

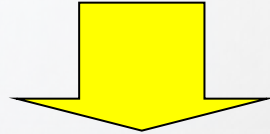
Teori Bahasa dan Operasi Matematis

<http://www.brigidaarie.com>

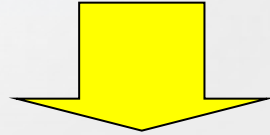
Terminologi Bahasa

- Manfaat bahasa adalah sebagai media komunikasi yang menggunakan sekumpulan simbol dan dikombinasikan menurut aturan sintaksis tertentu (grammar).
- Sementara Semantik bahasa mendefinisikan bagaimana sebuah kalimat dapat diinterpretasikan/diartikan secara benar (sesuai dengan grammar-nya).
- Terminologi penting di dalam memahami teori bahasa adalah pemahaman terhadap alphabet dan string.

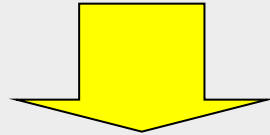
Bahasa = { Kalimat }



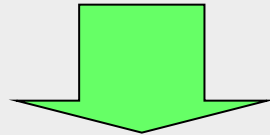
Kalimat = { Kata atau String }



Kata atau String = { Karakter }



Karakter = { Alphabet \cup Digit \cup Symbol }



Bahasa = { { { Alphabet \cup Digit \cup Symbol } } }

Pengertian Dasar

- Simbol adalah sebuah entitas abstrak (seperti halnya pengertian *titik dalam* geometri).
 - Sebuah huruf atau sebuah angka adalah contoh simbol.
- String adalah deretan terbatas (*finite*) *simbol-simbol*.
 - a , b , dan c adalah tiga buah simbol maka $abcb$ adalah sebuah string yang dibangun dari ketiga simbol tersebut.
- Jika w adalah sebuah string maka panjang string dinyatakan sebagai $|w|$ dan didefinisikan sebagai cacahan (banyaknya) simbol yang menyusun string tersebut.
 - Sebagai contoh, jika $w = abcb$ maka $|w| = 4$.

- String hampa adalah sebuah string dengan nol buah simbol. String hampa dinyatakan dengan simbol ϵ (atau \wedge) sehingga $|\epsilon| = 0$. String hampa dapat dipandang sebagai simbol hampa karena keduanya tersusun dari nol buah simbol.
- Alfabet adalah himpunan hingga (*finite set*) simbol-simbol

Operasi Alphabet

Diberikan string : $x = abc$

- Prefik string w adalah string yang dihasilkan dari string w dengan menghilangkan nol atau lebih simbol-simbol paling belakang dari string w tersebut
 - $abc, ab, a, \text{ dan } \varepsilon$ adalah semua Prefix
- ProperPrefix string w adalah string yang dihasilkan dari string w dengan menghilangkan satu atau lebih simbol-simbol paling belakang dari string w tersebut.
 - $ab, a, \text{ dan } \varepsilon$ adalah semua ProperPrefix(x)

Diberikan string : $x = abc$

- Postfix (atau Sufix) string w adalah string yang dihasilkan dari string w dengan menghilangkan nol atau lebih simbol-simbol paling depan dari string w tersebut.
 - $abc, bc, c, \text{ dan } \varepsilon$ adalah semua $Postfix(x)$
- ProperPostfix (atau PoperSufix) string w adalah string yang dihasilkan dari string w dengan menghilangkan satu atau lebih simbol-simbol paling depan dari string w tersebut.
 - $bc, c, \text{ dan } \varepsilon$ adalah semua $ProperPostfix(x)$

Diberikan string : $x = abc$

- Head string w adalah simbol paling depan dari string w .
 - a adalah $Head(x)$
- Tail string w adalah string yang dihasilkan dari string w dengan menghilangkan simbol paling depan dari string w tersebut.
 - bc adalah $Tail(x)$

Diberikan string : $x = abc$

- Substring string w adalah string yang dihasilkan dari string w dengan menghilangkan nol atau lebih simbol-simbol paling depan dan/atau simbol-simbol paling belakang dari string w tersebut.
 - $abc, ab, bc, a, b, c,$ dan ε adalah semua $Substring(x)$
- ProperSubstring string w adalah string yang dihasilkan dari string w dengan menghilangkan satu atau lebih simbol-simbol paling depan dan/atau simbol-simbol paling belakang dari string w tersebut.
 - $ab, bc, a, b, c,$ dan ε adalah semua $Substring(x)$

Diberikan string : $x = abc$

- Subsequence string w adalah string yang dihasilkan dari string w dengan menghilangkan nol atau lebih simbol-simbol dari string w tersebut.
 - $abc, ab, bc, ac, a, b, c,$ dan ε adalah semua $Subsequence(x)$
- ProperSubsequence string w adalah string yang dihasilkan dari string w dengan menghilangkan satu atau lebih simbol-simbol dari string w tersebut.
 - $ab, bc, ac, a, b, c,$ dan ε adalah semua $Subsequence(x)$

Diberikan dua string : $x = abc$, dan $y = 123$

- Concatenation adalah penyambungan dua buah string. Operator concatenation adalah *concate* atau *tanpa lambang apapun*.
 - $\text{concate}(xy) = xy = abc123$
- Alternation adalah pilihan satu di antara dua buah string. Operator alternation adalah *alternate* atau $|$.
 - $\text{alternate}(xy) = x|y = abc \text{ atau } 123$

- Kleene Closure : $x^* = \epsilon | x | xx | xxx | \dots = \epsilon | x | x^2 | x^3 | \dots$
- Positive Closure : $x^+ = x | xx | xxx | \dots = x | x^2 | x^3 | \dots$

Misalkan terdapat 2 himpunan karakter sebarang L dan M. Maka operasi² yang dapat dilakukan terhadap kedua himpunan tersebut antara lain adalah :

No	Nama Operasi	Simbol	Keterangan
1	UNION	$L \cup M$	$\{s \mid s \text{ ada di } L \text{ atau } M\}$
2	CONCATENATION	LM	$\{st \mid s \text{ ada di } L \text{ dan } t \text{ ada di } M\}$
3	KLEENE CLOSURE	L^*	$L^* = \cup_{i=0}^{\infty} L^i$ (penggabungan nol atau lebih L)
4	POSITIVE CLOSURE	L^+	$L^+ = \cup_{i=1}^{\infty} L^i$ (penggabungan satu atau lebih L)
5	REVERSE of STRING	$Rev(x)$	Sebuah string x yang ditulis dalam urutan terbalik
6	LENGTH of STRING	$Length(x)$	Menghitung jumlah karakter pada sebuah string x
7	PALINDROME	$x = Rev(x)$	Rangkaian karakter dalam sebuah string x yang ditulis dalam urutan terbalik tetap menghasilkan string x

Sifat Operasi

- Tidak Selalu Berlaku
 - $x = \text{Prefix}(x)\text{Postfix}(x)$
 - $\text{Prefix}(x) = \text{Postfix}(x)$ atau $\text{Prefix}(x) \neq \text{Postfix}(x)$
- Selalu berlaku
 - $x = \text{Head}(x)\text{Tail}(x)$
 - $\text{ProperPrefix}(x) \neq \text{ProperPostfix}(x)$
 - $\text{Head}(x) \neq \text{Tail}(x)$
- Setiap $\text{Prefix}(x)$, $\text{ProperPrefix}(x)$, $\text{Postfix}(x)$, $\text{ProperPostfix}(x)$, $\text{Head}(x)$, dan $\text{Tail}(x)$ adalah $\text{Substring}(x)$, tetapi tidak sebaliknya
- Setiap $\text{Substring}(x)$ adalah $\text{Subsequence}(x)$, tetapi tidak sebaliknya

- Dua sifat aljabar concatenation
 - Operasi concatenation bersifat asosiatif : $x(yz) = (xy)z$
 - Elemen identitas operasi concatenation adalah ε : $\varepsilon x = x\varepsilon = x$
- Tiga sifat aljabar alternation
 - Operasi alternation bersifat komutatif : $x|y = y|x$
 - Operasi alternation bersifat asosiatif : $x|(y|z) = (x|y)|z$
 - Elemen identitas operasi alternation adalah dirinya sendiri : $x|x = x$
- Sifat distributif concatenation terhadap alternation :
 $x(y|z) = xy|xz$

- Beberapa Kesamaan

- Kesamaan ke-1 : $(x^*)^* = (x^*)$

- Kesamaan ke-2 : $\varepsilon | x^+ = x^+ | \varepsilon = x^*$

- Kesamaan ke-3 : $(x|y)^* = \varepsilon | x|y|xx|yy|xy|yx|... = \textit{semua string}$
yang merupakan concatenation dari nol atau lebih x , y ,
atau keduanya.

Misal terdapat sebuah himpunan alphabet $\Sigma = \{x\}$

dan misalkan akan didefinisikan sebuah bahasa L_1

$$L_1 = \{ x, xx, xxx, xxxx, \dots \}$$

Maka L_1 dapat dinyatakan secara formal sebagai

$$L_1 = \{ x^n, \text{ untuk } n = 1, 2, 3, \dots \}$$

Atau, didefinisikan sebuah bahasa L_2

$$L_2 = \{ x, xxx, xxxxx, \dots \}$$

secara formal, L_2 dapat dinyatakan sebagai

$$L_2 = \{ x^n, \text{ untuk } n = 1, 3, 5, \dots \}$$

Operasi Bahasa

Contoh :

Misal terdapat himpunan string

$S = \{ a, aa, aaa \}$ dan $T = \{ bb, bbb \}$

Maka,

$S \cup T = \{ a, aa, aaa, bb, bbb \}$

$ST = \{ abb, abbb, aabb, aabbb, aaabb, aaabbb \}$

Contoh :

Misal terdapat sebuah himpunan alphabet $\Sigma = \{ 0, 1 \}$

Maka $\Sigma^* = \{ \lambda, 0, 1, 00, 01, 10, 11, 000, \dots \}$

Contoh :

Misal terdapat sebuah himpunan alphabet $\Sigma = \{ x \}$

Maka $\Sigma^+ = \{ x, xx, xxx, \dots \}$

Definisi Rekursif

- Secara umum, definisi rekursif dapat diartikan sebagai upaya untuk mendefinisikan sebuah himpunan melalui penerapan (sebuah/ sekelompok) aturan secara berulang-ulang.
- Mendefinisikan himpunan melalui **Definisi Rekursif** akan melibatkan *three-steps process*, yaitu:
 1. Menentukan **obyek dasar** dari himpunan yang akan dibentuk
 2. Menentukan **aturan²** untuk membangkitkan obyek² lain dalam himpunan tersebut
 3. Mendeklarasikan bahwa tidak ada obyek² lain yang dapat dibentuk, kecuali menggunakan aturan di atas.

Misalkan kita akan mendefinisikan himpunan bilangan genap

- Melalui definisi rekursif, masalah di atas dapat didekati dengan tahapan penyelesaian seperti berikut :
 1. 2 adalah anggota himpunan bilangan genap (*2 adalah sebuah obyek dasar*)
 2. Jika x adalah anggota dalam himpunan tersebut, maka $x + 2$ juga merupakan anggota himpunan bilangan genap (*$x + 2$ adalah aturan yang dapat memproduksi anggota himpunan bilangan genap yang lain seperti 4, 6, 8, dst*)
 3. Jika y adalah anggota himpunan bilangan genap, maka y pasti dihasilkan dari aturan di atas (*deklarasi bahwa $x + 2$ adalah satu²-nya aturan untuk memproduksi elemen himpunan bilangan genap*)

Contoh :

Misal terdapat bahasa $L_1 = \{ \lambda, x, xx, xxx, \dots \}$. Melalui Definisi Rekursif, bahasa L_1 dapat dinyatakan melalui cara berikut :

1. λ adalah anggota L_1
2. Jika Q adalah sembarang string di L_1 , maka xQ juga merupakan string dalam L_1

Contoh :

Misal terdapat bahasa $L_2 = \{ x, xxx, xxxxx, \dots \}$. Melalui Definisi Rekursif, bahasa L_2 dapat dinyatakan melalui cara berikut :

1. x adalah anggota L_2
2. Jika Q adalah sembarang string di L_2 , maka xxQ juga merupakan string dalam L_2

Metode Pendefinisian Bahasa

- Jumlah alphabet dan digit memang berhingga (*finite*). Tetapi, jumlah string/kata dan kalimat yang dapat dibentuk dari kombinasi alphabet dan digit tersebut bisa tak berhingga (*infinite*) !!!
- Oleh karenanya, untuk mendefinisikan sebuah bahasa, umumnya tidak dilakukan dengan cara *me-listing* semua string dan kalimat yang dimiliki oleh bahasa tersebut, melainkan dengan mengemukakan syarat² yang dimiliki oleh bahasa yang bersangkutan.
- Dengan kata lain, karena bahasa adalah suatu bentuk himpunan, maka cara mengekspresikan himpunan yang paling praktis adalah melalui set theoretic notation.

Metode untuk mendefinisikan bahasa secara berhingga (untuk bahasa yang tidak berhingga) adalah melalui :

- **Grammar**

kita dapat membangun sebuah kalimat dengan sintaksis yang benar sesuai dengan kaidah yang telah ditetapkan pada serangkaian aturan yang disebut **production(s)**

- **Recognizer** atau **Finite Automata**

diberikan sebuah input string, maka recognizer akan melakukan penelusuran karakter per karakter untuk mengetahui apakah input string tersebut merupakan anggota suatu bahasa tertentu atau tidak

Namun dalam kenyataannya, kedua metode tersebut sebenarnya teraplikasi untuk tujuan yang berbeda :

- **Grammar**

lebih berfungsi sebagai pembangkit string dan sentence.

- **Recognizer** atau **Finite Automata**

sesuai namanya, recognizer lebih berfungsi sebagai pengenal string atau sentence.

Latihan

1. Misal terdapat bahasa S^* dengan $S = \{ a, b \}$ terdapat berapa banyak kata dalam bahasa di atas yang memiliki :
 - a. Length(2)
 - b. Length(3)
 - c. Length(n)
2. Misalkan terdapat sebuah himpunan string $S = \{ a, bb, bab, abaab \}$
 - a. Apakah $abbabaabab$ dan $abaabbabbaabb$ terdapat dalam S^* ?
 - b. Adakah string pada S^* yang memiliki karakter b berjumlah ganjil ?