Grammatica BNF di Simpla

program → *var-decl-list func-decl-list body* **.**
var-decl-list → *var-decl var-decl-list* | **ε**
var-decl → *id-list* **:** *type* **;**
id-list → **id ,** *id-list* | **id**
type → **integer** | **real** | **string** | **boolean** | **void**
func-decl-list → *func-decl func-decl-list* | **ε**
func-decl → **func id (** *opt-param-list* **) :** *type var-decl-list body* **;**
opt-param-list → *param-list* | **ε**
param-list → *param-decl* **,** *param-list* | *param-decl*
param-decl → **id :** *type*
body → **body** *stat-list* **end**
stat-list → *stat* **;** *stat-list* | *stat* **;**
stat → *assign-stat* | *if-stat* | *while-stat* | *for-stat* | *return-stat* | *read-stat* | *write-stat* | *func-call* | **break**
assign-stat → **id =** *expr*
if-stat → **if** *expr* **then** *stat-list opt-else-stat* **end**
opt-else-stat → **else** *stat-list* | **ε**
while-stat → **while** *expr* **do** *stat-list* **end**
for-stat → **for id =** *expr* **to** *expr* **do** *stat-list* **end**
return-stat → **return** *opt-expr*
opt-expr → *expr* | **ε**
read-stat → **read (** *id-list* **)**
write-stat → *write-op* **(** *expr-list* **)**
write-op → **write** | **writeln**
expr-list → *expr* **,** *expr-list* | *expr*
expr → *expr logic-op bool-term* | *bool-term*
logic-op → **and** | **or**
bool-term → *rel-term rel-op rel-term* | *rel-term*
rel-op → **==** | **!=** | **>** | **>=** | **<** | **<=**
rel-term → *rel-term low-prec-op low-term* | *low-term*
low-prec-op → **+** | **-**
low-term → *low-term high-prec-op factor* | *factor*
high-prec-op → ***** | **/**
factor → *unary-op factor* | **(** *expr* **)** | **id** | *const* | *func-call* | *cond-expr* | *cast* **(** *expr* **)**
unary-op → **-** | **not**
const → **intconst** | **realconst** | **strconst** | **boolconst**
func-call → **id (** *opt-expr-list* **)**
opt-expr-list → *expr-list* | **ε**
cond-expr → **if** *expr* **then** *expr* **else** *expr* **end**
cast → **integer** | **real**