingga tidak terjadi kemacetan di tempat pembayaran parkir.

## 2. GAMBARAN SISTEM

### 2.1. Kamera

Sistem ini membutuhkan 2 buah kamera utama yang ditempatkan di portal tempat masuk parkir dan di portal tempat keluar parkir (dekat loket pembayaran parkir). Kamera ini diasumsikan sudah cukup canggih untuk dapat membaca nomor polisi kendaraan bermotor yang akan keluar masuk di tempat parkir dan hasil pengambilan gambar oleh kamera ini tidak berupa file gambar melainkan file teks (string) yang berisi nomor polisi kendaraan bermotor tersebut.



### 2.2. Database

String hasil pengambilan gambar ini kemudian akan disimpan di dalam database. Dan ketika kendaraan bermotor mulai masuk pada tempat parkir, sistem juga melakukan pencatatan waktu. Waktu ini akan digunakan nanti ketika melakukan perhitungan lama parkir dan tarif parkir. Waktu ini juga disimpan di dalam database.

Jadi database ini merupakan suatu tabel yang terdiri dari 2 kolom yaitu :

- string yang menyatakan nomor polisi kendaraan yang berada di tempat parkir
- waktu yang menyatakan waktu awal kendaraan bermotor memasuki tempat parkir.

Tabel yang disimpan dalam database :

Nomor polisi	Waktu masuk tempat parkir
D 1234 EF	12.30
D 3253 RT	13.15
B 1050 RI	15.00
... dst	

Sistem dapat membaca isi database dan menjadikan isi database tersebut menjadi suatu untaian string yang merupakan kumpulan string nomor polisi kendaraan dan waktu awal kendaraan memasuki tempat parkir. String ini berupa nomor polisi dan waktu masuk suatu kendaraan dipisahkan dengan tanda koma ',' dan antar kendaraan dipisahkan dengan spasi ' ' serta pada akhir untaian string diberi penanda '.' sebagai tanda bahwa untaian string telah berakhir (EOP). Untaian string ini dapat digambarkan seperti di bawah ini

D1234EF,1230 D3253RT,1315 B1050RI,1500 ....dst
--

String di atas berarti ada beberapa kendaraan bermotor yang sedang parkir di tempat parkir antara lain kendaraan dengan nomor polisi D1234EF yang masuk di tempat parkir pada pukul 12.30, nomor polisi D3253RT yang masuk di tempat parkir pada pukul 13.15, nomor polisi B1050RI yang masuk di tempat parkir pada pukul 15.00, dan seterusnya.

Pada sistem ini terdapat kursor pemindai karakter, sebut saja CC yang berisi karakter pada posisi kursor. Terdapat prosedur nextChar yang berfungsi memajukan CC satu karakter. Selain itu juga terdapat prosedur nextKata yang memajukan CC pada kata selanjutnya.

### 2.3. Prosedur parkir

1. Kendaraan bermotor berhenti di depan portal masuk tempat parkir, pengemudi mengambil karcis parkir, kamera mengambil gambar nomor polisi kendaraan
2. Portal terbuka, kendaraan bermotor ke tempat parkir, sistem melakukan set waktu masuk kendaraan, sistem menambahkan nomor polisi dan waktu masuk kendaraan dalam database
3. Kendaraan bermotor parkir beberapa lama
4. Kendaraan bermotor hendak keluar dari tempat parkir menuju loket pembayaran portal keluar tempat parkir, kamera mengambil gambar nomor polisi kendaraan bermotor
5. Sistem melakukan pencarian nomor polisi dan melakukan perhitungan waktu parkir dan tarif

parkir, pengemudi kendaraan bermotor menyerahkan karcis parkir dan melakukan pembayaran pada petugas

6. Portal terbuka dan kendaraan bermotor keluar dari tempat parkir

### 3. ALGORITMA PENCARIAN STRING

Algoritma Pencocokan String adalah suatu algoritma memeriksa kesamaan pettern suatu kata dengan kata yang terdapat dalam kamus kata. Dua buah kata dikatakan cocok apabila jumlah huruf kedua kata itu sama dan huruf-huruf pada indeks yang bersesuaian bernilai sama. Terdapat 3 macam algoritma pencocokan string yang lazim digunakan yaitu algoritma brute force, algoritma Knuth-Morris-Pratt (KMP), dan algoritma Boyer-Moore (BM).

Algoritma yang pertama adalah Algoritma Brute Force Pencocokan string. Kita asumsikan bahwa database sistem (untaian string nomor polisi dan waktu) berada di dalam array T[1..n] dan pattern yang akan dicocokkan berada di dalam array P[1..m], maka algoritma brute force pencocokan string adalah sebagai berikut:

1. Mula-mula pattern P dicocokkan pada awal teks T.
2. Jika cocok, dengan bergerak dari kiri ke kanan, bandingkan setiap setiap karakter di dalam pattern P dengan karakter yang bersesuaian di dalam teks T sampai:
  - a. semua karakter yang dibandingkan cocok atau sama (pencarian berhasil)
  - b. dijumpai sebuah ketidakcocokan karakter (pencarian belum berhasil)
3. Bila pattern P belum ditemukan kecocokannya dan teks T belum habis, geser pattern P satu karakter ke kanan dan ulangi langkah 2.

Contoh:

Teks: nobody noticed him

Pattern: not

```

nobody noticed him
s=0    not
s=1    not
s=2    not
s=3    not
s=4    not
s=5    not
s=6    not
s=7    not

```

Kompleksitas algoritma brute-force:

Kompleksitas kasus terbaik adalah O(n). Kasus terbaik terjadi jika yaitu bila karakter pertama pattern P tidak pernah sama dengan karakter teks T yang dicocokkan. Pada kasus ini, jumlah perbandingan yang dilakukan

paling banyak  $n$  kali misalnya:

Teks: Ini adalah string panjang yang berakhir dengan zz

Pattern: zz

Kasus terburuk membutuhkan  $m(n - m + 1)$  perbandingan, yang mana kompleksitasnya adalah  $O(mn)$ , misalnya:

Teks:

aaa

Pattern: aaaab

Algoritma yang lain adalah KMP. Algoritma ini memiliki kompleksitas waktu yang lebih baik dari pada algoritma Brute Force. Pada algoritma brute force, setiap kali ditemukan ketidakcocokan pattern dengan teks, maka pattern digeser satu karakter ke kanan. Sedangkan pada algoritma KMP, kita memelihara informasi yang digunakan untuk melakukan pergeseran yang lebih jauh. Algoritma ini mangkus, hanya jika persoalan pencocokan string yang kita hadapi berupa mencari substring suatu string terhadap string yang lain

Lain lagi dengan algoritma Boyer-Moore yang merupakan variasi lain dari pencarian string dengan melompat maju sejauh mungkin. Algoritma BM melakukan perbandingan pattern mulai dari kanan (belakang), dimana algoritma KMP melakukan perbandingan mulai dari kiri (depan). Sama halnya seperti algoritma KMP, algoritma ini mangkus, hanya jika persoalan pencocokan string yang kita hadapi berupa mencari substring suatu string terhadap string yang lain.

Algoritma KMP dan BM memang lebih mangkus jika dibandingkan dengan algoritma brute force. Tapi itu berlaku hanya jika persoalan pencocokan string yang kita hadapi, berupa mencari substring suatu string terhadap string yang lain. Sedangkan masalah pencocokan string dalam sistem pembayaran parkir ini tidak demikian. Suatu kata (nomor polisi) dikatakan sah, apabila kata tersebut persis sama dengan salah satu kata (nomor polisi) yang terdapat di dalam database sistem, dengan artian jumlah kata dan huruf-huruf yang bersesuaian pada kedua string persisi sama.

Oleh karena itu algoritma pencocokan string yang akan dipakai dalam sistem ini adalah Algoritma Brute Force Pencocokan string dengan sedikit penyesuaian.

Kita asumsikan bahwa database sistem (untaian string nomor polisi dan waktu) berada di dalam array  $T[1..n]$  dan pattern yang akan dicocokkan berada di dalam array  $P[1..m]$ , maka algoritma brute force pencocokan string adalah sebagai berikut:

1. Mula-mula pattern  $P$  dicocokkan pada awal teks  $T$ .

2. Jika cocok, dengan bergerak dari kiri ke kanan, bandingkan setiap setiap karakter di dalam pattern  $P$  dengan karakter yang bersesuaian di dalam teks  $T$  sampai:
  - c. semua karakter yang dibandingkan cocok atau sama (pencarian berhasil)
  - d. dijumpai sebuah ketidakcocokan karakter (pencarian belum berhasil)
3. Bila pattern  $p$  belum ditemukan kecocokannya dan teks  $T$  belum habis, geser pattern  $P$  menuju awal kata selanjutnya di dalam teks (ditandai dengan "spasi") dan ulangi langkah 2. Bila semua karakter yang dibandingkan cocok, periksa apakah teks sudah mencapai tanda koma ','. Jika sudah, berarti pencarian sukses, jika belum berarti gagal, geser pattern  $P$  menuju awal kata selanjutnya di dalam teks, ulangi langkah 2.

#### 4. ALGORITMA SISTEM PEMBAYARAN PARKIR

Berikut ini akan dijelaskan prosedur-prosedur dan fungsi-fungsi yang akan digunakan dalam sistem pembayaran parkir.

##### 4.1. Prosedur nextKata

Kita akan menerapkan overloading dalam prosedur ini. Overloading berarti, kita mendefinisikan lebih dari satu body terhadap subprogram yang sama. Yang pertama, prosedur nextKata tanpa parameter, (nextKata()) akan meletakkan cursor scanning terhadap database sistem (untaian string nomor polisi dan waktu) pada kata selanjutnya. Hal ini bisa diketahui dengan mencari karakter spasi dan meletakkan cursor pada huruf setelahnya. Kemudian definisi yang kedua, prosedur nextKata dengan parameter, (nextKata(char karakterAwal)) akan meletakkan cursor scanning terhadap database sistem (untaian string nomor polisi dan waktu) pada kata pertama yang berawalan karakterAwal.

Ada beberapa langkah dalam prosedur nextKata(), yaitu:

1. Scanning database sistem (untaian string nomor polisi dan waktu) sampai menemukan karakter spasi.
2. Majukan cursor satu huruf untuk menemukan huruf pertama dari kata berikutnya.

Begitu pula dengan algoritma nextKata(char karakterAwal). Langkah-langkahnya adalah:

1. Scanning database sistem (untaian string nomor polisi dan waktu) sampai menemukan karakter spasi.
2. Majukan cursor satu huruf untuk menemukan huruf pertama dari kata berikutnya.

3. Jika huruf yang sedang di scan adalah karakterAwal, maka berhenti. Jika tidak, ulangi dari langkah 1.

#### 4.2. Fungsi stringToTime

Fungsi ini untuk melakukan konversi dari array of character ke dalam tipe time (waktu). Fungsi ini akan digunakan untuk mengambil nilai waktu awal parkir yang disimpan dalam untaian string. Waktu ini kemudian akan digunakan untuk perhitungan lama parkir dan tarif parkir yang harus dibayar.

#### 4.3. Fungsi hitungSelisihWaktu

Fungsi ini merupakan fungsi yang menerima 2 buah parameter input bertipe time (waktu) dan menghasilkan output bertipe float. Parameter pertama merupakan waktu awal dan parameter kedua merupakan waktu akhir. Nilai float keluaran merupakan hasil selisih waktu akhir dengan waktu awal dalam jam. Misal a=12.00 dan b=17.30 jika  $x = \text{hitungSelisihWaktu}(a,b)$  maka x akan bernilai 5,5.

#### 4.4. Fungsi hitungTarif

Inilah inti dari sistem pembayaran parkir. Di dalam fungsi hitungTarif, kita akan mengimplementasikan algoritma pencocokan string. Langkah-langkah nya adalah seperti yang telah dijelaskan pada bab 3.

Fungsi hitungTarif adalah fungsi yang mengembalikan nilai bertipe integer yang merupakan besar tarif yang akan dihitung berdasarkan lama waktu parkir. Kita asumsikan bahwa database sistem (untaian string nomor polisi dan waktu) berada di dalam array T[1..n] dan pattern yang akan dicocokkan berada di dalam array P[1..m]. Untuk pertama kali posisi kursor di database sistem (untaian string nomor polisi dan waktu) berada pada huruf pertama kata pertama. Maka langkah-langkah dalam fungsi hitungTarif adalah sebagai berikut:

1. Panggil prosedur nextKata(P[1]), yang berarti kita akan menuju huruf pertama dari kata pertama yang berawalan P[1] dalam database sistem (untaian string nomor polisi dan waktu). Assignment variable Indeks dengan nilai 1 (Indeks=1)
2. Cocokkan CC dengan P[Indeks]
3. Jika cocok, majukan CC dan incrementasi Indeks. Ulangi langkah 2 sampai karakter pada pattern habis.
4. Jika cocok dan karakter pada pattern habis, panggil prosedur nextChar(), periksa apakah CC sama dengan tanda koma ','. Jika ya, berarti cocok.
5. Jika tidak cocok, panggil fungsi nextKata(). Jika CC sama dengan karakterAwal, ulangi dari

langkah 2 sampai pattern ditemukan.

6. Isi JamAwal dengan karakter di antara tanda koma ',' dan tanda spasi ' '. Dengan memanggil nextChar() sampai bertemu tanda spasi ' '. Konversikan nilai pada JamAwal tersebut agar menjadi tipe time dengan memanggil fungsi stringToTime(JamAwal).
7. Hitung lama waktu dengan memanggil fungsi hitungSelisihWaktu(a,b) dengan a=JamAwal dan b=waktu sekarang.
8. Kalikan hasil perhitungan dari nomor 7 dengan tarif per jamnya (asumsi tarif parkir per jam = 1000 rupiah).

```
function hitungTarif(P: array [1..n] of char) → integer
{
mengembalikan nilai bertipe integer yang merupakan
besar tarif yang akan dihitung berdasarkan lama waktu
parkir dari kendaraan yang bernomor polisi P
}
```

#### Deklarasi:

```
CC : char
habis : boolean
Indeks : integer
tarif : integer
lama : float
temp : array[1..4] of char
waktuAwal : time
waktuAkhir : time
procedure nextChar()
procedure nextKata()
procedure nextKata(c: char)
function stringToTime(x: array[1..4] of char) → time
function hitungSelisihWaktu(a: time, b: time) → float
```

#### Algoritma:

```
nextKata(P[1])
do
Indeks ← 1
while ((CC = P[Indeks]) and (CC != EOP))
nextChar
Indeks ← Indeks+1
if (Indeks > P.length)
break
endif
endwhile
if (CC = ',') and (Indeks > P.length)
break
else
nextKata
```

```
while (CC!=EOP)

Indeks ← 1
while (CC != ' ')
    nextChar
    temp[Indeks] ← CC
endwhile
waktuAwal ← stringToTime(temp)
waktuAkhir ← waktu sekarang
lama ← hitungSelisihWaktu(waktuAwal, wakuAkhir)
tarif ← lama * 1000

→ tarif
```

## 5. SIMPULAN

Dalam menyusun makalah ini, penulis dapat menyimpulkan :

1. Algoritma pencarian string dapat diterapkan dalam sistem pembayaran parkir
2. Algoritma pencarian string yang paling cocok dalam sistem pembayaran parkir adalah algoritma greedy.

## 6. REFERENSI

- [1] Rinaldi Munir, 2009, "Diktat Kuliah IF3051 Strategi Algoritma", Program Studi Teknik Informatika ITB.

## 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, 8 Desember 2010



Andi Kurniawan Dwi P - 13508028