Parsing dengan Definite clause grammar

[0] Definite Clause Grammar atau bisa disingkat DCG merupakan sebuah cara untuk mengekspresikan sebuah tatabahasa, DCG dibuat pertama kali untuk prolog system, karena salah satu tujuan prolog dubuat pertama kali ialah untuk melakukan proses pada bahasa natural.

ketika kita membahas mengenai tatabahasa, kita juga akan bertemu dengan notasi [1] BNF, BNF atau Backus–Naur form, BNF digunakan untuk mendeskripsikan syntax dari sebuah bahasa program, saya akan menjelaskan kenapa DCG berhubungan dengan BNF

```
a → "".
a → "a", a.
```

kode diatas adalah contoh sederhana dcg, syntax dcg dalam prolog mirip saat kita membuat rule, syntax dcg adalah "Head → Body", dalam "Body" bisa berupa terminal, nonterminal dan grammar goals, jika kita baca kode prolog diatas:

"rule pertama adalah str kosong, rule kedua adalah string "a" dan sebuah str"

ketika kita menjalankan kode diatas, prolog system akan seperti ini:

```
?- a(X, []).
%@    X = []
%@ ;  X = "a"
%@ ;  X = "aa"
%@ ;  X = "aaa"
%@ ;  X = "aaaa"
%@ ;  ... .
```

Prolog system akan mencari semua cara agar rulenya adalah benar, dari kode prolog diatas saya bisa langsung mengubahnya ke notasi BNF atau sebaliknya, seperti inilah notasi BNF:

```
<a> ::= "a" ( <a> | " " )
```

Dengan kekuatan ini, DCG merupakan sebuah language feature yang sangat kuat, karna kita bisa melakukan parsing dengan mudah.

untuk contoh saya akan menggunakan kata–kata mudah "lahir tahun 2006", untuk DCG seperti inilah kodenya

```
kalimat     → frasa_kerja, spasi, frasa_waktu.
frasa_kerja → "lahir".
spasi       → " ".
frasa_waktu → "tahun", tahun.
tahun       → "2006".
```

outputnya adalah seperti ini:

```
?- kalimat(X, []).
%@    X = "lahir tahun 2006".
```

seperti sebelumnya Prolog system akan mencari variable X apa agar rulenya benar disini nilai X adalah "lahir tahun 2006", kita bisa langsung mengkonfersinya ke Backus-Naur form.

```
<kalimat>     ::= <frasa_kerja> <spasi> <frasa_waktu>
<frasa_kerja> ::= "lahir"
<frasa_waktu> ::= "tahun" <spasi> <tahun>
<tahun>       ::= "2000" | "2001" | ... | "2099"
<spasi>       ::= " "
```

cukup untuk perkenalanya, kita langsung ke parsing, selanjutnya saya akan mencoba memparsing nomor tahun pada kode prolog sebelumnya, berekut kode yang sudah saya modifikasi:

```
kalimat(Tahun)     → frasa_kerja, spasi, frasa_waktu(Tahun).
frasa_kerja        → "lahir".
spasi              → " ".
frasa_waktu(Tahun) → "tahun", spasi, tahun(Tahun).
tahun([D|Ds])      → [D], tahun(Ds).
tahun([])          → [].
```

seperti inilah ketika saya menjalankan kode diatas:

```
?- kalimat(X, "lahir tahun 2006", []).
%@    X = "2006"
%@ ;  false
```

setelah saya mempelajari DCG saya membuat aplikasi sederhana, yaitu kalkulator sederhana dengan fitur interpreter seperti bahasa program python dkk, dan saya juga membuat aplikasi truth table generator, dimana saya bisa membuat tabel dengan input sebuah expresi boolean algebra.

trimakasih telah membaca sampai akhir, bila ada tulisan yang salah mohon maaf atau bisa menghubungi saya via email, sampai jumpa.

Referensi

[0] *Definite Clause Grammar* https://en.wikipedia.org/wiki/Definite_clause_grammar

[1] *Backus Naur Form* https://en.wikipedia.org/wiki/Backus-Naur_form