<program> ::= begin <stmt\_list> end

<stmt\_list> ::= <stmt> | <stmt> ; <stmt\_list> | <stml> ;

<stmt> ::= <var> := <expression>

<var> ::= A | B | C | **<var> + <var> | <var> - <var>**

<expression> ::= <var> + <var> | <var> - <var> | <var>

Q : turunkan perintah ini “begin A := B + C – B ; B := C + A end” dari simbol awal <program>

A :

<program> → begin <stmt\_list> end rule 1

→ begin <stmt> ; <stmt\_list> end rule 2

→ begin <var> := <expression> ; <stmt\_list> end rule 3

→ begin <var> := <var> + <var> ; <stmt\_list> end rule 5

→ begin A := <var> + <var> ; <stmt\_list> end rule 4

→ begin A := B + <var> ; <stmt\_list> end rule 4

→ begin A := B + C ; <stmt\_list> end rule 4

→ begin A := B + C ; <stmt> end rule 2

→ begin A := B + C ; <var> := <expression> end rule 3

→ begin A := B + C ; <var> := <var> + <var> end rule 5

→ begin A := B + C ; B := <var> + <var> end rule 3

→ begin A := B + C ; B := C + <var> end rule 3

→ begin A := B + C ; B := C + A end rule 3

Q tidak ditemukan sehingga syntax error.

Q : begin A := B + B ; B := C + A; end

A :

<program> → begin <stmt\_list> end rule 1

→ begin <stmt> ; <stmt\_list> end rule 2

→ begin <var> := <expression> ; <stmt\_list> end rule 3

→ begin <var> := <var> + <var> ; <stmt\_list> end rule 5

→ begin A := <var> + <var> ; <stmt\_list> end rule 4

→ begin A := B + <var> ; <stmt\_list> end rule 4

→ begin A := B + B ; <stmt\_list> end rule 4

→ begin A := B + B ; <stmt> end rule 2

→ begin A := B + B ; <var> := <expression> end rule 3

→ begin A := B + B ; <var> := <var> + <var> end rule 5

→ begin A := B + B ; B := <var> + <var> end rule 3

→ begin A := B + B ; B := C + <var> end rule 3

→ begin A := B + B ; B := C + A end rule 3

Semicolon (;) tidak ditemukan, sehingga syntax error.

<program> ::= begin <stmt\_list> end rule 1

<stmt\_list> ::= <stmt> | <stmt> ; <stmt\_list> | <stml> ; rule 2

<stmt> ::= <var> := <expression> rule 3

<var> ::= A | B | C <var> ::= A | B | C | **<var> + <var> | <var> - <var>** rule 4

<expression> ::= <var> + <var> | <var> - <var> | <var> rule 5

Q : begin A := B + B ; B := C + A; end

A :

<program> → begin <stmt\_list> end rule 1

→ begin <stmt> ; <stmt\_list> end rule 2

→ begin <stmt> ; <stmt> ; end rule 2

→ begin <var> := <expression> ; <stmt> ; end rule 3

→ begin <var> := <expression> ; <var> := <expression> ; end rule 3

→ begin B := <expression> ; <var> := <expression> ; end rule 4

→ begin B := <var> + <var> ; <var> := <expression> ; end rule 5

→ begin B := B : <var> ; <var> := <expression> ; end rule 4

→ begin B := B + B ; <var> := <expression> ; end rule 4

→ begin B := B + B ; <var> := <var> + <var> ; end rule 5

→ begin B := B + B ; B := <var> + <var> ; end rule 4

→ begin B:= B + B ; B := C + <var> ; end rule 4

→ begin B:= B + B; B := C + A ; end rule 4

Q : begin A := B + B ; B := C + A ; end

A : begin B := B + B ; B := C + A ; end

Karena simbol awal dapat diturunkan menjadi program, maka tidak ada syntax error.

<program> ::= begin <stmt\_list> end rule 1

<stmt\_list> ::= <stmt> | <stmt> ; <stmt\_list> | **<stml> ;** rule 2

<stmt> ::= <var> := <expression> rule 3

<var> ::= A | B | C <var> ::= A | B | C | **<var> + <var> | <var> - <var>** rule 4

<expression> ::= <var> + <var> | <var> - <var> | <var> rule 5

Q : begin A := B + C – B + A – A ; B := C + A ; end

A :

<program> → begin <stmt\_list> end rule 1

→ begin <stmt> ; <stmt\_list> end rule 2

→ begin <stmt> ; <stmt>; end rule 2

→ begin <var> := <expression> ; <stmt> ; end rule 3

→ begin <var> := <var> + <var> ; <stmt> ; end rule 5

→ begin <var> := <var> + <var> - <var> ; <stmt> ; end rule 4

→ begin <var> := <var> + <var> - <var> + <var> ; <stmt> ; end rule 4

→ begin <var> := <var> + <var> - <var> + <var> - <var> ; <stmt> ; end rule 4

→ begin A := <var> + <var> - <var> + <var> - <var> ; <stmt> ; end rule 4

→ begin A := B + <var> - <var> + <var> - <var> ; <stmt> ; end rule 4

→ begin A := B + C - <var> + <var> - <var> ; <stmt> ; end rule 4

→ begin A := B + C - B + <var> - <var> ; <stmt> ; end rule 4

→ begin A := B + C - B + A - <var> ; <stmt> ; end rule 4

→ begin A := B + C - B + A - A ; <stmt> ; end rule 4

→ begin A := B + C - B + A - A ; <var> := <expression> ; end rule 3

→ begin A := B + C - B + A - A ; B := <expression> ; end rule 4

→ begin A := B + C - B + A - A ; B := <var> + <var> ; end rule 5

→ begin A := B + C - B + A - A ; B := C + <var> ; end rule 4

→ begin A := B + C - B + A - A ; B := C + A ; end rule 4

A : begin A := B + C - B + A - A ; B := C + A ; end

Q : begin A := B + C – B + A – A ; B := C + A ; end