Penentuan Nomor Kendaraan dengan Algoritma *Brute Force* dan Pencocokan *String*

Nira Rizki Ramadhani - 13516018
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesa 10 Bandung 40132, Indonesia
nirarizkiramadhani@gmail.com

***Abstrak*—Tanda Nomor Kendaraan Bermotor (TNKB) adalah sebuah atribut penting dan wajib ada pada setiap kendaraan bermotor yang dioperasikan di Indonesia. Nomor kendaraan ini diletakkan pada bagian depan dan belakang kendaraan bermotor. Penentuan nomor kendaraan ini memiliki prosedur tersendiri berdasarkan ketentuan yang berlaku di Indonesia. Masyarakat Indonesia pun dapat mengajukan permohonan untuk pembuatan nomor kendaraan unik apabila melaksanakan syarat dan ketentuan yang ada. Algoritma *brute force* merupakan algoritma yang lempang (*straightforward*) dan dapat digunakan untuk menyelesaikan berbagai macam masalah, salah satunya adalah dalam membangkitkan nomor kendaraan ini. Dalam pembangkitan nomor kendaraan unik berdasarkan permintaan, dapat digunakan algoritma pencocokan *string* Knuth-Morris-Pratt (KMP) untuk mengecek keunikan nomor tersebut. Apabila nomor tersebut belum terpakai, maka nomor kendaraan tersebut dapat digunakan. Pada makalah ini akan dijelaskan implementasi algoritma *Brute Force* dan Knuth-Morris-Pratt (KMP) dalam penentuan nomor kendaraan.**

Kata Kunci—Nomor Kendaraan; Brute Force; straightforward; KMP

I. PENDAHULUAN

Dewasa ini, perkembangan teknologi dalam bidang otomotif kian semakin maju, masyarakat Indonesia pun kerap mengikuti perkembangan teknologi otomotif tersebut. Sudah tidak bisa dipungkiri bahwa masyarakat Indonesia sangat senang menggunakan kendaraan pribadi, bahkan tidak sedikit yang mengoleksi kendaraan pribadi. Hal tersebut bisa dilihat di jalanan pada kota-kota besar, banyak sekali kendaraan minibus, bus, truk dan kendaraan bermotor lainnya dengan berbagai nomor kendaraan, terlebih lagi apabila sedang masa libur panjang.

Saat sedang libur panjang, tidak jarang kita melihat kendaraan dengan nomor kendaraan yang berbeda-beda huruf depan maupun belakangnya. Terkadang, terdapat pula nomor kendaraan yang unik seperti memiliki sebuah arti. Lalu, pernahkah kita berpikir apa arti dari susunan nomor kendaraan tersebut?

Indonesia adalah negara kepualauan yang memiliki banyak provinsi. Hingga saat ini telah tercatat 34 provinsi di Indonesia yang tersebar dari Sabang sampai Merauke[2]. Pelat nomor adalah salah satu jenis identifikasi kendaraan bermotor. Pelat nomor juga disebut pelat registrasi kendaraan, di Amerika Serikat dikenal sebagai pelat izin (*license plate*). Bentuknya berupa potongan pelat logam yang dipasang pada kendaraan bermotor sebagai identifikasi resmi. Jumlah dari pelat nomor kendaraan untuk sebuah kendaraan adalah sepasang, yakni untuk di depan dan di belakang[3]. Pembuatan pelat nomor kendaraan unik atau berdasarkan keinginan pun dapat dilakukan dengan mengikuti prosedur dan syarat yang berlaku.

Algoritma *brute force* dapat digunakan untuk menentukan susunan nomor kendaraan karena algoritma ini dapat menyelesaikan masalah apapun dengan sangat sederhana, langsung dan cara yang jelas[1].

Selain itu, dalam pembuatan nomor kendaraan yang sesuai dengan permintaan pengguna kendaraan dapat dilakukan pencocokan *string* terlebih dahulu untuk memeriksa eksistensi nomor tersebut dalam *database*. Algoritma pencocokan *string* yang akan digunakan dalam makalah ini adalah algoritma Knuth-Morris-Pratt atau biasa disingkat menjadi algoritma KMP.

II. TEORI DASAR

A. Tanda Nomor Kendaraan Bermotor (TNKB)

Setiap kendaraan bermotor yang beroperasi di jalan wajib dilengkapi dengan dengan Surat Tanda Kendaraan Bermotor dan Tanda Nomor Kendaraan Bermotor. Tanda Nomor Kendaraan Bermotor (TNKB) tersebut harus memuat kode wilayah, nomor registerasi dan masa berlaku. Tanda Nomor Kendaraan Bermotor ini sering disebut dengan Nomor Polisi (Nopol) atau Pelat Nomor. Bahan yang digunakan untuk pelat nomor kendaraan bermotor di Indonesia terbuat dari aluminium dan wajib dipasangkan di depan dan belakang bagian kendaraan bagi setiap kendaraan bermotor yang dioperasikan di jalan[5].

Nomor kendaraan diberikan sesuai dengan urutan pendaftaran kendaraan bermotor. Nomor urut tersebut terdiri dari 1 hingga 4 angka dan ditempatkan setelah kode wilayah pendaftaran[4]. Nomor urut pendaftaran kendaraan bermotor dialokasikan sesuai kelompok jenis kendaraan bermotor, yaitu

nomor urut 1 sampai dengan 1999 untuk kelompok jenis kendaraan penumpang, nomor urut 2000 sampai dengan 6999 untuk kelompok jenis kendaraan sepeda motor, nomor urut 7000 sampai dengan 7999 untuk kelompok jenis kendaraan bus, dan nomor urut 8000 sampai dengan 9999 untuk kelompok jenis kendaraan beban[5].

Warna pelat nomor atau Tanda Nomor Kendaraan Bermotor di Indonesia pun memiliki arti tersendiri, di antaranya adalah sebagai berikut.

- Warna dasar hitam dengan tulisan berwarna putih adalah untuk kendaraan bermotor perseorangan atau sewa
- Warna dasar kuning dengan tulisan berwarna hitam adalah untuk kendaraan bermotor umum
- Warna dasar merah dengan tulisan berwarna putih adalah untuk kendaraan bermotor milik pemerintah
- Warna dasar putih atau merah dengan tulisan berwarna hitam adalah untuk kendaraan bermotor Korps Diplomatik Negara Asing
- Warna dasar hitam dengan tulisan berwarna putih dan terdiri dari lima angka serta kode angka negara yang dicetak lebih kecil dengan format sub-bagian adalah diperuntukan kendaraan bermotor *staff* operasional Korps Diplomatik Negara Asing

Selain warna, masa berlaku Tanda Nomor Kendaraan Bermotor di Indonesia juga penting. Masa berlaku tersebut terletak di bawah nomor pendaftaran dengan tulisan yang lebih kecil. Masa berlaku tersebut terdiri dari dua angka pertama yang menunjukkan bulan berakhirnya masa berlaku dan dua angka terakhir adalah tahun berakhirnya masa berlaku.

Berikut ini adalah beberapa daftar Tanda Nomor Kendaraan Bermotor di Indonesia berdasarkan wilayahnya[5].

Kode Wilayah	Daerah	Kabupaten / Kota
BL	Provinsi Nanggroe Aceh Darussalam	Seluruh Kabupaten/Kota di Provinsi Nanggroe Aceh Darussalam
BB	Provinsi Sumatera Utara	• Kabupaten Tapanuli Utara • Kabupaten Tapanuli Tengah • Kabupaten Tapanuli Selatan • Kabupaten Sibolga • Kabupaten Dairi • Kabupaten Nias
BK	Provinsi Sumatera Utara	• Kota Medan • Kabupaten Deli Serdang • Kabupaten Tebing Tinggi • Kabupaten Langkat • Kabupaten Binjai • Kabupaten Simalungun • Kabupaten Pematang Siantar • Kabupaten Tanah Karo • Kabupaten Asahan • Kabupaten Labuhan Batu
BM	Provinsi Riau	Seluruh Kabupaten/Kota di Provinsi Riau
BP	Provinsi Kepulauan Riau	Seluruh Kabupaten/Kota di Provinsi Kepulauan Riau
BG	Provinsi Sumatera Selatan	Seluruh Kabupaten/Kota di Provinsi Sumatera Selatan
BN	Provinsi Bangka-Belitung	Seluruh Kabupaten/Kota di Provinsi Bangka-Belitung
BE	Provinsi Lampung	Seluruh Kabupaten/Kota di Provinsi Lampung
BD	Provinsi Bengkulu	Seluruh Kabupaten/Kota di Provinsi Bengkulu
BH	Provinsi Jambi	Seluruh Kabupaten/Kota di Provinsi Jambi
B	Provinsi DKI Jakarta	Seluruh Kabupaten/Kota di Provinsi DKI Jakarta
D	Provinsi Jawa Barat	• Kota Bandung • Kabupaten Bandung
F	Provinsi Jawa Barat	• Kodya Bogor • Kabupaten Bogor • Kabupaten Cianjur • Kabupaten Sukabumi
T	Provinsi Jawa Barat	• Kabupaten Purwakarta • Kabupaten Karawang • Kabupaten Subang
E	Provinsi Jawa Barat	• Kota Cirebon • Kabupaten Cirebon • Kabupaten Indramay • Kabupaten Malajengka • Kabupaten Kuningan
Z	Provinsi Jawa Barat	• Kabupaten Garut • Kabupaten Sumedang • Kabupaten Tasikmalaya • Kabupaten Ciamis
H	Provinsi Jawa Tengah	• Kota Semarang • Kabupaten Salatiga • Kabupaten Kendai • Kabupaten Demak • Kabupaten Grobogan
G	Provinsi Jawa Tengah	• Kota Pekalongan • Kabupaten Pekalogan • Kabupaten Brebes • Kabupaten Tegal • Kabupaten Siawi • Kabupaten Batang • Kabupaten Pemalang
K	Provinsi Jawa Tengah	• Kabupaten Pati • Kabupaten Kudus • Kabupaten Jepara • Kabupaten Rembang • Kabupaten Blora
R	Provinsi Jawa Tengah	• Kabupaten Banyumas • Kabupaten Cilacap • Kabupaten Purbalingga • Kabupaten Banjarnegara

Gambar 1. Daftar Tanda Kendaraan Bermotor di Indonesia

Huruf belakang nomor kendaraan pun memiliki arti tersendiri berdasarkan kabupaten/kota tempat pendaftaran nomor kendaraan dan untuk masing-masing kabupaten/kota memiliki beberapa macam pilihan huruf mulai dari 1 hingga 3 huruf. Misalkan untuk B XXXX XYZ, X umumnya mewakili tempat kendaraan tersebut terdaftar, Y umumnya jenis kedaraan berdasar golongan. Beberapa huruf yang mewakili kategori kendaraan:

- A: Sedan/Motor
- F: Minibus, *Hatchback*, *City Car*
- V: Minibus
- J: *Jeep* dan SUV
- D: Truk
- T: Taksi
- U: Kendaraan Staf Pemerintah
- Q: Kendaraan Staf Pemerintah

Dan Z merupakan huruf acak untuk pembeda.[4]

Selain melakukan pembuatan nomor kendaraan sesuai prosedur normal, masyarakat juga dapat mengajukan kombinasi angka dan huruf sesuai yang diinginkan dengan mengajukan formulir permohonan kepada polisi. Apabila kombinasi pelat nomor belum pernah dipakai sebelumnya, pemohon selanjutnya dapat melakukan pembayaran sesuai tarif Penerimaan Negara Bukan Pajak (PNBP)[6].

B. Algoritma Brute Force

Brute force adalah sebuah pendekatan yang lempang (*straightforward*) untuk memecahkan suatu masalah. Algoritma ini didasarkan langsung pada pernyataan masalah dan definisi konsep yang dilibatkan. Algoritma *brute force* umumnya tidak mangkus dan tidak cerdas, sering kali membutuhkan jumlah langkah yang besar dalam penyelesaiannya, terutama dalam memecahkan masalah yang memerlukan ukuran masukan yang besar. Algoritma ini sering juga disebut sebagai algoritma naif.

Seluruh permasalahan dapat diselesaikan dengan algoritma *brute force*, namun algoritma ini sering kali menjadi pilihan yang kurang disukai karena ketidakmangkusannya. Namun, untuk masalah yang ukurannya kecil, kesederhanaan algoritma *brute force* lebih diperhitungkan daripada ketidakmangkusannya. Algoritma *brute force* sering kali lebih mudah diimplementasikan daripada algoritma lain yang lebih canggih, dan karena kesederhanaannya, terkadang algoritma ini dapat lebih mangkus ditinjau dari segi implementasinya. Beberapa contoh implementasi algoritma *brute force* di antaranya dalam mencari elemen terbesar/terkecil dari sebuah tabel, menghitung jumlah dari n buah bilangan, metode pengurutan *Bubble Sort*, dan uji keprimaan.

Algoritma *brute force* memiliki kekuatan dan kelemahan. Kekuatan dari algoritma *brute force* adalah sebagai berikut.

1. Metode *brute force* dapat digunakan untuk memecahkan hampir sebagian besar masalah (*wide applicability*).
2. Metode *brute force* sederhana dan mudah dipahami.
3. Metode *brute force* menghasilkan algoritma yang layak untuk beberapa masalah penting, di antaranya pencarian, pengurutan, pencocokan *string*, dan perkalian matriks.
4. Metode *brute force* menghasilkan algoritma baku untuk tugas-tugas komputasi seperti penjumlahan/perkalian n buah bilangan, menentukan elemen minimum atau maksimum di dalam tabel.

Kelemahan algoritma *brute force* adalah sebagai berikut.

1. Metode *brute force* jarang menghasilkan algoritm yang mangkus.
2. Beberapa algoritma *brute force* lambat.
3. Tidak sekreatif teknik pemecahan masalah lainnya.

Meskipun metode *brute force* bukan sebuah metode canggih, bukan berarti metode *brute force* merupakan algoritma yang bodoh karena semua tergantung pada jenis persoalan yang ingin diselesaikan. Jika persoalan tidak terlalu besar, waktu *running* program *brute force* jauh lebih murah daripada waktu yang dibutuhkan pemrogram untuk mengembangkan algoritma yang lebih cerdas. "*When in doubt, use brute force",* begitu kata Ken Thompson, salah seorang penemu Unix[1].

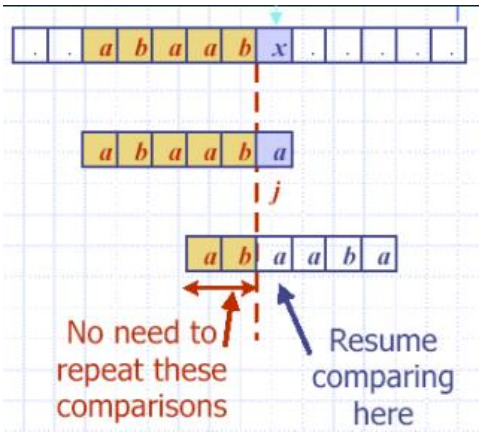
C. Algoritma Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt merupakan algoritma yang dikembangkan oleh D. E. Knuth, bersama-sama dengan J. H. Morris dan V. R. Pratt. Algoritma ini merupakan salah satu algoritma yang sering digunakan untuk menyelesaikan masalah pencocokan *string* dan merupakan penyempurnaan dari algoritma pencocokan *string* menggunakan algoritma *brute force*. Pada algoritma *brute force*, setiap kali ditemukan ketidakcocokan *pattern* dengan teks, maka *pattern* akan digeser satu ke kanan. Sedangkan pada algoritma Knuth-Morris-Pratt, kita memelihara informasi yang digunakan untuk melakukan jumlah pergeseran. Algoritma menggunakan informasi tersebut untuk membuat pergeseran yang lebih jauh. Dengan algoritma Knuth-Morris-Pratt ini, waktu pencarian dapat dikurangi secara signifikan.

Gambar 2. Donald Ervin Knuth (Sumber: [8])

Algoritma KMP melakukan proses awal (*preprocessing*) terhadap *pattern* P dengan menghitung fungsi pinggiran (*border function*), yakni mengindikasikan pergeseran *s*

terbesar yang mungkin dengan menggunakan perbandingan yang dibentuk sebelum pencarian *string*. Fungsi pinggiran ini berguna untuk mencegah pergeseran yang tidak berguna, seperti halnya pada algoritma *brute force* yang melakukan pergeseran satu karakter. Fungsi pinggiran hanya bergantung pada karakter-karakter di dalam *pattern*, dan bukan pada karakter-karakter di dalam teks. Oleh karena itu, perhitungan fungsi pinggiran dapat dilakukan di awal sebelum pencarian *string* dilakukan. Fungsi pinggiran *b(j)* didefinisikan sebagai ukuran awalan terpanjang dari P yang merupakan akhiran dari P[1..j]. Contohnya adalah sebagai berikut.



Gambar 3. Contoh Implementasi Algoritma KMP
(Sumber: [9])

Cara yang dilakukan adalah mencari *prefix* terbesar dari **ab**aab yang juga merupakan *suffix* dari aba**ab**. Maka ditemukan jawabannya adalah ab dengan panjang 2. Lalu, kita atur j menjadi 2, dengan j merupakan posisi j baru untuk melakukan perbandingan selanjutnya. Jumlah pergeseran yang dilakukan adalah *s* = panjang(abaab) – panjang(**ab**) = 5 – 2 = 3[9].

Algoritma untuk menghitung fungsi pinggiran adalah sebagai berikut.

```
procedure HitungPinggiran(input m : integer, p :
array[1..m] of char, output b : array[1..m] of
integer)

{Menghitung nilai b[1..m] untuk pattern P[1..m]}

Deklarasi:
  k, q : integer

Algoritma:
  b[1] ← 0
  q ← 2
  k ← 0
  for q ← 2 to m do
    while ((k > 0) and (P[q] ≠ P[k+1])) do
      k←b[k]
    endwhile
    if P[q] = P[k+1] then
      k←k+1
    endif
    b[q]=k
  endfor
```

Misalkan kita ingin meninjau *pattern P* = abcabd dari teks = abcabcabd. Nilai fungsi pinggiran untuk setiap karakter di dalam *P* adalah sebagai berikut.

TABEL I. FUNGSI PINGGIRAN

j	1	2	3	4	5	6
P[*j*]	a	b	c	a	b	d
b(*j*)	0	0	0	1	2	0

Lakukan pencocokan pertama sebagai berikut: samakan (*align*) ujung kiri *pattern* dengan ujung kiri teks. Karakter-karakter pada posisi 1..5 cocok, namun c dan d posisi 6 menghasilkan ketidakcocokan. Jumlah pergeseran selanjutnya ditentukan oleh pinggiran dari awalan P yang bersesuaian (cocok).

```
          1 2 3 4 5 6 7 8 9...
Teks:     abcabcabd
Pattern:  abcabd
            ↑
          j = 3
```

Pada contoh di atas, awalan yang bersesuaian adalah abcab, dengan panjang l = 5. Pinggiran yang terpanjang untuk *string P*[1..5] adalah ab yang panjangnya adalah *b*(5) = 2. Jarak pergeseran adalah l – b = 5 – 2 = 3. Jadi, *pattern P* digeser sebanyak 3 karakter dan perbandingan dilakukan mulai pada posisi *j* = 3 dihitung dari awal *pattern*.

Algoritma KMP memiliki keuntungan dan kerugian. Keuntungan dari algoritma ini adalah tidak perlu bergerak mundur dalam teks, hal ini membuat algoritma KMP bagus untuk melakukan pemrosesan file yang sangat besar yang dibaca dari perangkat eksternal. Kerugian dari algoritma ini adalah tidak berfungsi dengan baik pada ukuran alfabet yang meningkat, selain itu kemungkinan ketidakcocokan pun lebih banyak dan ketidakcocokan cenderung terjadi di awal *pattern*, sedangkan algoritma ini lebih cepat ketika ketidakcocokan terjadi di akhir *pattern*.

Kompleksitas waktu algoritma KMP adalah $O(m+n)$, dengan waktu untuk menghitung fungsi pinggiran sebesar $O(m)$ dan pencarian *string* dibutuhkan waktu sebesar $O(n)$.[1]

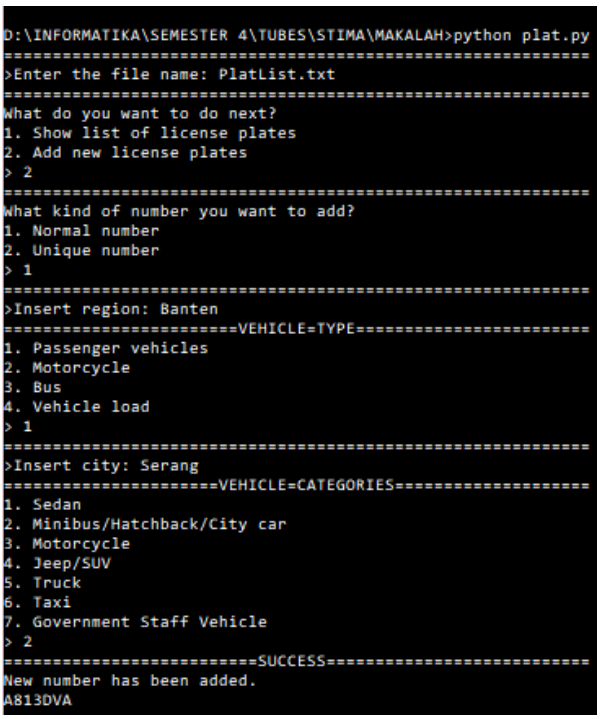
III. PENENTUAN NOMOR KENDARAAN MENGGUNAKAN ALGORITMA BRUTE FORCE

Brute force merupakan sebuah algoritma yang dapat digunakan untuk memecahkan masalah apapun. Dalam penentuan nomor kendaraan yang tersusun atas *XX YYYY ZZZ*, dengan *XX* merupakan huruf sebanyak 1 sampai 2 karakter berdasarkan wilayah pendaftaran, *YYYY* merupakan angka sebanyak 1 sampai 4 digit berdasarkan urutan pendaftaran, dan *ZZZ* merupakan huruf sebanyak 1 sampai 3 karakter berdasarkkan nomor urut pendaftaran kendaraan bermotor yang dialokasikan sesuai kelompok jenis kendaraan bermotor, algoritma *brute force* dapat digunakan, yaitu dengan menyusun komponen dari nomor kendaraan satu persatu.

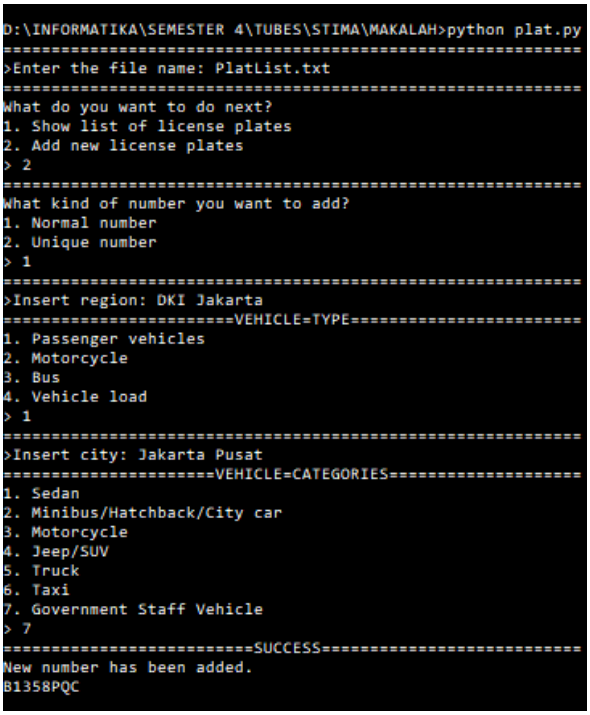
Implementasi algoritmanya akan dilakukan dengan pendekatan yang lempang, yakni dalam penyusunan huruf depan *XX* hanya akan ditentukan berdasarkan wilayah pendaftaran yang contoh daftarnya telah dilampirkan pada Bab II dalam makalah ini. Oleh karena itu, penentuan *XX* akan dibutuhkan masukan berupa wilayah pendaftaran.

Dalam penyusunan angka dari 1 hingga 4 digit, akan diperlukan masukan berupa kelompok jenis kendaraan bermotor terlebih dahulu, selanjutnya angka-angka tersebut akan dibangkitkan secara *brute force*, yakni dengan menambahkannya dengan satu untuk setiap pendaftaran yang baru. Misalkan, kendaraan dengan kelompok jenis kendaraan beban yang melakukan pendaftaran di Jakarta Pusat, DKI Jakarta telah tercatat nomor urut hingga 7653, maka untuk pendaftar selanjutnya akan mendapatkan nomor urut 7654.

Dalam penyusunan huruf belakang *YYY* akan dilakukan dengan meminta masukkan berupa kabupaten/kota yang memiliki kombinasi huruf berbeda-beda untuk kabupaten/kota berbeda pula. Nomor kendaraan dibangkitkan secara *brute force* dan akan terdapat banyak sekali kombinasi nomor kendaraan yang hingga saat ini masih terus dapat memenuhi permintaan nomor kendaraan untuk seluruh kendaraan di Indonesia meskipun jumlahnya sangatlah banyak. Bayangkan saja, untuk nomor kendaraan dengan huruf depan *XX* dari suatu wilayah dapat memiliki kombinasi angka dari 1 hingga 9999 ditambah dengan kombinasi huruf belakang *YYY*. Oleh karena itu, sangat dibutuhkan pencatatan nomor kendaraan dalam sebuah *database*. Untuk dapat mengilustrasikannya, berikut contoh sederhana dengan cara *brute force* untuk membangkitkan nomor kendaraan.



Gambar 4. Contoh 1 Ilustrasi Pembangkitan Nomor Kendaraan dengan Menggunakan Algoritma *Brute Force*



Gambar 5. Contoh 2 Ilustrasi Pembangkitan Nomor Kendaraan dengan Menggunakan Algoritma *Brute Force*

IV. PENENTUAN NOMOR KENDARAAN MENGGUNAKAN ALGORITMA PENCOCOKAN STRING KNUTH-MORRIS-PRATT

Nomor kendaraan tidak hanya berupa kombinasi nomor sesuai jenis kendaraan, urutan dan wilayah pendaftaran. Kombinasi nomor kendaraan juga dapat disusun sesuai dengan keinginan masyarakat yang mengajukan permohonan kepada pihak yang bersangkutan. Saat pemohon mengajukan kombinasi nomor kendaraan yang diinginkan, keunikan nomor tersebut akan dicek terlebih dahulu. Jika tidak terdapat nomor dengan kombinasi yang sama, maka nomor tersebut diperbolehkan. Jika sudah tersedia atau sudah ada yang memiliki, pemohon diminta untuk mengajukan kombinasi angka yang lain.

Dalam pencarian nomor kendaraan unik yang diinginkan pemohon dalam *database* nomor kendaraan yang dimiliki pihak terkait, algoritma pencocokan *string* dapat digunakan. Dalam hal ini akan digunakan algoritma Knuth-Morris-Pratt dengan mencocokan *pattern* yang berupa kombinasi angka pemohon dan teks yang berupa seluruh daftar nomor kendaraan yang sudah dipakai digabung menjadi sebuah *string* yang panjang. Algoritma Knuth-Morris-Pratt akan menghitung fungsi pinggiran terlebih dahulu yang merupakan ukuran awalan terpanjang dari P[0..j] yang juga merupakan akhiran dari P[1..j]. Misalkan dalam *database* terdapat nomor-nomor kendaraan sebagai berikut.

```
… AA11BA AA11A AA11AA A3213F A813DVA B1358PQC
B1843CV B2313AA B318PV B3840DC C9234XY F1192BV
F2211FF F4352AA Z1632DV …
```

Kemudian terdapat pemohon yang ingin membuat nomor kendaraan dengan kombinasi `AA11AA`. Pertama-tama, kita buat terlebih dahulu tabel fungsi pinggiran sebagai berikut.

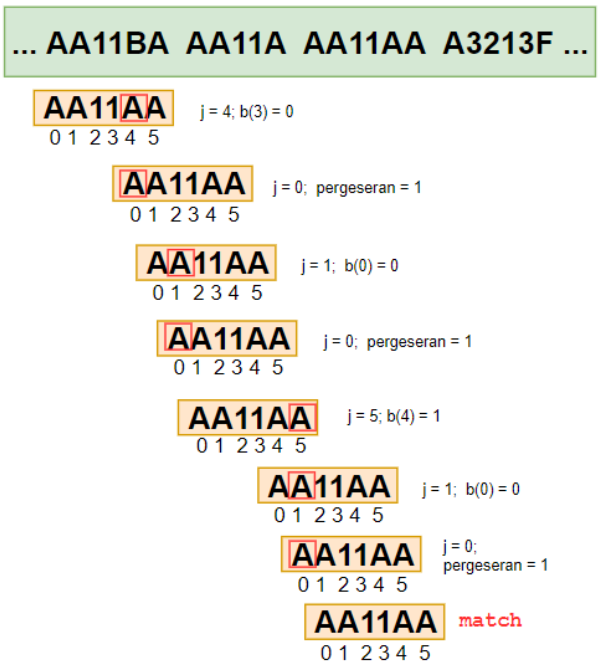
TABEL II. FUNGSI PINGGIRAN

j	0	1	2	3	4	5
P[*j*]	A	A	1	1	A	A
b(*j*)	0	1	0	0	1	2

Dengan *j* = posisi ketidakcocokan pada P[] dan fungsi pinggiran b(*j*). Misalkan kita akan menghitung nilai fungsi pinggiran b(4), maka kita cari *prefix* pada P[0..4] yang juga merupakan *suffix* dari P[1..4]. Fungsi pinggiran b(4) artinya pada `AA11A` terdapat *prefix* `A, AA, AA1, AA11` dan *suffix* `A, 1A, 11A, A11A`, maka nilai *prefix* terpanjang yang juga merupakan *suffix* adalah `A`, yaitu 1 sehingga nilai fungsi pinggiran b(4) adalah 1.

`AA11A = A11A`

Ilustrasi pencocokan *string* menggunakan algortima KMP adalah sebagai berikut.



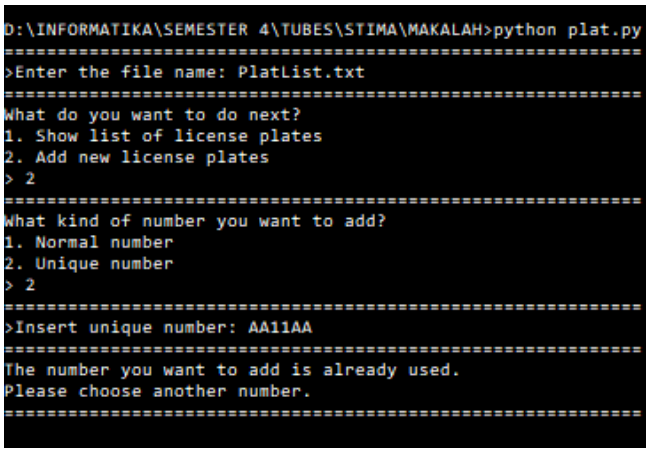
Gambar 6. Ilustrasi Pencocokan *String* dengan KMP

Saat `AA11AA` dicocokan dengan *string* bagian `AA11BA_AA11AA_AA11AA` akan ditemukan ketidakcocokan pada *j* = 4, kemudian kita periksa pada tabel fungsi pinggiran nilai b(*j-1*) atau b(3) adalah 0, maka pergeseran dilakukan dengan menyesuaikan posisi ketidakcocokan dengan P[b(*j-1*)], atau dapat dikatakan, posisi indeks ke-4 sebelumnya disesuaikan dengan posisi indeks ke 0 untuk pencocokan selanjutnya karena hasil b(3) = 0. Selanjutnya, apabila ketidakcocokan ditemukan pada indeks *j* = 0, maka lakukan pergeseran sebanyak 1 karakter. Lakukan hal yang sama hingga ditemukan kecocokan.

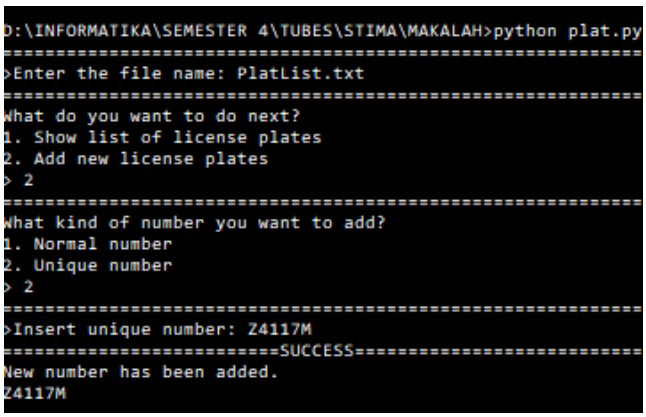
Dalam implementasinya, fungsi pinggiran pada algoritma Knuth-Morris-Pratt akan mengembalikan indeks *pattern* pada teks apabila ditemukan kecocokan, dan akan mengembalikan nilai -1 apabila tidak ditemukan kecocokan. Berikut algoritma fungsi pinggiran (*border function*) atau disebut juga *failure function* (disingkat menjadi *fail*) dan algoritma KMP.

```
def compute_fail(pattern) :
    fail = []
    i = 0
    while i < len(pattern) :
        fail.append(0)
        i += 1
    j = 0
    i = 1

    while i < len(pattern) :
        if pattern[j] == pattern[i] :
            fail[i] = j + 1
            i += 1
            j += 1
        elif j > 0 :
            j = fail[j-1]
        else :
            fail[i] = 0
            i += 1

    return fail

def KMP(text, pattern) :
    fail = compute_fail(pattern)
    i = 0
    j = 0

    while i < len(text) :
        if pattern[j] == text[i] :
            if j == len(pattern) - 1 :
                return i - len(pattern) + 1
            i += 1
            j += 1
        elif j > 0 :
            j = fail[j-1]
        else :
            i += 1

    return -1
```

Untuk dapat mengilustrasikannya, berikut contoh sederhana menggunakan algoritma Knuth-Morris-Pratt untuk membangkitkan nomor kendaraan unik.



Gambar 7. Contoh 1 Ilustrasi Pembangkitan Nomor Kendaraan Unik Menggunakan Algoritma Knuth-Morris-Pratt



Gambar 8. Contoh 2 Ilustrasi Pembangkitan Nomor Kendaraan Unik Menggunakan Algoritma Knuth-Morris-Pratt

V. KESIMPULAN

Tanda Nomor Kendaraan Bermotor atau nomor kendaraan yang wajib dipasang pada bagian depan dan belakang kendaraan bermotor di seluruh Indonesia bukan merupakan nomor acak yang tidak memiliki arti. Setiap karakter dari nomor tersebut disusun berdasarkan ketentuan yang berlaku. Namun, nomor kendaraan tersebut juga dapat disusun sesuai keinginan pemilik kendaraan. Dalam pembangkitan nomor kendaraan sesuai dengan prosedur normal dapat digunakan algoritma *brute force*, yaitu dengan menyusun komponen dari nomor kendaraan satu persatu. Sedangkan dalam pembangkitan nomor kendaraan sesuai dengan keinginan pemohon (nomor kendaraan unik) dapat digunakan algoritma pencocokan *string* Knuth-Morris-Pratt (KMP) yang akan mencari nomor kendaraan unik di dalam *database*. Jika nomor kendaraan unik tersebut belum dipakai, maka pemohonan atas nomor kendaraan unik tersebut dapat dilayani oleh pihak terkait.

REFERENSI

[1] Munir, Rinaldi. 2018. Diktat Kuliah IF2211 Strategi Algoritma. Bandung: Institut Teknologi Bandung.

[2] Daftar Provinsi di Indonesia dan Ibukotanya – 34 atau 36 ? | Pengertian, Sejarah dan Pemekaran | Salamadian. Diakses dari https://salamadian.com/provinsi-di-indonesia/ pada 13 Mei 2018 pukul 14.20.

[3] Inilah Kode Rahasia Dibalik Huruf dan Angka Plat Nomor. Diakses dari https://otomania.gridoto.com/read/03217402/inilah-kode-rahasia-dibalik-huruf-dan-angka-plat-nomor?page=all#!%2F pada 13 Mei 2018 pukul 14.23.

[4] Mengetahui Arti Angka dan Huruf Plat Nomor Kendaraan. Diakses dari https://nuhaminullah.wordpress.com/2015/05/22/mengetahui-arti-angka-dan-huruf-plat-nomer-kendaraan/ pada 13 Mei 2018 pukul 14.24.

[5] Kode Plat Nomor / Tanda Nomor Kendaraan Bermotor di Indonesia. Diakses dari https://ilmupengetahuanumum.com/kode-plat-nomor-tanda-nomor-kendaraan-bermotor-di-indonesia/ pada 13 Mei 2018 pukul 14.27.

[6] Begini Prosedur Memilih Pelat Nomor Cantik. Diakses dari https://news.detik.com/berita/d-3389744/begini-prosedur-memilih-pelat-nomor-cantik pada 13 Mei 2018 pukul 14.29.

[7] KodePlatNomor. Diakses dari https://kodeplat.blogspot.co.id pada 13 Mei 2018 pukul 13.34.

[8] Standford Computer Science. Diakses dari https://www-cs-faculty.stanford.edu/~knuth/dek-14May10-2.jpeg pada 13 Mei 2018 pukul 14.30.

[9] Munir, Rinaldi. 2015. Slide Kuliah IF2211 Strategi Algoritma: Pencocokan String (2015). Bandung: Institut Teknologi Bandung

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, 14 Mei 2018

Nira Rizki Ramadhani 13516018