

Tugas Besar 2 IF3170 Intelegensi Buatan
Membuat Agen Minesweeper Berdasarkan *Knowledge Based System*
Semester I Tahun 2020/2021



Anggota Kelompok:

Felicia Gillian Tekad Tuerah	13518070
Radhinansyah Hemsah Ghaida	13518087
Mutiara Arifazzahra	13518139
Fabianus Harry Setiawan	13518144

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2020

Daftar Isi

Daftar Isi	2
BAB I	
Tahapan Pembangunan Aplikasi Minesweeper	3
BAB II	
Repository dan Dokumentasi	25
BAB III	
User Manual	26
BAB IV	
Proses Updating dan Inferencing Fakta	27

BAB I

Tahapan Pembangunan Aplikasi Minesweeper

Aplikasi agen penyelesaian permainan *minesweeper* ini dibangun di atas program CLIPS dan Python. Program CLIPS akan menangani proses agen dalam menjalankan proses-proses dengan *shell* yang dimilikinya dan bertujuan untuk melakukan inferensi/*reasoning*. *Shell* CLIPS inilah yang menyediakan elemen dasar dari suatu *expert system*. Elemen-elemen tersebut terdiri dari *fact-list*, *instance-list*, *rule-list*, dan *object-list*. CLIPS menjalankan proses inferensi dengan menerapkan *rule* yang memenuhi *fact* yang tersedia. *Rule* yang belum dijalankan akan tersimpan pada suatu agenda.

Pada awal pembangunan program, terlebih dahulu dibuat sebuah program Python yaitu `processMatrix.py` dan `main.py`. Program `processMatrix.py` berisi fungsi-fungsi yang akan menghubungkan program pada Python dan CLIPS. Program `main.py` merupakan program utama untuk menjalankan agen *minesweeper*. Program tersebut akan meminta input dari pengguna berupa besar dari papan *minesweeper* yang diinginkan, jumlah bom yang akan di-*generate*, dan koordinat-koordinat setiap bom yang ingin dimasukkan. Setelah itu, masukkan tersebut akan di input ke program CLIPS yang dibuat yaitu `minesweeper.clp` oleh beberapa fungsi pada Python dengan memanfaatkan *library* `clipsy`.

Pada program agen *minesweeper* pada CLIPS yang dibuat, terdapat beberapa elemen sebagai aplikasi agen penyelesaian *minesweeper* terdiri dari:

1. Template

- `deftemplate boxValue`

```
(deftemplate boxValue
  (slot x)
  (slot y)
  (slot value)
  (slot flagcount)
  (slot opened)
)
```

- `deftemplate select`

```
(deftemplate select
  (slot x)
  (slot y)
)
```

- deftemplate opened

```
(deftemplate opened
  (slot x)
  (slot y)
)
```

- deftemplate markMine

```
(deftemplate markMine
  ; jika markMine, auto opened
  (slot x)
  (slot y)
)
```

- deftemplate cekCountFlags

```
(deftemplate cekCountFlags
  (slot xHome)
  (slot yHome)
  (slot xCek)
  (slot yCek)
)
```

- deftemplate cekCountOpened

```
(deftemplate cekCountOpened
  (slot xHome)
  (slot yHome)
  (slot xCek)
  (slot yCek)
)
```

- deftemplate boxValuePython

```
(deftemplate boxValuePython
  (slot x)
  (slot y)
  (slot value)
)
```

- deftemplate cekMarkVal

```
(deftemplate cekMarkVal
  (slot x)
  (slot y)
)
```

- deftemplate banyakMinesPython

```
(deftemplate banyakMinesPython
  (slot n)
)
```

- deftemplate sizeBoardPython

```
(deftemplate sizeBoardPython
  (slot n)
)
```

- deftemplate minesKoordpython

```
(deftemplate minesKoordpython
  (slot x)
  (slot y)
)
```

- deftemplate countMine

```
(deftemplate countMine
  (slot n)
)
```

- deftemplate gameOver

```
(deftemplate gameOver
  (slot bool)
)
```

2. Rules

- defrule initialize

```
(defrule initialize
  (initial-fact)
  (banyakMinesPython(n ?n))
  =>
  (assert (select (x 0) (y 0)))
  (assert (countMine (n ?n)))
  (assert (gameOver (bool False))))
```

- defrule countOpened1

```
(defrule countOpened1
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?flagcount) (opened ?opened))
  (opened (x ?a) (y ?b))

  ?f1 <- (cekCountOpened(xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
  (test(= ?a (+ ?x 1)))
  (test(= ?b (+ ?y 1)))
  =>
  (modify ?f (opened (+ ?opened 1)))
  (retract ?f1)
)
```

- defrule countOpened2

```
(defrule countOpened2
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?flagcount) (opened ?opened))
  (opened (x ?a) (y ?b))

  ?f1 <- (cekCountOpened(xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
  (test(= ?a (+ ?x 1)))
  (test(= ?b ?y))
  =>
  (modify ?f (opened (+ ?opened 1)))
  (retract ?f1)
)
```

- defrule countOpened3

```
(defrule countOpened3
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?flagcount) (opened ?opened))
  (opened (x ?a) (y ?b))

  ?f1 <- (cekCountOpened(xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
  (test(= ?a (+ ?x 1)))
  (test(= ?b (- ?y 1)))
  =>
  (modify ?f (opened (+ ?opened 1)))
  (retract ?f1)
)
```

- defrule countOpened4

```
(defrule countOpened4
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?flagcount) (opened ?opened))
  (opened (x ?a) (y ?b))

  ?f1 <- (cekCountOpened(xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
  (test(= ?a ?x))
  (test(= ?b (+ ?y 1)))
  =>
  (modify ?f (opened (+ ?opened 1)))
  (retract ?f1)
)
```

- defrule countOpened5

```
(defrule countOpened5
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?flagcount) (opened ?opened))
  (opened (x ?a) (y ?b))

  ?f1 <- (cekCountOpened(xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
  (test(= ?a ?x))
  (test(= ?b (- ?y 1)))
  =>
```

```

      (modify ?f (opened (+ ?opened 1)))
      (retract ?f1)
    )

```

- defrule countOpened6

```

(defrule countOpened6
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?flagcount) (opened ?opened))
  (opened (x ?a) (y ?b))

  ?f1 <- (cekCountOpened(xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
  (test(= ?a (- ?x 1)))
  (test(= ?b (+ ?y 1)))
  =>
  (modify ?f (opened (+ ?opened 1)))
  (retract ?f1)
)

```

- defrule countOpened7

```

(defrule countOpened7
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?flagcount) (opened ?opened))
  (opened (x ?a) (y ?b))

  ?f1 <- (cekCountOpened(xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
  (test(= ?a (- ?x 1)))
  (test(= ?b ?y))
  =>
  (modify ?f (opened (+ ?opened 1)))
  (retract ?f1)
)

```

- defrule countOpened8

```

(defrule countOpened8
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?flagcount) (opened ?opened))
  (opened (x ?a) (y ?b))

```



```

    ?f1 <- (cekCountOpened(xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
    (test(= ?a (- ?x 1)))
    (test(= ?b (- ?y 1)))
    =>
    (modify ?f (opened (+ ?opened 1)))
    (retract ?f1)
)

```

- defrule countflag1

```

(defrule countflag1
  (declare (salience 10))
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?count) (opened ?opened))
  (markMine(x ?a) (y ?b))

  ?f1 <- (cekCountFlags (xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
  (test(= ?a (+ ?x 1)))
  (test(= ?b (+ ?y 1)))
  =>
  (modify ?f (flagcount (+ ?count 1)))
  (retract ?f1)
)

```

- defrule countflag2

```

(defrule countflag2
  (declare (salience 10))
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?count) (opened ?opened))
  (markMine(x ?a) (y ?b))

  ?f1 <- (cekCountFlags (xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
  (test(= ?a (+ ?x 1)))
  (test(= ?b ?y))
  =>
  (modify ?f (flagcount (+ ?count 1)))
  (retract ?f1)
)

```

- defrule countflag3

```
(defrule countflag3
  (declare (salience 10))
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?count) (opened ?opened))
  (markMine(x ?a) (y ?b))

  ?f1 <- (cekCountFlags (xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
  (test(= ?a (+ ?x 1)))
  (test(= ?b (- ?y 1)))
  =>
  (modify ?f (flagcount (+ ?count 1)))
  (retract ?f1)
)
```

- defrule countflag4

```
(defrule countflag4
  (declare (salience 10))
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?count) (opened ?opened))
  (markMine(x ?a) (y ?b))

  ?f1 <- (cekCountFlags (xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
  (test(= ?a ?x))
  (test(= ?b (+ ?y 1)))
  =>
  (modify ?f (flagcount (+ ?count 1)))
  (retract ?f1)
)
```

- defrule countflag5

```
(defrule countflag5
  (declare (salience 10))
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?count) (opened ?opened))
  (markMine(x ?a) (y ?b))

  ?f1 <- (cekCountFlags (xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
```

```

    (test(= ?a ?x))
    (test(= ?b (- ?y 1)))
    =>
    (modify ?f (flagcount (+ ?count 1)))
    (retract ?f1)
  )

```

- defrule countflag6

```

(defrule countflag6
  (declare (salience 10))
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?count) (opened ?opened))
  (markMine(x ?a) (y ?b))

  ?f1 <- (cekCountFlags (xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
  (test(= ?a (- ?x 1)))
  (test(= ?b (- ?y 1)))
  =>
  (modify ?f (flagcount (+ ?count 1)))
  (retract ?f1)
)

```

- defrule countflag7

```

(defrule countflag7
  (declare (salience 10))
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?count) (opened ?opened))
  (markMine(x ?a) (y ?b))

  ?f1 <- (cekCountFlags (xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
  (test(= ?a (- ?x 1)))
  (test(= ?b ?y))
  =>
  (modify ?f (flagcount (+ ?count 1)))
  (retract ?f1)
)

```

- defrule countflag8

```
(defrule countflag8
  (declare (salience 10))
  ?f <- (boxValue (x ?x) (y ?y) (value ?value) (flagcount
?count) (opened ?opened))
  (markMine(x ?a) (y ?b))

  ?f1 <- (cekCountFlags (xHome ?x) (yHome ?y) (xCek ?a) (yCek
?b))
  (test(= ?a (- ?x 1)))
  (test(= ?b (+ ?y 1)))
  =>
  (modify ?f (flagcount (+ ?count 1)))
  (retract ?f1)
)
```

- defrule markval

```
(defrule markval
  (sizeBoardPython(n ?size))
  (boxValue (x ?x) (y ?y) (value ?val) (flagcount ?flag)
(opened ?opened))
  (test(= ?val (+ ?flag (- 8 ?opened))))
  ; modify fakta countMine
  ; ?f1 <- (countMine(n ?nilai))

  =>
  ; ditandain pake facts
  (if (and (= ?x 0) (= ?y 0)) then ;Kiri atas
    (assert (cekMarkVal (x (+ ?x 1)) (y (+ ?y 1))))
    (assert (cekMarkVal (x (+ ?x 1)) (y ?y)))
    (assert (cekMarkVal (x ?x) (y (+ ?y 1))))
  else ( if (and (= ?x 0) (= ?y ?size)) then ; Kanan atas
    (assert (cekMarkVal (x (+ ?x 1)) (y ?y)))
    (assert (cekMarkVal (x (+ ?x 1)) (y (- ?y 1))))
    (assert (cekMarkVal (x ?x) (y (- ?y 1))))
  else( if (and (= ?x ?size) (= ?y 0)) then; Kiri bawah
    (assert (cekMarkVal (x (- ?x 1)) (y ?y)))
    (assert (cekMarkVal (x (- ?x 1)) (y (+ ?y 1))))
    (assert (cekMarkVal (x ?x) (y (+ ?y 1))))
  else ( if (and (= ?x ?size) (= ?y ?size)) then ;
Kanan Bawah
    (assert (cekMarkVal (x (- ?x 1)) (y ?y)))
```

```

1))))
    (assert (cekMarkVal (x (- ?x 1)) (y (- ?y
1))))
    (assert (cekMarkVal (x ?x) (y (- ?y 1))))
else ( if (= ?x 0) then ; Baris Atas
    (assert (cekMarkVal (x (+ ?x 1)) (y (+ ?y
1))))
    (assert (cekMarkVal (x (+ ?x 1)) (y ?y)))
    (assert (cekMarkVal (x (+ ?x 1)) (y (- ?y
1))))
    (assert (cekMarkVal (x ?x) (y (- ?y 1))))
    (assert (cekMarkVal (x ?x) (y (+ ?y 1))))
else( if (= ?y 0) then ; Kolom kiri
    (assert (cekMarkVal (x (+ ?x 1)) (y
(+ ?y 1))))
    (assert (cekMarkVal (x (+ ?x 1)) (y
?y)))
    (assert (cekMarkVal (x ?x) (y (+ ?y
1))))
    (assert (cekMarkVal (x (- ?x 1)) (y
?y)))
    (assert (cekMarkVal (x (- ?x 1)) (y
(+ ?y 1))))
    else( if (= ?x ?size) then ;Baris
bawah
        (assert (cekMarkVal (x ?x) (y (-
?y 1))))
        (assert (cekMarkVal (x ?x) (y (+
?y 1))))
        (assert (cekMarkVal (x (- ?x 1))
(y (- ?y 1))))
        (assert (cekMarkVal (x (- ?x 1))
(y ?y)))
        (assert (cekMarkVal (x (- ?x 1))
(y (+ ?y 1))))
        else( if (= ?y ?size) then ;
Kolom kanan
            (assert (cekMarkVal (x (+ ?x
1)) (y ?y)))
            (assert (cekMarkVal (x (+ ?x
1)) (y (- ?y 1))))
            (assert (cekMarkVal (x ?x) (y
(- ?y 1))))
            (assert (cekMarkVal (x (- ?x

```

[illegible]

- `defrule markValOppened`

```
(defrule markValOppened
  ?f <- (cekMarkVal (x ?x) (y ?y))
  (opened(x ?x) (y ?y))
=>
  (retract ?f)
)
```

- `defrule markValUnOppened`

```
(defrule markValUnOpened
  ?f <- (cekMarkVal (x ?x) (y ?y))
```

```

(not(opened(x ?x) (y ?y)))
; modify fakta countMine
?f1 <- (countMine(n ?nilai))
=>
(assert(opened(x ?x) (y ?y)))
(assert(markMine(x ?x) (y ?y)))
(printout t "Mine yang di Flag: ( " ?x " , " ?y " )" clrf)
(retract ?f)
(modify ?f1 (n (- ?nilai 1)))
)

```

- defrule oneMineMarked

```

(defrule oneMineMarked
  (sizeBoardPython(n ?size))
  ; ada satu yang sudah di markMine
  (boxValue (x ?x) (y ?y) (value ?val) (flagcount ?f1))
  ; jika flagcount=val , sisanya dibuka
  (test(= ?f1 ?val))
  =>
  (if (and (= ?x 0) (= ?y 0)) then ;Kiri atas
    (assert (select (x (+ ?x 1)) (y (+ ?y 1))))
    (assert (select (x (+ ?x 1)) (y ?y)))
    (assert (select (x ?x) (y (+ ?y 1))))
  else ( if (and (= ?x 0) (= ?y ?size)) then ; Kanan atas
    (assert (select (x (+ ?x 1)) (y ?y)))
    (assert (select (x (+ ?x 1)) (y (- ?y 1))))
    (assert (select (x ?x) (y (- ?y 1))))
  else( if (and (= ?x ?size) (= ?y 0)) then; Kiri bawah
    (assert (select (x (- ?x 1)) (y ?y)))
    (assert (select (x (- ?x 1)) (y (+ ?y 1))))
    (assert (select (x ?x) (y (+ ?y 1))))
  else ( if (and (= ?x ?size) (= ?y ?size)) then ;
Kanan Bawah
    (assert (select (x (- ?x 1)) (y ?y)))
    (assert (select (x (- ?x 1)) (y (- ?y 1))))
    (assert (select (x ?x) (y (- ?y 1))))
  else ( if (= ?x 0) then ; Baris Atas
    (assert (select (x (+ ?x 1)) (y (+ ?y
1))))
    (assert (select (x (+ ?x 1)) (y ?y)))
    (assert (select (x (+ ?x 1)) (y (- ?y
1))))

```

```

(assert (select (x ?x) (y (- ?y 1))))
(assert (select (x ?x) (y (+ ?y 1))))
else( if (= ?y 0) then ; Kolom kiri
(assert (select (x (+ ?x 1)) (y (+ ?y
1))))

(assert (select (x (+ ?x 1)) (y ?y)))
(assert (select (x ?x) (y (+ ?y 1))))
(assert (select (x (- ?x 1)) (y ?y)))
(assert (select (x (- ?x 1)) (y (+ ?y
1))))

else( if (= ?x ?size) then
(assert (select (x ?x) (y (- ?y
1))))

(assert (select (x ?x) (y (+ ?y
1))))

(assert (select (x (- ?x 1)) (y
(- ?y 1))))

(assert (select (x (- ?x 1)) (y
?y)))

(assert (select (x (- ?x 1)) (y
(+ ?y 1))))

else( if (= ?y ?size) then
(assert (select (x (+ ?x 1))
(y ?y)))

(assert (select (x (+ ?x 1))
(y (- ?y 1))))

(assert (select (x ?x) (y (-
?y 1))))

(assert (select (x (- ?x 1))
(y (- ?y 1))))

(assert (select (x (- ?x 1))
(y ?y)))

else
(assert (select (x (+ ?x
1)) (y (+ ?y 1))))

(assert (select (x (+ ?x
1)) (y ?y)))

(assert (select (x (+ ?x
1)) (y (- ?y 1))))

(assert (select (x ?x) (y
(- ?y 1))))

(assert (select (x ?x) (y
(+ ?y 1))))

```



```

1)) (y (- ?y 1)))

1)) (y ?y))

1)) (y (+ ?y 1)))

)

)

)

)

)

)

)

)

```

- `defrule openOpenedBox`

```
(defrule openOpenedBox
  ; buka box yang sudah dibuka
  ?f <- (select (x ?x) (y ?y))
  (opened (x ?x) (y ?y))
=>
  (retract ?f)
)
```

- `defrule openUnOpenedBox`

```
(defrule openUnOpenedBox
  ; buka box yang belum dibuka
  ?f <- (select (x ?x) (y ?y))
  (not(opened (x ?x) (y ?y)))
  ; ngambil value dari python
  (boxValuePython(x ?x) (y ?y) (value ?val))
  (sizeBoardPython(n ?size))

=>
  (assert(opened(x ?x) (y ?y)))
  (assert(boxValue (x ?x) (y ?y) (value ?val) (flagcount 0) (opened
0)))
  ; tambah fakta checkCountFlags
  (if (and (= ?x 0) (= ?y 0)) then ;Kiri atas
      (assert(cekCountFlags (xHome ?x) (yHome ?y) (xCek (+ ?x
```

```

1)) (yCek (+ ?y 1)))
      (assert(cekCountFlags (xHome ?x) (yHome ?y) (xCek (+ ?x
1)) (yCek ?y)))
      (assert(cekCountFlags (xHome ?x) (yHome ?y) (xCek ?x)
(yCek (+ ?y 1))))
      else ( if (and (= ?x 0) (= ?y ?size)) then ; Kanan atas
      (assert(cekCountFlags (xHome ?x) (yHome ?y) (xCek (+
?x 1)) (yCek ?y)))
      (assert(cekCountFlags (xHome ?x) (yHome ?y) (xCek (+
?x 1)) (yCek (- ?y 1))))
      (assert(cekCountFlags (xHome ?x) (yHome ?y) (xCek ?x)
(yCek (- ?y 1))))
      else( if (and (= ?x ?size) (= ?y 0)) then; Kiri bawah
      (assert(cekCountFlags (xHome ?x) (yHome ?y) (xCek
?x) (yCek (+ ?y 1))))
      (assert(cekCountFlags (xHome ?x) (yHome ?y) (xCek
(- ?x 1)) (yCek (+ ?y 1))))
      (assert(cekCountFlags (xHome ?x) (yHome ?y) (xCek
(- ?x 1)) (yCek ?y)))
      else ( if (and (= ?x ?size) (= ?y ?size)) then ;
Kanan Bawah
      (assert(cekCountFlags (xHome ?x) (yHome ?y)
(xCek ?x) (yCek (- ?y 1))))
      (assert(cekCountFlags (xHome ?x) (yHome ?y)
(xCek (- ?x 1)) (yCek ?y)))
      (assert(cekCountFlags (xHome ?x) (yHome ?y)
(xCek (- ?x 1)) (yCek (- ?y 1))))
      else ( if (= ?x 0) then ; Baris Atas
      (assert(cekCountFlags (xHome ?x) (yHome
?y) (xCek (+ ?x 1)) (yCek (+ ?y 1))))
      (assert(cekCountFlags (xHome ?x) (yHome
?y) (xCek (+ ?x 1)) (yCek ?y)))
      (assert(cekCountFlags (xHome ?x) (yHome
?y) (xCek (+ ?x 1)) (yCek (- ?y 1))))
      (assert(cekCountFlags (xHome ?x) (yHome
?y) (xCek ?x) (yCek (+ ?y 1))))
      (assert(cekCountFlags (xHome ?x) (yHome
?y) (xCek ?x) (yCek (- ?y 1))))
      else( if (= ?y 0) then ; Kolom kiri
      (assert(cekCountFlags (xHome ?x)
(yHome ?y) (xCek (+ ?x 1)) (yCek (+ ?y 1))))
      (assert(cekCountFlags (xHome ?x)
(yHome ?y) (xCek (+ ?x 1)) (yCek ?y)))

```

```

                                (assert(cekCountFlags (xHome ?x)
(yHome ?y) (xCek ?x) (yCek (+ ?y 1))))
                                (assert(cekCountFlags (xHome ?x)
(yHome ?y) (xCek (- ?x 1)) (yCek (+ ?y 1))))
                                (assert(cekCountFlags (xHome ?x)
(yHome ?y) (xCek (- ?x 1)) (yCek ?y)))
                                else( if (= ?x ?size) then ; Baris
bawah
                                (assert(cekCountFlags (xHome ?x)
(yHome ?y) (xCek ?x) (yCek (+ ?y 1))))
                                (assert(cekCountFlags (xHome ?x)
(yHome ?y) (xCek ?x) (yCek (- ?y 1))))
                                (assert(cekCountFlags (xHome ?x)
(yHome ?y) (xCek (- ?x 1)) (yCek (+ ?y 1))))
                                (assert(cekCountFlags (xHome ?x)
(yHome ?y) (xCek (- ?x 1)) (yCek ?y)))
                                (assert(cekCountFlags (xHome ?x)
(yHome ?y) (xCek (- ?x 1)) (yCek (- ?y 1))))
                                else( if (= ?y ?size) then ;
Kolom kanan
                                (assert(cekCountFlags (xHome
?x) (yHome ?y) (xCek (+ ?x 1)) (yCek ?y)))
                                (assert(cekCountFlags (xHome
?x) (yHome ?y) (xCek (+ ?x 1)) (yCek (- ?y 1))))
                                (assert(cekCountFlags (xHome
?x) (yHome ?y) (xCek ?x) (yCek (- ?y 1))))
                                (assert(cekCountFlags (xHome
?x) (yHome ?y) (xCek (- ?x 1)) (yCek ?y)))
                                (assert(cekCountFlags (xHome
?x) (yHome ?y) (xCek (- ?x 1)) (yCek (- ?y 1))))
                                else ; Semua berhasil brow
                                (assert(cekCountFlags
(xHome ?x) (yHome ?y) (xCek (+ ?x 1)) (yCek (+ ?y 1))))
                                (assert(cekCountFlags
(xHome ?x) (yHome ?y) (xCek (+ ?x 1)) (yCek ?y)))
                                (assert(cekCountFlags
(xHome ?x) (yHome ?y) (xCek (+ ?x 1)) (yCek (- ?y 1))))
                                (assert(cekCountFlags
(xHome ?x) (yHome ?y) (xCek ?x) (yCek (+ ?y 1))))
                                (assert(cekCountFlags
(xHome ?x) (yHome ?y) (xCek ?x) (yCek (- ?y 1))))
                                (assert(cekCountFlags
(xHome ?x) (yHome ?y) (xCek (- ?x 1)) (yCek (+ ?y 1))))

```

```
(assert(cekCountFlags
(xHome ?x) (yHome ?y) (xCek (- ?x 1)) (yCek ?y)))
(assert(cekCountFlags
(xHome ?x) (yHome ?y) (xCek (- ?x 1)) (yCek (- ?y 1))))
)
)
)
)
)
)
)
)
(if (and (= ?x 0) (= ?y 0)) then ;Kiri atas
(assert(cekCountOpened (xHome ?x) (yHome ?y) (xCek (+ ?x
1)) (yCek (+ ?y 1))))
(assert(cekCountOpened (xHome ?x) (yHome ?y) (xCek (+ ?x
1)) (yCek ?y)))
(assert(cekCountOpened (xHome ?x) (yHome ?y) (xCek ?x)
(yCek (+ ?y 1))))
else ( if (and (= ?x 0) (= ?y ?size)) then ; Kanan atas
(assert(cekCountOpened (xHome ?x) (yHome ?y) (xCek (+
?x 1)) (yCek ?y)))
(assert(cekCountOpened (xHome ?x) (yHome ?y) (xCek (+
?x 1)) (yCek (- ?y 1))))
(assert(cekCountOpened (xHome ?x) (yHome ?y) (xCek
?x) (yCek (- ?y 1))))
else( if (and (= ?x ?size) (= ?y 0)) then; Kiri bawah
(assert(cekCountOpened (xHome ?x) (yHome ?y)
(xCek ?x) (yCek (+ ?y 1))))
(assert(cekCountOpened (xHome ?x) (yHome ?y)
(xCek (- ?x 1)) (yCek (+ ?y 1))))
(assert(cekCountOpened (xHome ?x) (yHome ?y)
(xCek (- ?x 1)) (yCek ?y)))
else ( if (and (= ?x ?size) (= ?y ?size)) then ;
Kanan Bawah
(assert(cekCountOpened (xHome ?x) (yHome ?y)
(xCek ?x) (yCek (- ?y 1))))
(assert(cekCountOpened (xHome ?x) (yHome ?y)
(xCek (- ?x 1)) (yCek ?y)))
(assert(cekCountOpened (xHome ?x) (yHome ?y)
(xCek (- ?x 1)) (yCek (- ?y 1))))
else ( if (= ?x 0) then ; Baris Atas
(assert(cekCountOpened (xHome ?x) (yHome
```

```

?y) (xCek (+ ?x 1)) (yCek (+ ?y 1)))
      (assert(cekCountOpened (xHome ?x) (yHome
?y) (xCek (+ ?x 1)) (yCek ?y)))
      (assert(cekCountOpened (xHome ?x) (yHome
?y) (xCek (+ ?x 1)) (yCek (- ?y 1))))
      (assert(cekCountOpened (xHome ?x) (yHome
?y) (xCek ?x) (yCek (+ ?y 1))))
      (assert(cekCountOpened (xHome ?x) (yHome
?y) (xCek ?x) (yCek (- ?y 1))))
      else( if (= ?y 0) then ; Kolom kiri
      (assert(cekCountOpened (xHome ?x)
(yHome ?y) (xCek (+ ?x 1)) (yCek (+ ?y 1))))
      (assert(cekCountOpened (xHome ?x)
(yHome ?y) (xCek (+ ?x 1)) (yCek ?y)))
      (assert(cekCountOpened (xHome ?x)
(yHome ?y) (xCek ?x) (yCek (+ ?y 1))))
      (assert(cekCountOpened (xHome ?x)
(yHome ?y) (xCek (- ?x 1)) (yCek (+ ?y 1))))
      (assert(cekCountOpened (xHome ?x)
(yHome ?y) (xCek (- ?x 1)) (yCek ?y)))
      else( if (= ?x ?size) then ; Baris
bawah
      (assert(cekCountOpened (xHome ?x)
(yHome ?y) (xCek ?x) (yCek (+ ?y 1))))
      (assert(cekCountOpened (xHome ?x)
(yHome ?y) (xCek ?x) (yCek (- ?y 1))))
      (assert(cekCountOpened (xHome ?x)
(yHome ?y) (xCek (- ?x 1)) (yCek (+ ?y 1))))
      (assert(cekCountOpened (xHome ?x)
(yHome ?y) (xCek (- ?x 1)) (yCek ?y)))
      (assert(cekCountOpened (xHome ?x)
(yHome ?y) (xCek (- ?x 1)) (yCek (- ?y 1))))
      else( if (= ?y ?size) then ;
Kolom kanan
      (assert(cekCountOpened (xHome
?x) (yHome ?y) (xCek (+ ?x 1)) (yCek ?y)))
      (assert(cekCountOpened (xHome
?x) (yHome ?y) (xCek (+ ?x 1)) (yCek (- ?y 1))))
      (assert(cekCountOpened (xHome
?x) (yHome ?y) (xCek ?x) (yCek (- ?y 1))))
      (assert(cekCountOpened (xHome
?x) (yHome ?y) (xCek (- ?x 1)) (yCek ?y)))
      (assert(cekCountOpened (xHome

```

```
(?x) (yHome ?y) (xCek (- ?x 1)) (yCek (- ?y 1)))  
                                else ; Semua berhasil brow  
                                (assert(cekCountOpened  
(xHome ?x) (yHome ?y) (xCek (+ ?x 1)) (yCek (+ ?y 1))))  
                                (assert(cekCountOpened  
(xHome ?x) (yHome ?y) (xCek (+ ?x 1)) (yCek ?y)))  
                                (assert(cekCountOpened  
(xHome ?x) (yHome ?y) (xCek (+ ?x 1)) (yCek (- ?y 1))))  
                                (assert(cekCountOpened  
(xHome ?x) (yHome ?y) (xCek ?x) (yCek (+ ?y 1))))  
                                (assert(cekCountOpened  
(xHome ?x) (yHome ?y) (xCek ?x) (yCek (- ?y 1))))  
                                (assert(cekCountOpened  
(xHome ?x) (yHome ?y) (xCek (- ?x 1)) (yCek (+ ?y 1))))  
                                (assert(cekCountOpened  
(xHome ?x) (yHome ?y) (xCek (- ?x 1)) (yCek ?y)))  
                                (assert(cekCountOpened  
(xHome ?x) (yHome ?y) (xCek (- ?x 1)) (yCek (- ?y 1))))  
                            )  
                    )  
            )  
    )  
  
(retract ?f)  

```

- defrule openBoxZeroValue

```
(defrule openBoxZeroValue
  ; buka box sekitar box dengan value 0
  (opened(x ?x) (y ?y))
  (boxValue(x ?x) (y ?y) (value 0))
  (sizeBoardPython(n ?size))
=>
  (if (and (= ?x 0) (= ?y 0)) then ;Kiri atas
      (assert (select (x (+ ?x 1)) (y (+ ?y 1))))
      (assert (select (x (+ ?x 1)) (y ?y)))
      (assert (select (x ?x) (y (+ ?y 1))))
      else ( if (and (= ?x 0) (= ?y ?size)) then ; Kanan atas
              (assert (select (x (+ ?x 1)) (y ?y)))
```

```

(assert (select (x (+ ?x 1)) (y (- ?y 1))))
(assert (select (x ?x) (y (- ?y 1))))
else( if (and (= ?x ?size) (= ?y 0)) then; Kiri bawah
(assert (select (x (- ?x 1)) (y ?y)))
(assert (select (x (- ?x 1)) (y (+ ?y 1))))
(assert (select (x ?x) (y (+ ?y 1))))
else ( if (and (= ?x ?size) (= ?y ?size)) then ;
Kanan Bawah
(assert (select (x (- ?x 1)) (y ?y)))
(assert (select (x (- ?x 1)) (y (- ?y 1))))
(assert (select (x ?x) (y (- ?y 1))))
else ( if (= ?x 0) then ; Baris Atas
(assert (select (x (+ ?x 1)) (y (+ ?y
1))))
(assert (select (x (+ ?x 1)) (y ?y)))
(assert (select (x (+ ?x 1)) (y (- ?y
1))))
(assert (select (x ?x) (y (- ?y 1))))
(assert (select (x ?x) (y (+ ?y 1))))
else( if (= ?y 0) then ; Kolom kiri
(assert (select (x (+ ?x 1)) (y (+ ?y
1))))
(assert (select (x (+ ?x 1)) (y ?y)))
(assert (select (x ?x) (y (+ ?y 1))))
(assert (select (x (- ?x 1)) (y ?y)))
(assert (select (x (- ?x 1)) (y (+ ?y
1))))
else( if (= ?x ?size) then ; Baris
bawah
(assert (select (x ?x) (y (- ?y
1))))
(assert (select (x ?x) (y (+ ?y
1))))
(assert (select (x (- ?x 1)) (y
(- ?y 1))))
(assert (select (x (- ?x 1)) (y
?y)))
(assert (select (x (- ?x 1)) (y
(+ ?y 1))))
else( if (= ?y ?size) then ;
Kolom kanan
(assert (select (x (+ ?x 1))
(y ?y)))

```


BAB II

Repository dan Dokumentasi

Kumpulan program aplikasi ini dapat diakses pada [repository Github ini](#). Aplikasi agen *minesweeper* ini terdiri atas beberapa file, yakni:

- `main.py`

File ini berperan sebagai bagian penghubung antara *user* dengan agen. *File* ini menerima masukan dari *user* berupa ukuran *board*, banyaknya *mine*, dan koordinat lokasi *mine*. Setelah masukan diterima, program memuat data dari agen CLIPS dan menambahkan *facts* agar agen dapat melakukan inferensi. *File* juga akan menjalankan (*run*) agen CLIPS dan menampilkan *board* kepada *user*.

- `processMatrix.py`

File ini berperan sebagai penghubung antara `main.py` dengan file CLIPS. Program mengubah data dari `main.py` menjadi berbentuk string yang kemudian dioper ke file CLIPS untuk dijadikan *fact*, Program ini juga berperan sebagai pembentuk *board minesweeper*.

- `minesweeper.clp`

File ini berperan sebagai agen yang melakukan inferensi. Program terdiri atas `deftemplate` yang menyediakan struktur untuk *fact* yang tersedia dan `defrule` yang dapat dimanfaatkan untuk melakukan proses inferensi. *File* ini juga memiliki fungsi yang bertujuan untuk mengekstrak data masukan *user* yang didapat dari file `main.py`.

Tampilan dari hasil *running* program ini dapat dilihat sebagai berikut.



```
Masukkan ukuran board: 5
Masukkan banyaknya mines atau ranjau: 2
Masukkan koordinat mines:
1, 3
2, 4
0 0 1 1 1
0 0 1 m 2
0 0 1 2 m
0 0 0 1 1
0 0 0 0 0
[BLOAD2] File minesweeper.clp is not a binary construct file.
=====
INI RUN KE-1
=====
0 . . . .
. . . .
. . . .
. . . .
. . . .
Membuka tile pada posisi: ( 0 , 0 ) dengan nilai: 0
=====
INI RUN KE-2
=====
0 . . . .
. . . .
. . . .
. . . .
. . . .
=====
```

```
=====
INI RUN KE-166
=====
0 0 1 . .
0 . 1 . .
0 . 1 2 .
0 . 0 1 1
0 0 . 0 0
Membuka tile pada posisi: ( 4 , 2 ) dengan nilai: 0
=====
INI RUN KE-167
=====
0 0 1 . .
0 . 1 . .
0 . 1 2 .
0 . 0 1 1
0 0 0 0 0
=====
INI RUN KE-168
=====
0 0 1 . .
0 . 1 . .
0 . 1 2 .
0 . 0 1 1
0 0 0 0 0
=====
INI RUN KE-169
=====
0 0 1 . .
```

```

=====
INI RUN KE-182
=====
0 0 1 . .
0 . 1 . .
0 . 1 2 .
0 . 0 1 1
0 0 0 0 0
=====
INI RUN KE-183
=====
0 0 1 . .
0 . 1 . .
0 . 1 2 .
0 . 0 1 1
0 0 0 0 0
Flag mine pada posisi: ( 2 , 4 )
=====
INI RUN KE-184
=====
0 0 1 . .
0 . 1 . .
0 . 1 2 m
0 . 0 1 1
0 0 0 0 0
=====
INI RUN KE-185
=====
0 0 1 . .
0 . 1 . .

```

```

0 0 1 1 1
0 0 1 m 2
0 0 1 2 m
0 0 0 1 1
0 0 0 0 0
=====
INI RUN KE-729
=====
0 0 1 1 1
0 0 1 m 2
0 0 1 2 m
0 0 0 1 1
0 0 0 0 0
=====
INI RUN KE-730
=====
0 0 1 1 1
0 0 1 m 2
0 0 1 2 m
0 0 0 1 1
0 0 0 0 0
=====
INI RUN KE-731
=====
0 0 1 1 1
0 0 1 m 2
0 0 1 2 m
0 0 0 1 1
0 0 0 0 0

```

Mines yang diprediksi oleh agen adalah sebagai berikut:
Flag mine pada posisi: (1 , 3)
Flag mine pada posisi: (2 , 4)

Pada tiap run, akan ditampilkan matriks yang merepresentasikan keadaan *board minesweeper*. Keterangan representasi *board minesweeper* yang terdapat pada elemen matriks dapat dilihat pada tabel berikut.

Nilai pengisi elemen matriks	Keterangan
.	Agen belum mencapai elemen tersebut (kotak belum dipilih/dibuka)
Bilangan bulat (0, 1, 2, dst.)	Agen telah memilih elemen tersebut (kotak telah dipilih/dibuka) dan kotak berisi angka yang tercantum
m	Agen telah memilih elemen tersebut (kotak telah dipilih/dibuka) dan kotak berisi <i>mine/bomb</i>

BAB III

User Manual

Aplikasi ini merupakan program agen penyelesaian permainan *minesweeper* secara otomatis yang dengan memanfaatkan bahasa Python dan program CLIPS. Untuk menjalankan program, *user* harus mengunduh beberapa hal sebagai berikut.

1. Python

Tautan unduh: <https://www.python.org/downloads/>

- Library clips

Tautan unduh: <https://pypi.org/project/clips/>

- Library clipspy

Tautan unduh: <https://pypi.org/project/clipspy/>

2. CLIPS

Tautan unduh: <https://sourceforge.net/projects/clipsrules/>

3. Kode program

Tautan unduh: <https://github.com/gilliantuerah/GGGG-Minesweeper>

Jika *user* telah mengunduh seluruh kelengkapan yang dibutuhkan, *user* dapat melakukan langkah-langkah berikut.

1. Buka terminal pada direktori (folder) tempat kode program tersimpan
2. Jalankan file main.py

Dapat dilakukan dengan menulis perintah `python main.py` pada terminal

3. Masukkan data-data berikut (sesuai urutan)

- a. Ukuran *board*
- b. Banyaknya *mine* (bom) pada *board*
- c. Koordinat *mine* pada *board*

Diisikan dengan format “*x, y*” (tanpa tanda petik) dengan *x* adalah baris dan *y* adalah kolom (penomoran mengacu pada penomoran elemen matriks)

4. Program akan menampilkan hasil inferensi berupa hasil iterasi tiap run yang menunjukkan urutan run dan matriks hasil representasi *board minesweeper*

BAB IV

Proses *Updating* dan *Inferencing* Fakta

Di program python, aplikasi *minesweeper* akan *men-generate* matriks yang sudah berisi nilai di setiap kotaknya sesuai dengan masukan ukuran dan koordinat ranjau/*mines* oleh pengguna. Data-data yang diperoleh dari masukan pengguna melalui program Python kemudian dimasukkan ke dalam *working memory* agen sebagai fakta awal sebelum agen dijalankan. Fakta ini berupa nilai dari setiap kotak di matriks, banyaknya ranjau yang ada, ukuran *board*, dan koordinat ranjau yang ada. Fakta-fakta ini nantinya akan digunakan untuk mengambil nilai saat pembangkitan kotak di *board* agen.

Setelah dijalankan, agen akan menjalankan rule “initialize” dan membangkitkan beberapa fakta awal yaitu memilih titik (0,0) untuk dibuka terlebih dahulu. Kemudian, dari fakta yang diperoleh, akan dijalankan beberapa rule untuk membuka kotak di sekitarnya sebagai berikut:

1. Pertama-tama, akan dijalankan rule “**openUnOpenedBox**” yaitu menambah fakta kotak yang dipilih terbuka dan nilai dari kotak tersebut.
2. Setelah itu, agen akan menjalankan rule “**openBoxZeroValue**” jika kotak yang dipilih agen bernilai 0. Pada rule ini, akan dibangkitkan sebanyak 8 fakta “select” untuk setiap koordinat kotak di sekitar kotak tersebut, dalam kasus ini (0,0).
3. Kemudian, agen akan menjalankan rule “**openOpenedBox**” untuk kotak yang dipilih namun sudah dibuka sebelumnya, pada rule ini, agen hanya akan membuang fakta “select” yang sebelumnya digunakan untuk memilih. Sedangkan, untuk kotak yang dipilih dan belum dibuka sebelumnya, agen akan menjalankan rule “**openUnOpenedBox**” dan menambah fakta kotak yang dipilih terbuka dan nilai dari kotak tersebut.
4. Lalu untuk tetap membuka kotak yang bernilai 0 atau kosong, agen akan menjalankan rule “**openBoxZeroValue**” untuk menambahkan 8 fakta “select” untuk kotak di sekitarnya.
5. Langkah ini akan terus berulang sampai tidak ada lagi kotak kosong yang dipilih.

Untuk melakukan proses *flag* pada kotak yang diprediksi sebagai ranjau, dijalankan rule “**markval**” jika terdapat nilai kotak yang merupakan penjumlahan dari jumlah flag yang ada di sekitar dan jumlah kotak di sekitar yang belum terbuka, maka akan diberi *flag* untuk semua kotak yang belum terbuka di sekitarnya dengan menambahkan fakta “cekMarkVal” untuk setiap kotak di sekitarnya sebagai fakta perantara untuk menjalankan rule “**markValOppened**” jika kotak yang dipilih sudah terbuka kemudian hanya membuang fakta perantara “cekMarkVal”. Sedangkan, agen akan menjalankan rule “**markValUnOppened**” jika kotak yang dipilih belum terbuka. Pada rule ini, akan ditambahkan fakta berupa kotak terbuka dan penandaan *flag* pada kotak, penghapusan fakta perantara, dan pengubahan fakta countMine dengan mengurangi nilainya dengan 1.

Kemudian, jika terdapat kotak dengan nilai yang sama dengan jumlah *flag* yang ada di sekitarnya, maka dapat dipastikan bahwa kotak sekitar yang belum terbuka bukanlah ranjau. Pada kasus ini, agen akan menjalankan rule “**oneMineMarked**” dan menambahkan fakta “select” ke 8 kotak di sekitarnya. Agen kemudian akan menjalankan langkah untuk membuka kotak di sekitarnya seperti yang dilakukan pada 5 langkah di atas.