

Lengkapilah parser yang telah anda buat dengan code generator untuk bagian :

1. deklarasi variabel pada outblock (global variable)
2. compound statement (begin – end statement)
3. assignment statement
4. expression (termasuk block term dan block factor)

sehingga dapat mengompilasi program dan menghasilkan kode dalam instruksi mesin objek ST4469. Untuk jelasnya, lihat lagi syntax graph pada lembar praktikum 5-6.

Deklarasi variabel

Source code	Object code
var i,j,k;	int 0 0 3

Catatan : untuk setiap identifier yang merupakan variabel, dialokasikan 1 (satu) area di stack memory, dan alamat area tersebut teregister pada symbol table.

Assignment statement

Source code	Object code
a := expression;	Code untuk <i>expression</i> sto b i a

Testing Code :

Input Source	Output
<pre> program example111; var i,j,k; k:= i + j. </pre>	<pre> int 0 0 3 lod 0 0 0 lod 0 0 1 opr 0 0 1 sto 0 0 2 hlt 0 0 0 </pre>
<pre> program example112; var i,j,k; begin i := 20; j := 40; k := (30 + i) * (80 -j div 4) end. </pre>	<pre> int 0 0 3 lit 0 0 20 sto 0 0 0 lit 0 0 40 sto 0 0 1 lit 0 0 30 lod 0 0 0 opr 0 0 1 lit 0 0 80 lod 0 0 1 lit 0 0 4 opr 0 0 4 opr 0 0 2 opr 0 0 3 sto 0 0 2 hlt 0 0 0 </pre>

Lampiran : Parser expression yang sudah ditambahkan code generator.

```
#define PLUSOP 0
#define MINUSOP 1

Void expression(void)
{
    int sign = PLUSOP;

    if(isminus){
        sign = MINUSOP;
        get_token();
    } else if(isplus) get_token();

    term();

    if (sign == MINUSOP) gencode(OPR,0,0,0);

    while(isplus||isminus) {
        if(isplus) sign = PLUSOP;
        else sign = MINUSOP;

        get_token();
        term();

        if(sign==MINUSOP) gencode(OPR,0,0,2);
        else gencode(OPR,0,0,1);
    }
}
```