

**Tugas Besar**  
**IF3170 Intelegensi Buatan**  
**Implementasi CLIPS**

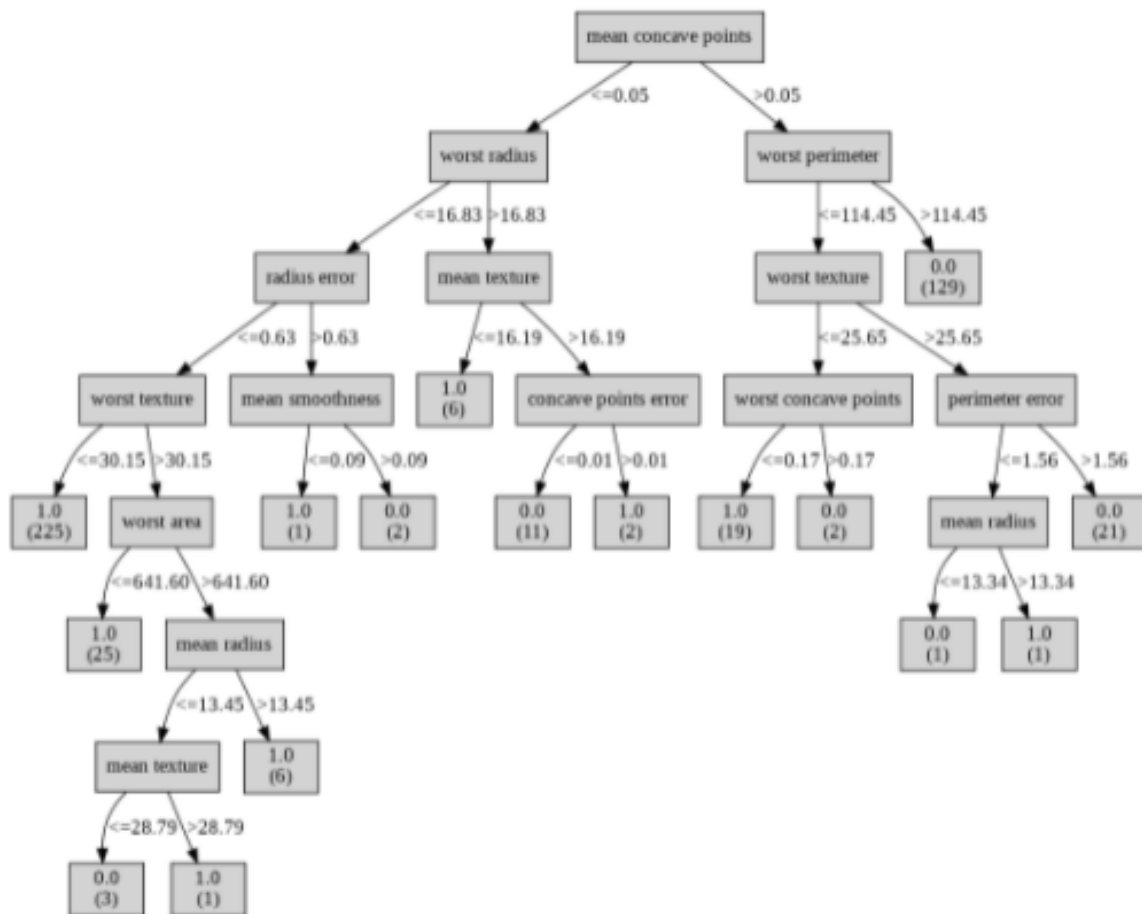


**Disiapkan oleh:**  
**Kelompok Mango**  
Fadel Ananda Dotty - 13519146  
Christian Gunawan - 13519199  
Tanur Rizaldi Rahardjo - 13519214

**Teknik Informatika**  
**Sekolah Teknik Elektro dan Informatika**  
**Institut Teknologi Bandung**  
**2021**

## Penjelasan Program

Pada tugas kali ini, diminta untuk membuat program sederhana dengan menggunakan bahasa CLIPS untuk merepresentasikan decision tree yang dibentuk menggunakan dataset breast cancer ke dalam bahasa CLIPS. Adapun hasil yang ingin didapatkan sebagai berikut.



## Implementasi Program

Berikut hasil program CLIPS dari kelompok kami

```
(defrule tanya-mean-concave-points-eval
=>
  (printout t
    "mean_concave points? "
  )
  (assert(mean-concave-points (read)))
)
```

Program akan dimulai dengan rule *tanya-mean-concave-points-eval* untuk bertanya *mean concave points*, hasil yang didapat rule adalah mencetak *mean\_concave points* untuk menerima *input* dan nilai akan disimpan pada variabel *mean-concave-points*.

```
(defrule mean-concave-points-eval
  (mean-concave-points ?x)
=>
  (if (<= ?x 0.05)
    then
      (assert (worst-radius))
      (assert (branch-kiri))
    else
      (assert (worst-perimeter))
      (assert (branch-kanan))
  )
)
```

Rule *mean-concave-points-eval* akan berjalan jika memiliki nilai *mean-concave-points*. Hasil yang dijalankan oleh rule jika hasil input angka/*integer* yang didapatkan kurang dari sama dengan 0.05, maka akan mendapatkan variabel *worst-radius* dan *branch-kiri*, Jika nilai lebih dari 0.05, maka akan mendapatkan variabel *worst-perimeter* dan *branch-kanan*. Fakta *branch* digunakan untuk membedakan cara evaluasi yang akan dilakukan untuk peraturan evaluasi yang menggunakan nama fakta sama.

```
(defrule tanya-worst-radius
  (worst-radius)
=>
  (printout t "worst_radius? ")
  (assert(worst-radius (read)))
)
```

Hasil yang akan dijalankan adalah mencetak *worst\_radius* untuk menerima input dan nilai akan disimpan pada variabel *worst\_radius*

```
(defrule worst-radius-eval
  (worst-radius ?x)
=>
  (if (<= ?x 16.83)
    then
      (assert (radius-error))
      (assert (texture-kiri))
    else
      (assert (mean-texture))
      (assert (texture-kanan))
  )
)
```

Rule *worst-radius-eval* akan berjalan jika memiliki variabel *worst\_radius* yang memiliki nilai. Hasil yang akan dijalankan jika hasil input angka/*integer* yang didapatkan kurang dari sama dengan 16.83, maka akan mendapatkan variabel *radius-error* dan *texture-kiri*, Jika nilai lebih dari 16.83, maka akan mendapatkan variabel *mean-texture* dan *texture-kanan*.

```
(defrule tanya-worst-perimeter
  (worst-perimeter)
=>
  (printout t "worst_perimeter? ")
  (assert(worst-perimeter (read)))
)
```

Rule *tanya-worst-perimeter* akan berjalan jika memiliki variabel *worst-perimeter*. Hasil yang akan dijalankan adalah mencetak *worst\_perimeter* untuk menerima input dan nilai akan disimpan pada variabel *worst\_perimeter*

```
(defrule worst-perimeter-eval
  (worst-perimeter ?x)
=>
  (if (<= ?x 114.45)
    then
      (assert (worst-texture))
    else
      (printout t "Hasil Prediksi = Tidak Terprediksi Kanker Payudara"
      crlf)
      (halt)
  )
)
```

Rule *worst-perimeter-eval* akan berjalan jika memiliki variabel *worst-perimeter* yang memiliki nilai. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 114.45, maka akan mendapatkan variabel *worst-texture*, Jika nilai lebih dari 114.45, maka akan mencetak “Hasil Prediksi = Tidak Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi.

```
(defrule tanya-radius-error
  (radius-error)
=>
  (printout t "radius_error? ")
  (assert(radius-error (read)))
)
```

Rule *tanya-radius-error* akan berjalan jika memiliki variabel *radius-error*. Hasil yang akan dijalankan adalah mencetak *radius\_error* untuk menerima input dan nilai akan disimpan pada variabel *radius\_error*

```
(defrule radius-error-eval
  (radius-error ?x)
=>
  (if (<= ?x 0.63)
    then
      (assert (worst-texture))
```

```
        else
        (assert (mean-smoothness))
    )
)
```

Rule *radius-error-eval* akan berjalan jika memiliki variabel *radius-error* yang memiliki nilai. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 0.63, maka akan mendapatkan variabel *worst-texture*, Jika nilai lebih dari 0.63, maka akan mendapatkan variabel *mean-smoothness*.

```
(defrule tanya-mean-texture
  (mean-texture)
=>
  (printout t "mean_texture? ")
  (assert(mean-texture (read)))
)
```

Rule *tanya-mean-texture* akan berjalan jika memiliki variabel *mean-texture*. Hasil yang akan dijalankan adalah mencetak *mean\_texture* untuk menerima input dan nilai akan disimpan pada variabel *mean\_texture*

```
(defrule mean-texture-kanan-eval
  (mean-texture ?x)
  (texture-kanan)
=>
  (if (<= ?x 16.19)
    then
      (printout t "Hasil Prediksi = Terprediksi Kanker Payudara" crlf)
      (halt)
    else
      (assert (concave-point-error))
  )
)
```

Rule *mean-texture-kanan-eval* akan berjalan jika memiliki variabel *mean-texture* yang memiliki nilai dan variabel *texture-kanan*. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 16.19, maka akan mencetak hasil “Hasil Prediksi = Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi, Jika nilai lebih dari 16.19, maka akan mendapatkan variabel *concave-point-error*.

```
(defrule tanya-worst-texture
  (worst-texture)
=>
  (printout t "worst_texture? ")
  (assert (worst-texture (read)))
)
```

Rule *tanya-worst-texture* akan berjalan jika memiliki variabel *worst-texture*. Hasil yang akan dijalankan adalah mencetak *worst\_texture* untuk menerima input dan nilai akan disimpan pada variabel *worst\_texture*.

```
(defrule worst-texture-kanan-eval
  (worst-texture ?x)
  (branch-kanan)
=>
  (if (> ?x 25.65)
    then
      (assert (perimeter-error))
    else
      (assert (worst-concave-points))
  )
)
```

Rule *worst-texture-kanan-eval* akan berjalan jika memiliki variabel *worst-texture* yang memiliki nilai dan variabel *branch-kanan*. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 25.65, maka akan mendapatkan variabel *perimeter-error*, Jika nilai lebih dari 25.65, maka akan mendapatkan variabel *worst-concave-points*.

```
(defrule tanya-worst-texture
  (worst-texture)
=>
  (printout t "worst_texture? ")
  (assert (worst-texture (read)))
)
```

Rule *tanya-worst-texture* akan berjalan jika memiliki variabel *worst-texture*. Hasil yang akan dijalankan adalah mencetak *worst\_texture* untuk menerima input dan nilai akan disimpan pada variabel *worst\_texture*.

```
(defrule worst-texture-kiri-eval
  (worst-texture ?x)
  (branch-kiri)
=>
```

```

    (if (<= ?x 30.15)
        then
        (printout t "Hasil Prediksi = Terprediksi Kanker Payudara" crlf)
        (halt)
        else
        (assert (worst-area))
    )
)

```

Rule *worst-texture-kiri-eval* akan berjalan jika memiliki variabel *worst-texture* yang memiliki nilai dan variabel *branch-kanan*. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 30.15, maka akan mencetak hasil “Hasil Prediksi = Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi, Jika nilai lebih dari 30.15, maka akan mendapatkan variabel *worst-area*.

```

(defrule tanya-mean-smoothness
  (mean-smoothness)
=>
  (printout t "mean_smoothness? ")
  (assert(mean-smoothness (read)))
)

```

Rule *tanya-mean-smoothness* akan berjalan jika memiliki variabel *mean-smoothness*. Hasil yang akan dijalankan adalah mencetak *mean-smoothness* untuk menerima input dan nilai akan disimpan pada variabel *mean-smoothness*.

```

(defrule mean-smoothness-eval
  (mean-smoothness ?x)
=>
  (if (<= ?x 0.09)
      then
      (printout t "Hasil Prediksi = Terprediksi Kanker Payudara" crlf)
      (halt)
      else
      (printout t "Hasil Prediksi = Tidak Terprediksi Kanker Payudara"
crlf)
      (halt)
  )
)

```

Rule *mean-smoothness-eval* akan berjalan jika memiliki variabel *mean-smoothness* yang memiliki nilai. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 0.09, maka akan mencetak hasil “Hasil Prediksi = Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi, Jika nilai lebih dari 0.09, maka akan mencetak hasil “Hasil Prediksi = Tidak Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi

```
(defrule tanya-concave-point-error
  (concave-point-error)
=>
  (printout t "concave_point_error? ")
  (assert(concave-point-error (read)))
)
```

Rule *tanya-concave-point-error* akan berjalan jika memiliki variabel *concave-point-error*. Hasil yang akan dijalankan adalah mencetak *concave\_point\_error* untuk menerima input dan nilai akan disimpan pada variabel *concave\_point\_error*.

```
(defrule concave-point-error-eval
  (concave-point-error ?x)
=>
  (if (<= ?x 0.01)
    then
      (printout t "Hasil Prediksi = Tidak Terprediksi Kanker Payudara"
        crlf)
      (halt)
    else
      (printout t "Hasil Prediksi = Terprediksi Kanker Payudara" crlf)
      (halt)
  )
)
```

Rule *concave-point-error-eval* akan berjalan jika memiliki variabel *concave-point-error* yang memiliki nilai. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 0.01, maka akan mencetak hasil “Hasil Prediksi = Tidak Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi, Jika nilai lebih dari 0.01, maka akan mencetak hasil “Hasil Prediksi = Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi

```
(defrule tanya-worst-concave-points
  (worst-concave-points)
=>
  (printout t "worst_concave_points? ")
  (assert(worst-concave-points (read)))
)
```

Rule *tanya-worst-concave-points* akan berjalan jika memiliki variabel *worst-concave-points*. Hasil yang akan dijalankan adalah mencetak *worst\_concave\_points* untuk menerima input dan nilai akan disimpan pada variabel *worst\_concave\_points*.

```
(defrule worst-concave-points-eval
```



```

(worst-concave-points ?x)
=>
  (if (> ?x 0.17)
      then
        (printout t "Hasil Prediksi = Tidak Terprediksi Kanker Payudara"
          crlf)
        (halt)
      else
        (printout t "Hasil Prediksi = Terprediksi Kanker Payudara" crlf)
        (halt)
  )
)

```

Rule *worst-concave-points-eval* akan berjalan jika memiliki variabel *concave-point-error* yang memiliki nilai. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 0.17, maka akan mencetak hasil “Hasil Prediksi = Tidak Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi, Jika nilai lebih dari 0.17, maka akan mencetak hasil “Hasil Prediksi = Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi

```

(defrule tanya-perimeter-error
  (perimeter-error)
=>
  (printout t "perimeter_error? ")
  (assert(perimeter-error (read)))
)

```

Rule *tanya-perimeter-error* akan berjalan jika memiliki variabel *perimeter-error*. Hasil yang akan dijalankan adalah mencetak *perimeter-error* untuk menerima input dan nilai akan disimpan pada variabel *perimeter-error*.

```

(defrule perimeter-error-eval
  (perimeter-error ?x)
=>
  (if (> ?x 1.56)
      then
        (printout t "Hasil Prediksi = Tidak Terprediksi Kanker Payudara"
          crlf)
        (halt)
      else
        (assert (mean-radius))
  )
)

```

Rule *perimeter-error-eval* akan berjalan jika memiliki variabel *perimeter-error* yang memiliki nilai. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 1.56, maka akan mencetak hasil “Hasil Prediksi = Tidak Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi, Jika nilai lebih dari 1.56, maka akan mendapatkan variabel *mean-radius*

```
(defrule tanya-worst-area
  (worst-area)
=>
  (printout t "worst_area? ")
  (assert(worst-area (read)))
)
```

Rule *tanya-worst-area* akan berjalan jika memiliki variabel *worst-area*. Hasil yang akan dijalankan adalah mencetak *worst-area* untuk menerima input dan nilai akan disimpan pada variabel *worst-area*.

```
(defrule worst-area-eval
  (worst-area ?x)
=>
  (if (<= ?x 641.60)
    then
      (printout t "Hasil Prediksi = Terprediksi Kanker Payudara" crlf)
      (halt)
    else
      (assert (mean-radius))
  )
)
```

Rule *worst-area-eval* akan berjalan jika memiliki variabel *worst-area* yang memiliki nilai. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 641.60, maka akan mencetak hasil “Hasil Prediksi = Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi, Jika nilai lebih dari 641.60, maka akan mendapatkan variabel *mean-radius*.

```
(defrule tanya-mean-radius
  (mean-radius)
=>
  (printout t "mean_radius? ")
  (assert (mean-radius (read)))
)
```

Rule *tanya-mean-radius* akan berjalan jika memiliki variabel *mean-radius*. Hasil yang akan dijalankan adalah mencetak *mean-radius* untuk menerima input dan nilai akan disimpan pada variabel *mean-radius*.

```
(defrule mean-radius-kanan-eval
  (branch-kanan)
  (mean-radius ?x)
=>
  (if (> ?x 13.34)
```

```

        then
        (printout t "Hasil Prediksi = Terprediksi Kanker Payudara" crlf)
        (halt)
      else
        (printout t "Hasil Prediksi = Tidak Terprediksi Kanker Payudara"
crlf)
        (halt)
    )
)

```

Rule *mean-radius-kanan-eval* akan berjalan jika memiliki variabel *mean-radius* yang memiliki nilai dan variabel *branch-kanan*. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 13.34, maka akan mencetak hasil “Hasil Prediksi = Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi, Jika nilai lebih dari 13.34, maka akan mencetak hasil “Hasil Prediksi = Tidak Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi

```

(defrule tanya-mean-radius
  (mean-radius)
=>
  (printout t "mean_radius? ")
  (assert (mean-radius (read)))
)

```

Rule *tanya-mean-radius* akan berjalan jika memiliki variabel *mean-radius*. Hasil yang akan dijalankan adalah mencetak *mean-radius* untuk menerima input dan nilai akan disimpan pada variabel *mean-radius*.

```

(defrule mean-radius-kanan-eval
  (branch-kanan)
  (mean-radius ?x)
=>
  (if (> ?x 13.34)
    then
      (printout t "Hasil Prediksi = Terprediksi Kanker Payudara" crlf)
      (halt)
    else
      (printout t "Hasil Prediksi = Tidak Terprediksi Kanker Payudara"
crlf)
      (halt)
  )
)

```

Rule *mean-radius-kanan-eval* akan berjalan jika memiliki variabel *mean-radius* yang memiliki nilai dan variabel *branch-kanan*. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 13.34, maka akan mencetak hasil “Hasil Prediksi = Terprediksi Kanker

Payudara” dan program akan selesai mengeksekusi, Jika nilai lebih dari 13.34, maka akan mencetak hasil “Hasil Prediksi = Tidak Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi

```
(defrule mean-radius-kiri-eval
  (mean-radius ?x)
  (branch-kiri)
=>
  (if (<= ?x 13.45)
    then
      (assert (mean-texture))
    else
      (printout t "Hasil Prediksi = Terprediksi Kanker Payudara" crlf)
      (halt)
  )
)
```

Rule *mean-radius-kiri-eval* akan berjalan jika memiliki variabel *mean-radius* yang memiliki nilai dan variabel *branch-kiri*. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 13.45, maka akan mendapatkan variabel *mean-texture*, Jika nilai lebih dari 13.45, maka akan mencetak hasil “Hasil Prediksi = Tidak Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi.

```
(defrule mean-radius-kanan-eval
  (branch-kanan)
  (mean-radius ?x)
=>
  (if (> ?x 13.34)
    then
      (printout t "Hasil Prediksi = Terprediksi Kanker Payudara" crlf)
      (halt)
    else
      (printout t "Hasil Prediksi = Tidak Terprediksi Kanker Payudara"
crlf)
      (halt)
  )
)
```

Rule *mean-radius-kanan-eval* akan berjalan jika memiliki variabel *mean-radius* yang memiliki nilai dan variabel *branch-kanan*. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 13.34, maka akan mencetak hasil “Hasil Prediksi = Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi, Jika nilai lebih dari 13.34, maka akan mencetak hasil “Hasil Prediksi = Tidak Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi

```
(defrule mean-radius-kiri-eval
  (mean-radius ?x)
```

```

    (branch-kiri)
=>
    (if (<= ?x 13.45)
        then
            (assert (mean-texture))
        else
            (printout t "Hasil Prediksi = Terprediksi Kanker Payudara" crlf)
            (halt)
    )
)

```

Rule *mean-radius-kiri-eval* akan berjalan jika memiliki variabel *mean-radius* yang memiliki nilai dan variabel *branch-kiri*. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 13.45, maka akan mendapatkan variabel *mean-texture*. Jika nilai lebih dari 13.45, maka akan mencetak hasil “Hasil Prediksi = Tidak Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi

```

(defrule mean-texture-kiri-eval
  (mean-texture ?x)
  (texture-kiri)
=>
  (if (<= ?x 28.79)
      then
        (printout t "Hasil Prediksi = Tidak Terprediksi Kanker Payudara"
crlf)
        (halt)
      else
        (printout t "Hasil Prediksi = Terprediksi Kanker Payudara" crlf)
        (halt)
  )
)

```

Rule *mean-texture-kiri-eval* akan berjalan jika memiliki variabel *mean-texture* yang memiliki nilai dan variabel *texture-kiri*. Hasil yang akan dijalankan jika hasil input angka/integer yang didapatkan kurang dari sama dengan 28.79, maka akan mencetak hasil “Hasil Prediksi = Tidak Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi, Jika nilai lebih dari 28.79, maka akan mencetak hasil “Hasil Prediksi = Terprediksi Kanker Payudara” dan program akan selesai mengeksekusi.

Berikut contoh hasil program ketika dijalankan:

```
CLIPS> (reset)
CLIPS> (run)
mean_concave_points? 0.04
worst_radius? 7
radius_error? 17
mean_smoothness? 0.005
Hasil Prediksi = Terprediksi Kanker Payudara
```

Pertama, setelah melakukan (reset) dan (run), program akan menjalankan rule *tanya-mean-concave-points-eval* yang akan mencetak "mean\_concave\_points? " dan jawaban akan dibaca pada *mean-concave-points*. Maka program memiliki variabel *mean-concave-points* yang memiliki nilai. Selanjutnya program akan masuk pada rule *mean-concave-points-eval* karena telah memiliki syarat yaitu variabel *mean-concave-points* yang telah memiliki nilai. Karena masukkan nilai kurang dari 0.05, maka program akan memiliki variabel *worst-radius* dan *branch-kiri*. Program akan berlanjut dengan rule *tanya-worst-radius* karena program memiliki prasyarat yaitu *worst-radius*. Program mencetak "worst\_radius? " dan jawaban akan dibaca pada *worst\_radius*. Program akan masuk pada rule *worst-radius-eval* karena telah memiliki syarat yaitu variabel *worst-radius* yang telah memiliki nilai. Karena masukan nilai kurang dari 7, maka program akan memiliki variabel *radius-error* dan *texture-kiri*. Program akan berlanjut dengan rule *tanya-radius-error* karena program memiliki prasyarat yaitu *radius-error*. Program akan masuk pada rule *radius-error-eval* karena telah memiliki syarat yaitu variabel *radius-error* yang telah memiliki nilai. Karena masukkan nilai lebih dari 17, maka program akan memiliki variabel *mean-smoothness*. Program akan berlanjut dengan rule *tanya-mean-smoothness* karena program memiliki prasyarat yaitu *mean-smoothness*. Program akan masuk pada rule *mean-smoothness-eval* karena telah memiliki syarat yaitu variabel *mean-smoothness* yang telah memiliki nilai. Karena masukan nilai kurang dari 0.09, maka program akan mencetak hasil "Terprediksi Kanker Payudara" dan program akan selesai dieksekusi.

## Pembagian Tugas

13519146	13519199	13519214
Mean Smoothness	Mean Texture Kanan	Mean Concave Points
Perimeter Error	Concave Points Error	Worst Radius
Mean Radius	Worst Perimeter	Radius Error
Worst Area	Worst Texture	Worst Texture
Mean Radius	Worst Concave Point	Mean Texture Kiri
Laporan	Laporan	Laporan