**Left Factored & Recursive CFGs**

**Rexton Keys:**

<IF\_Statement>🡪 agar

<IF\_ELSE>🡪 agar Nahi To

<ELSE>🡪warna

<WHILE\_Statement>🡪 Jab tak

<FOR\_Loop>🡪 Bar Bar

**General Rules:**

<CONST>🡪 INT\_CONST| FLOAT\_CONST | STRING\_CONST | CHAR\_CONST | BOOL\_CONST

<Static>🡪 Static

<ID\_Constant> 🡪 ID | <CONST>

<Access\_Modifier>🡪access\_modifier | Null

<Return\_Type>🡪void | DT

<M\_ST>🡪<S\_ST><M\_ST> | Null

<Body>🡪 ; | <S\_ST> | {<M\_ST>}

<List\_Param>🡪 DT ID <List\_Param1> | Null

<List\_Param1>🡪 , DT ID <List\_Param1> | Null

//<Param>🡪 <Exp> <Param1> | Null

<Param1>🡪 , <Exp> <Param1> | Null

<Method\_Call\_1>🡪 (<Param>)

**Single Statement:**

<S\_ST>🡪<Jab\_Tak> | DT <S\_St\_DT> | <Bar\_Bar> | <agar\_warna> | <Return> | inc\_dec ID<inc\_dec\_list>;|ID <S\_St\_ID>| <break> | <continue> |<this>

<S\_St\_ID>🡪inc\_dec; | <Assign\_Op>| <Object\_link> | <Object\_Call>; | <Method\_Call\_1>; | [<Exp>] <Assign\_Op>

<S\_St\_DT>🡪ID <S\_St\_DT2> | <Method\_DEC> | <Array\_DEC>

<S\_St\_DT2>🡪<Variable\_Link2>

**Variable Declaration:**

<DEC>🡪DT<Variable\_Link>

<Variable\_Link>🡪 ID <Varaiable\_Link2>

<Variable\_Link2>🡪AOP <Variable\_Value>| <LIST**>**

<Variable\_Value>🡪<Exp><LIST>

<LIST>🡪 , ID<Variable\_Link2> | ;

**Assignment of Variable:**

<Assign\_Op>🡪 AOP <Assign\_Op2>

<Assign\_Op2>🡪<Exp>;

**If Else Condition:**

<agar\_warna>🡪 agar (<Exp>) {<M\_ST>} <O\_Else>

<O\_Else>🡪warna {<M\_ST>} | Null

**While Statement:**

<Jab\_tak>🡪jabtak (<Exp>) <Body>

**Return:**

<Return>🡪 return <Return2>

<Return2> 🡪 ; | <Exp>;

**Break:**

<Break>🡪break ;

**Continue:**

<Continue>🡪continue ;

**This:**

<this>🡪 this.ID <LIST1>

<LIST1>🡪 ; | AOP <LIST2>

<LIST2>🡪 ID <INIT> ; | <CONST> ;

**Class Declaration:**

<Class\_Dec>🡪<Access\_Modifier><Class\_Link>

<Class\_Link>🡪 class ID<Class\_Base> {<Class\_Body>}

<Class\_Base>🡪 Null | : ID

<Class\_Body>🡪<Class\_Member><Class\_Body> | Null

<Class\_Member>🡪<Access\_Modifier><Member\_Link>

<Member\_Link>🡪<Static><SS\_A>| void ID <Method\_Link 3> | DT<DT\_2>|ID <Object\_Constructor\_DEC> | <Class\_Link>

<Object\_Constructor\_DEC> 🡪 <object\_link> | <Constructor\_DEC>

<DT\_2>🡪ID <ID\_1>|<Array\_DEC>

<ID\_1>🡪<Varaiable\_Link2> | <Method\_Link 3>

<SS\_A>🡪 DT <DT\_1> |ID<Id\_OArray> |void ID<Method\_Link3>

<DT\_1>🡪ID<ID\_2>| <Array\_DEC>

<ID\_2>🡪<Method\_Link3> | <Variable\_Link2>

<Id\_OArray>🡪ID<Id\_A> | <Array\_DEC>

<DT\_A>🡪<Variable\_Link2>|<Method\_Link3>

<Id\_A>🡪<Method\_Link3> | <Object\_Creation\_Exp>

**Constructor Declaration:**

<Constructor\_DEC>🡪 (<List\_Param>) {<M-St>}

**Array Declaration:**

<Array\_DEC>🡪 [] ID <INIT\_Array>

**<INIT\_Array>🡪 ; | = new DT [<ID\_Const>]<Array\_const>**

<Array\_const>🡪<Array\_C>| ;

<Array\_C>🡪{ <Exp><Array\_C2>

<Array\_C2>🡪 , <Exp><Array\_C2> | } ;

**Method Declaration:**

<Method\_DEC>🡪<Return\_Type> ID<Method\_Link 3>

<Method\_Link 3>🡪 (<List\_Param>) {<M\_St>}

**Object Declaration:**

<Object\_Link>🡪 ID <Object\_Creation\_Exp>| [] ID <object\_array\_dec>

<object\_array\_dec> 🡪 = new ID[<Exp>]<obj\_arr\_dec1>

<obj\_arr\_dec1> 🡪 ;| {<obj\_arr\_dec2>

<obj\_arr\_dec2> 🡪 new ID (<Param>)<obj\_arr\_dec3>

<obj\_arr\_dec3> 🡪 , <obj\_arr\_dec2>|};

<Object\_Creation\_Exp>🡪 = new ID (<Param>) <Object\_List>

<Object\_List>🡪 , ID<Object\_Creation\_Exp> | ;

**Object Calling:**

<Object\_Call>🡪 . <Exp> | [<Exp>].<Exp>

**FOR Loop:**

<Bar\_Bar>🡪barbar(<F1>; <F2>; <F3>) <Body>

<F1>🡪<DEC> |ID<Assign\_Op> | Null

<F2>🡪<Exp><X> | Null

<X>🡪 , <Exp><X> | Null

<F3>🡪inc\_dec ID | ID <F4>| Null

<F4>🡪inc\_dec | AOP <Exp>

**Expression:**

<Exp>🡪<OR\_Exp>

<OR\_Exp>🡪<AND\_Exp><OR\_Exp2>

<OR\_Exp2>🡪 || <AND\_Exp><OR\_Exp2> | Null

<AND\_Exp>🡪<ROP><AND\_Exp2>

<AND\_Exp2>🡪&&<ROP><AND\_Exp2> | Null

<ROP>🡪<E><ROP2>

<ROP2>🡪 ROP <E><ROP2> | Null

<E>🡪<T><E2>

<E2 >🡪Plus\_Minus<T ><E2> | Null

<T>🡪<F><T2>

<T2>🡪 M\_D\_M <F><T2> | Null

<F>🡪 ID <id\_op> |<Const> |!<F> | (<Exp>) | Inc\_Dec ID<inc\_dec\_list>

<inc\_dec\_list> 🡪 [<Exp>] | .ID[<Exp>] |Null

<id\_op> 🡪 Null | <Method\_Call\_1> | [ <Exp> ] |<Member\_exp> | Inc\_Dec

<Member\_exp> -> .ID < Member\_exp\_2>

f