

**Laporan Tugas Besar**  
**IF3170 Intelegensi Buatan**  
**CLIPS Predict Breast Cancer**



**Disusun oleh:**  
**Kelompok WakaranAI**

Rhapsodya Piedro Asmorobangun	/ 13519084
Benidictus Galih Mahar Putra	/ 13519159
Made Kharisma Jagaddhita	/ 13519176

## Daftar Isi

<b>Daftar Isi</b>	<b>2</b>
<b>Algoritma Program</b>	<b>3</b>
Root Node	3
Branch Node	3
Leaf Node	3
Conclusion Rule	3
<b>Fakta dan Rules dalam Program</b>	<b>4</b>
Fakta	4
Tabel 2.1.1. Deskripsi dari Facts dalam Program CLIPS Predict Breast Cancer	4
Rules	4
Tabel 2.2.1. Deskripsi dari Rules dalam Program CLIPS Predict Breast Cancer	4
<b>Tabel Kontribusi</b>	<b>10</b>
<b>Lampiran</b>	<b>10</b>

## 1. Algoritma Program

### 1.1. Root Node

Dalam implementasinya, root node tidak memiliki LHS untuk diaktifkan. Root node ini meminta masukan dari pengguna dan nilai masukan dari pengguna akan ditambahkan sebuah fakta baru dengan nama node saat itu. Fakta baru ini akan mengaktifkan node child yang berada dibawahnya.

```
# <no conditional required>
then input from user
add fact <current node> <input user>
```

### 1.2. Branch Node

Dalam implementasinya, RHS branch node mirip seperti RHS root node. Namun, LHS branch node diberikan kondisional berupa nama parent dari node sekarang dan nilainya terdapat dalam fakta dan kondisional simpul dengan melakukan perbandingan nilai masukan pengguna dengan *constraint* dari simpul dalam pohon.

```
if (<parent node> <value>) and compare (<value> <constraint node>)
then input from user
add fact <current node> <input user>
```

### 1.3. Leaf Node

Dalam implementasinya, LHS leaf node mirip seperti LHS branch node. Namun, RHS leaf node diisi dengan menambahkan fakta prediction dengan nilai sesuai dengan *constraint* pada daun pohon. Fakta ini akan mengaktifkan *conclusion rule*.

```
if (<parent node> <value>) and compare (<value> <constraint node>)
then add fact (prediction <0/1>)
```

### 1.4. Conclusion Rule

Dalam implementasinya, LHS conclusion rule memerlukan fakta prediction dengan valuenya dan fakta dari deffacts yang berupa pasangan result - value dari prediction - hasil prediksi. Kemudian, RHS conclusion ini akan menampilkan ke layar pengguna hasil dari prediksi dengan menggunakan variabel *hasil prediksi* dari pasangan *result - value dari prediction - hasil prediksi*

```
if (prediction <value>) and (result <value> <outcome>)
then print ("Hasil = " <outcome> "...")
```

## 2. Fakta dan Rules dalam Program

### 2.1. Fakta

Berikut merupakan fakta yang dipakai dalam program CLIPS yang disusun.

Tabel 2.1.1. Deskripsi dari Facts dalam Program CLIPS Predict Breast Cancer

Fakta	Deskripsi
result 0 "Tidak Terprediksi"	result 0 "Tidak Terprediksi" digunakan untuk menyimpan string "Tidak Terprediksi" yang akan dicetak dalam rule "conclusion" jika hasil prediksi yang didapatkan dari program adalah 0.
result 1 "Terprediksi"	result 1 "Terprediksi" digunakan untuk menyimpan string "Terprediksi" yang akan dicetak dalam rule "conclusion" jika hasil prediksi yang didapatkan dari program adalah 1.
<node name>	Fakta <node name> merupakan fakta dari setiap node yang dibentuk dalam tree. <node name> ini digunakan untuk mengaktifkan node child dari current node.
prediction <value>	Fakta prediction <value> digunakan untuk mengaktifkan rule kesimpulan dan mengambil nilai <value> sebagai pembanding dengan value fakta result. Nilai <value> diisi dengan 0 atau 1.

### 2.2. Rules

*Rules* yang ada pada program yang dibuat sama seperti yang ada pada bagian Algoritma Program. Terdapat beberapa jenis dari rule yang dibuat, yaitu: root node rule, branch node rule, leaf node rule, dan conclusion rule. Berikut merupakan rules yang dipakai dalam program CLIPS yang disusun:

Tabel 2.2.1. Deskripsi dari Rules dalam Program CLIPS Predict Breast Cancer

Rules	Deskripsi
mean-concave-points	mean-concave-points digunakan untuk menerima masukan nilai mean-concave-points dari pengguna. Rule ini berjenis root node yang merupakan permulaan dari program.  Aksi:

	<ul style="list-style-type: none"> <li>- printout t "mean-concave-points? "</li> <li>- assert (mean-concave-points (read))</li> </ul>
conclusion	<p>conclusion digunakan untuk mencetak prediksi yang didapatkan</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- prediction ?value</li> <li>- result ?value ?outcome</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- printout t "Hasil Prediksi = "?outcome"Kanker Payudara" crlf</li> </ul>
worst-radius	<p>worst-radius digunakan untuk menerima input worst radius dari pengguna</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- mean-concave-points ?value</li> <li>- test (&lt;= ?value 0.05)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- printout t "worst-radius? "</li> <li>- assert (worst-radius (read))</li> </ul>
worst-perimeter	<p>worst-perimeter digunakan untuk menerima input worst perimeter dari pengguna</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- mean-concave-points ?value</li> <li>- test (&gt; ?value 0.05)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- printout t "worst-perimeter? "</li> <li>- assert (worst-perimeter (read))</li> </ul>
radius-error	<p>radius-error digunakan untuk menerima input radius error dari pengguna</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- worst-radius ?value</li> <li>- test (&lt;= ?value 16.83)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- printout t "radius-error? "</li> <li>- assert (radius-error (read))</li> </ul>

mean-texture	<p>mean-texture digunakan untuk menerima input mean texture dari pengguna</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- worst-radius ?value</li> <li>- (test (&gt; ?value 16.83))</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- printout t "mean-texture? "</li> <li>- assert (mean-texture (read))</li> </ul>
predict-worst-perimeter	<p>predict-worst-perimeter digunakan untuk mendapatkan hasil prediksi jika worst-perimeter &gt; 114.45</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- worst-perimeter ?value</li> <li>- test (&gt; ?value 114.45)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- assert (prediction 0)</li> </ul>
worst-texture-right	<p>worst-texture-right digunakan untuk menerima input worst texture dari pengguna. Rule ini akan dijalankan jika program memasuki bagian kanan dari tree dan worst-perimeter &lt;= 114.45.</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- worst-perimeter ?value</li> <li>- test (&lt;= ?value 114.45)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- printout t "worst-texture? "</li> <li>- assert (worst-texture-right (read))</li> </ul>
predict-mean-texture	<p>predict-mean-texture digunakan untuk mendapatkan hasil prediksi jika mean-texture &lt;= 16.19</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- mean-texture ?value</li> <li>- test (&lt;= ?value 16.19)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- assert (prediction 1)</li> </ul>
perimeter-error	<p>perimeter-error digunakan untuk menerima input perimeter error dari pengguna.</p> <p>Pre Kondisi:</p>

	<ul style="list-style-type: none"> <li>- worst-texture-right ?value</li> <li>- test (&gt; ?value 25.65)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- printout t "perimeter-error? "</li> <li>- assert (perimeter-error (read))</li> </ul>
predict-perimeter-error	<p>predict-perimeter-error digunakan untuk mendapatkan hasil prediksi jika perimeter-error &gt; 1.56</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- perimeter-error ?value</li> <li>- test (&gt; ?value 1.56)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- assert (prediction 0)</li> </ul>
concave-points-error	<p>concave-points-error digunakan untuk menerima input concave points error dari pengguna.</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- mean-texture ?value</li> <li>- test (&gt; ?value 16.19)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- printout t "concave-points-error? "</li> <li>- assert (concave-points-error (read))</li> </ul>
worst-concave-points	<p>worst-concave-points digunakan untuk menerima input worst concave points dari pengguna.</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- worst-texture-right ?value</li> <li>- test (&lt;= ?value 25.65)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- printout t "worst-concave-points? "</li> <li>- assert (worst-concave-points (read))</li> </ul>
mean-radius-right	<p>mean-radius-right digunakan untuk menerima input mean radius dari pengguna. Rule ini akan dijalankan jika program memasuki bagian kanan dari tree dan perimeter-error &lt;= 1.56.</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- perimeter-error ?value</li> <li>- test (&lt;= ?value 1.56)</li> </ul>

	<p>Aksi:</p> <ul style="list-style-type: none"> <li>- printout t "mean-radius? "</li> <li>- assert (mean-radius-right (read))</li> </ul>
yes-concave-points-error	<p>yes-concave-points-error digunakan untuk mendapatkan hasil prediksi dari program</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- concave-points-error ?value</li> <li>- test (&gt; ?value 0.01)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- assert (prediction 1)</li> </ul>
no-concave-points-error	<p>no-concave-points-error digunakan untuk mendapatkan hasil prediksi dari program</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- concave-points-error ?value</li> <li>- test (&lt;= ?value 0.01)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- assert (prediction 0)</li> </ul>
yes-worst-concave-points	<p>yes-worst-concave-points digunakan untuk mendapatkan hasil prediksi dari program</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- worst-concave-points ?value</li> <li>- test (&lt;= ?value 0.17)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- assert (prediction 1)</li> </ul>
no-worst-concave-points	<p>no-worst-concave-points digunakan untuk mendapatkan hasil prediksi dari program</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- worst-concave-points ?value</li> <li>- test (&gt; ?value 0.17)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- assert (prediction 0)</li> </ul>
yes-mean-radius-right	<p>yes-mean-radius-right digunakan untuk mendapatkan hasil prediksi dari program</p> <p>Pre Kondisi:</p>



	<ul style="list-style-type: none"> <li>- mean-radius-right ?value</li> <li>- test (&gt; ?value 13.34)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- assert (prediction 1)</li> <li>-</li> </ul>
no-mean-radius-right	<p>no-mean-radius-right digunakan untuk mendapatkan hasil prediksi dari program</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- mean-radius-right ?value</li> <li>- test (&lt;= ?value 13.34)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- assert (prediction 0)</li> </ul>
worst-texture-left	<p>worst-texture-left digunakan untuk menerima input worst texture dari pengguna.</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- radius-error ?value</li> <li>- test (&lt;= ?value 0.63)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- (printout t "worst-texture? ")</li> <li>- assert (worst-texture-left (read)))</li> </ul>
mean-smoothness	<p>mean-smoothness digunakan untuk menerima input mean smoothness dari pengguna.</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- radius-error ?value</li> <li>- test (&gt; ?value 0.63)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- (printout t "mean-smoothness? ")</li> <li>- assert (mean-smoothness (read)))</li> </ul>
predict-worst-texture	<p>predict-worst-texture digunakan untuk mendapatkan hasil prediksi jika worst-texture-left &lt;= 30.15</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- worst-texture-left ?value</li> <li>- test (&lt;= ?value 30.15)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- assert (prediction 1))</li> </ul>
worst-area	<p>worst-area digunakan untuk menerima input worst area dari pengguna.</p>

	<p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- worst-texture-left ?value</li> <li>- test (&gt; ?value 30.15)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- (printout t "worst-area? ")</li> <li>- assert (worst-area (read)))</li> </ul>
yes-mean-smoothness	<p>yes-mean-smoothness digunakan untuk mendapatkan hasil prediksi jika mean-smoothness &lt;= 0.09</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- mean-smoothness ?value</li> <li>- test (&lt;= ?value 0.09)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- assert (prediction 1))</li> </ul>
no-mean-smoothness	<p>no-mean-smoothness digunakan untuk mendapatkan hasil prediksi jika mean-smoothness &gt; 0.09</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- mean-smoothness ?value</li> <li>- test (&gt; ?value 0.09)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- assert (prediction 0))</li> </ul>
predict-worst-area	<p>predict-worst-area digunakan untuk mendapatkan hasil prediksi jika worst-area &lt;= 641.60</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- worst-area ?value</li> <li>- test (&lt;= ?value 641.60)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- assert (prediction 1))</li> </ul>
mean-radius-left	<p>mean-radius-left digunakan untuk menerima input mean radius dari pengguna.</p> <p>Pre Kondisi:</p> <ul style="list-style-type: none"> <li>- worst-area ?value</li> <li>- test (&gt; ?value 641.60)</li> </ul> <p>Aksi:</p> <ul style="list-style-type: none"> <li>- (printout t "mean-radius? ")</li> <li>- assert (mean-radius-left (read)))</li> </ul>
predict-mean-radius-left	<p>predict-mean-radius-left digunakan untuk mendapatkan hasil prediksi jika mean-radius-left &gt; 13.45</p>

	Pre Kondisi: <ul style="list-style-type: none"> <li>- mean-radius-left ?value</li> <li>- test (&gt; ?value 13.45)</li> </ul> Aksi: <ul style="list-style-type: none"> <li>- assert (prediction 1))</li> </ul>
mean-texture-bottom	mean-texture-bottom digunakan untuk menerima input mean texture dari pengguna.  Pre Kondisi: <ul style="list-style-type: none"> <li>- mean-radius-left ?value</li> <li>- test (&lt;= ?value 13.45)</li> </ul> Aksi: <ul style="list-style-type: none"> <li>- (printout t "mean-texture? ")</li> <li>- assert (mean-texture-bottom (read)))</li> </ul>
yes-mean-texture-bottom	yes-mean-texture-bottom digunakan untuk mendapatkan hasil prediksi jika mean-texture-bottom > 28.79  Pre Kondisi: <ul style="list-style-type: none"> <li>- mean-texture-bottom ?value</li> <li>- test (&gt; ?value 28.79)</li> </ul> Aksi: <ul style="list-style-type: none"> <li>- assert (prediction 1))</li> </ul>
no-mean-texture-bottom	no-mean-texture-bottom digunakan untuk mendapatkan hasil prediksi jika mean-texture-bottom <= 28.79  Pre Kondisi: <ul style="list-style-type: none"> <li>- mean-texture-bottom ?value</li> <li>- test (&lt;= ?value 28.79)</li> </ul> Aksi: <ul style="list-style-type: none"> <li>- assert (prediction 0))</li> </ul>

### 3. Tabel Kontribusi

NIM	Kontribusi
13519084	<ul style="list-style-type: none"><li>- Source code : 5 branch node dan 7 leaf node</li><li>- Dokumen : Bab 1 Facts</li></ul>
13519159	<ul style="list-style-type: none"><li>- Source code : deffacts, root node, 4 branch node, 3 leaf node, dan conclusion rule</li><li>- Dokumen : Bab 1 Algoritma Program</li></ul>
13519176	<ul style="list-style-type: none"><li>- Source code: 5 branch node dan 6 leaf node</li><li>- Dokumen: Bab 2 Rules</li></ul>

### 4. Lampiran

<https://github.com/benidictusgalihmp/CLIPS-predict-breast-cancer>