

APPENDIX

programC45.sql

```
drop database if exists programC45;
```

```
create database programC45;
```

```
use programC45;
```

```
create table tblData
```

```
(
```

```
id_data int,
```

```
ID int,
```

```
inflight_wifi_service int,
```

```
departure_arrival_time int,
```

```
easy_of_booking int,
```

```
gate_location int,
```

```
food_and_drink int,
```

```
online_boarding int,
```

```
seat_comfort int,
```

```
inflight_entertainment int,
```

```
on_board_service int,
```

```
leg_room_service int,
```

```
baggage_handling int,
```

```
checking_service int,
```

```
inflight_service int,
```

```
cleanliness int,
```

```
classification varchar(30)
```

```
);
```

```
LOAD DATA LOCAL INFILE 'filedatatest.csv'  
INTO TABLE tblData  
FIELDS TERMINATED BY ','  
ENCLOSED BY ''''  
LINES TERMINATED BY '\n'  
IGNORE 1 ROWS;
```

```
DELIMITER &&
```

```
CREATE PROCEDURE algoC45()
```

```
BEGIN
```

```
declare rootval int;
```

```
declare branch1val int;
```

```
declare branch2val int;
```

```
declare branch3val int;
```

```
declare branch4val int;
```

```
declare branch5val int;
```

```
declare branch6val int;
```

```
declare branch7val int;
```

```
declare branch8val int;
```

```
declare branch9val int;
```

```
declare branch10val int;
```

```
declare branch11val int;
```

```
declare branch12val int;
```

```
declare branch13val int;
```

```
declare position int;

declare totalinfo int;

declare bestatt varchar(30);

declare class varchar(30);

declare info int;

declare totaldata int;

declare totalsatisfied int;

declare totaldissatisfied int;

declare entropyval double;

declare iteration int;

set position := 0;

set rootval := -1;

set branch1val := 0;

set branch2val := 0;

set branch3val := 0;

set branch4val := 0;

set branch5val := 0;

set branch6val := 0;

set branch7val := 0;

set branch8val := 0;

set branch9val := 0;

set branch10val := 0;

set branch11val := 0;

set branch12val := 0;

set branch13val := 0;
```

```
set iteration := 0;
```

```
CREATE TABLE tblTempData
```

```
(
```

```
id_data int,
```

```
ID int,
```

```
inflight_wifi_service int,
```

```
departure_arrival_time int,
```

```
easy_of_booking int,
```

```
gate_location int,
```

```
food_and_drink int,
```

```
online_boarding int,
```

```
seat_comfort int,
```

```
inflight_entertainment int,
```

```
on_board_service int,
```

```
leg_room_service int,
```

```
baggage_handling int,
```

```
checking_service int,
```

```
inflight_service int,
```

```
cleanliness int,
```

```
classification varchar(30)
```

```
);
```

```
INSERT INTO tblTempData SELECT * FROM tblData;
```

```
WHILE rootval <> 0 DO
```

```
DROP TABLE IF EXISTS tblCount;

CREATE TABLE tblCount
(
    atribut varchar(30),
    informasi int,
    jumlahdata int,
    satisfied int,
    dissatisfied int,
    nilai double,
    Gain double
);

set totaldata :=
(SELECT COUNT(*) FROM tblTempData);

set totalsatisfied :=
(SELECT COUNT(*) FROM tblTempData
WHERE classification = 'satisfied');

set totaldissatisfied :=
(SELECT COUNT(*) FROM tblTempData
WHERE classification = 'neutral or dissatisfied');

if totalsatisfied = 0 OR totaldissatisfied = 0 OR position = 14 then
```

```

if position = 14 then
  if branch13val = 1 then
    if branch12val = 1 then
      if branch11val = 1 then
        if branch10val = 1 then
          if branch9val = 1 then
            if branch8val = 1 then
              if branch7val = 1 then
                if branch6val = 1 then
                  if branch5val = 1 then
                    if branch4val = 1 then
                      if branch3val = 1 then
                        if branch2val = 1 then
                          if branch1val = 1 then
                            set rootval := rootval - 1;
                          end if;
                          set branch1val := branch1val - 1;
                        end if;
                        set branch2val := branch2val - 1;
                      end if;
                      set branch3val := branch3val - 1;
                    end if;
                    set branch4val := branch4val - 1;
                  end if;
                  set branch5val := branch5val - 1;
                end if;
              end if;
            end if;
          end if;
        end if;
      end if;
    end if;
  end if;
end if;

```

```
        set branch6val := branch6val - 1;
    end if;
    set branch7val := branch7val - 1;
end if;
set branch8val := branch8val - 1;
end if;
set branch9val := branch9val - 1;
end if;
set branch10val := branch10val - 1;
end if;
set branch11val := branch11val - 1;
end if;
set branch12val := branch12val - 1;
end if;
set branch13val := branch13val - 1;
elseif position = 13 then
    if branch12val = 1 then
        if branch11val = 1 then
            if branch10val = 1 then
                if branch9val = 1 then
                    if branch8val = 1 then
                        if branch7val = 1 then
                            if branch6val = 1 then
                                if branch5val = 1 then
                                    if branch4val = 1 then
```

```
if branch3val = 1 then
  if branch2val = 1 then
    if branch1val = 1 then
      set rootval := rootval - 1;
    end if;
    set branch1val := branch1val - 1;
  end if;
  set branch2val := branch2val - 1;
end if;
set branch3val := branch3val - 1;
end if;
set branch4val := branch4val - 1;
end if;
set branch5val := branch5val - 1;
end if;
set branch6val := branch6val - 1;
end if;
set branch7val := branch7val - 1;
end if;
set branch8val := branch8val - 1;
end if;
set branch9val := branch9val - 1;
end if;
set branch10val := branch10val - 1;
end if;
set branch11val := branch11val - 1;
```



```

end if;

set branch12val := branch12val - 1;

elseif position = 12 then
  if branch11val = 1 then
    if branch10val = 1 then
      if branch9val = 1 then
        if branch8val = 1 then
          if branch7val = 1 then
            if branch6val = 1 then
              if branch5val = 1 then
                if branch4val = 1 then
                  if branch3val = 1 then
                    if branch2val = 1 then
                      if branch1val = 1 then
                        set rootval := rootval - 1;
                      end if;
                    set branch1val := branch1val - 1;
                  end if;
                set branch2val := branch2val - 1;
              end if;
            set branch3val := branch3val - 1;
          end if;
        set branch4val := branch4val - 1;
      end if;
    set branch5val := branch5val - 1;
  end if;
end if;

```

```
end if;
set branch6val := branch6val - 1;
end if;
set branch7val := branch7val - 1;
end if;
set branch8val := branch8val - 1;
end if;
set branch9val := branch9val - 1;
end if;
set branch10val := branch10val - 1;
end if;
set branch11val := branch11val - 1;
elseif position = 11 then
if branch10val = 1 then
if branch9val = 1 then
if branch8val = 1 then
if branch7val = 1 then
if branch6val = 1 then
if branch5val = 1 then
if branch4val = 1 then
if branch3val = 1 then
if branch2val = 1 then
if branch1val = 1 then
set rootval := rootval - 1;
end if;
```

```
        set branch1val := branch1val - 1;
    end if;
    set branch2val := branch2val - 1;
end if;
set branch3val := branch3val - 1;
end if;
set branch4val := branch4val - 1;
end if;
set branch5val := branch5val - 1;
end if;
set branch6val := branch6val - 1;
end if;
set branch7val := branch7val - 1;
end if;
set branch8val := branch8val - 1;
end if;
set branch9val := branch9val - 1;
end if;
set branch10val := branch10val - 1;

elseif position = 10 then
    if branch9val = 1 then
        if branch8val = 1 then
            if branch7val = 1 then
                if branch6val = 1 then
                    if branch5val = 1 then
```

```
if branch4val = 1 then
  if branch3val = 1 then
    if branch2val = 1 then
      if branch1val = 1 then
        set rootval := rootval - 1;
      end if;
      set branch1val := branch1val - 1;
    end if;
    set branch2val := branch2val - 1;
  end if;
  set branch3val := branch3val - 1;
end if;
set branch4val := branch4val - 1;
end if;
set branch5val := branch5val - 1;
end if;
set branch6val := branch6val - 1;
end if;
set branch7val := branch7val - 1;
end if;
set branch8val := branch8val - 1;
end if;
set branch9val := branch9val - 1;

elseif position = 9 then
  if branch8val = 1 then
```

```
if branch7val = 1 then
  if branch6val = 1 then
    if branch5val = 1 then
      if branch4val = 1 then
        if branch3val = 1 then
          if branch2val = 1 then
            if branch1val = 1 then
              set rootval := rootval - 1;
            end if;
            set branch1val := branch1val - 1;
          end if;
          set branch2val := branch2val - 1;
        end if;
        set branch3val := branch3val - 1;
      end if;
      set branch4val := branch4val - 1;
    end if;
    set branch5val := branch5val - 1;
  end if;
  set branch6val := branch6val - 1;
end if;

set branch7val := branch7val - 1;

end if;

set branch8val := branch8val - 1;

elseif position = 8 then
```

```

if branch7val = 1 then
  if branch6val = 1 then
    if branch5val = 1 then
      if branch4val = 1 then
        if branch3val = 1 then
          if branch2val = 1 then
            if branch1val = 1 then
              set rootval := rootval - 1;
            end if;
            set branch1val := branch1val - 1;
          end if;
          set branch2val := branch2val - 1;
        end if;
        set branch3val := branch3val - 1;
      end if;
      set branch4val := branch4val - 1;
    end if;
    set branch5val := branch5val - 1;
  end if;
  set branch6val := branch6val - 1;
end if;
set branch7val := branch7val - 1;

```

```

elseif position = 7 then
  if branch6val = 1 then
    if branch5val = 1 then

```

```
if branch4val = 1 then
  if branch3val = 1 then
    if branch2val = 1 then
      if branch1val = 1 then
        set rootval := rootval - 1;
      end if;
      set branch1val := branch1val - 1;
    end if;
    set branch2val := branch2val - 1;
  end if;
  set branch3val := branch3val - 1;
end if;
set branch4val := branch4val - 1;
end if;
set branch5val := branch5val - 1;
end if;
set branch6val := branch6val - 1;

elseif position = 6 then
  if branch5val = 1 then
    if branch4val = 1 then
      if branch3val = 1 then
        if branch2val = 1 then
          if branch1val = 1 then
            set rootval := rootval - 1;
          end if;
        end if;
      end if;
    end if;
  end if;
end if;
```

```

    set branch1val := branch1val - 1;

end if;

set branch2val := branch2val - 1;

end if;

set branch3val := branch3val - 1;

end if;

set branch4val := branch4val - 1;

end if;

set branch5val := branch5val - 1;

elseif position = 5 then
    if branch4val = 1 then
        if branch3val = 1 then
            if branch2val = 1 then
                if branch1val = 1 then
                    set rootval := rootval - 1;
                end if;
                set branch1val := branch1val - 1;
            end if;
            set branch2val := branch2val - 1;
        end if;
        set branch3val := branch3val - 1;
    end if;
    set branch4val := branch4val - 1;

elseif position = 4 then

```



```
if branch3val = 1 then
  if branch2val = 1 then
    if branch1val = 1 then
      set rootval := rootval - 1;
    end if;
    set branch1val := branch1val - 1;
  end if;
  set branch2val := branch2val - 1;
end if;
set branch3val := branch3val - 1;

elseif position = 3 then
  if branch2val = 1 then
    if branch1val = 1 then
      set rootval := rootval - 1;
    end if;
    set branch1val := branch1val - 1;
  end if;
  set branch2val := branch2val - 1;

elseif position = 2 then
  if branch1val = 1 then
    set rootval := rootval - 1;
  end if;
  set branch1val := branch1val - 1;
```

```
elseif position = 1 then  
    set rootval := rootval - 1;  
end if;
```

```
set position := 0;
```

```
if totalsatisfied > totaldissatisfied then
```

```
    set class = 'satisfied';
```

```
else
```

```
    set class = 'dissatisfied';
```

```
end if;
```

```
select class;
```

```
DELETE from tblTempData;
```

```
INSERT INTO tblTempData
```

```
SELECT * FROM tblData;
```

```
else
```

```
set iteration := iteration + 1;
```

```
set entrophyval :=
```

```
(-(totaldissatisfied/totaldata)*log2(totaldissatisfied/totaldata))
```

```
+ (-(totalsatisfied/totaldata)*log2(totalsatisfied/totaldata));
```

```
INSERT INTO tblCount (atribut, jumlahdata, satisfied, dissatisfied, nilai)
VALUES ('TOTALDATA', totaldata, totalsatisfied, totaldissatisfied, entrophyval);
```

```
INSERT INTO tblCount (informasi, jumlahdata, satisfied, dissatisfied)
SELECT DISTINCT (a.inflight_wifi_service) AS 'INFLIGHT WIFI SERVICE',
COUNT(a.inflight_wifi_service) AS JUMLAHDATA,
```

```
(
SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'satisfied' AND
b.inflight_wifi_service = a.inflight_wifi_service
) AS satisfied,
```

```
(
SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'neutral or dissatisfied' AND
b.inflight_wifi_service = a.inflight_wifi_service
) AS dissatisfied
```

```
FROM tblTempData AS a
GROUP BY a.inflight_wifi_service;
```

```
UPDATE tblCount set atribut = 'INFLIGHT WIFI SERVICE'
WHERE atribut IS NULL;
```

```
INSERT INTO tblCount (informasi, jumlahdata, satisfied, dissatisfied)
SELECT DISTINCT (a.departure_arrival_time) AS 'DEPARTURE ARRIVAL
TIME',
```

```

COUNT(a.departure_arrival_time) AS JUMLAHDATA,
(
  SELECT COUNT(*) FROM tblTempData AS b
  WHERE b.classification = 'satisfied' AND
  b.departure_arrival_time = a.departure_arrival_time
) AS satisfied,
(
  SELECT COUNT(*) FROM tblTempData AS b
  WHERE b.classification = 'neutral or dissatisfied' AND
  b.departure_arrival_time = a.departure_arrival_time
) AS dissatisfied
FROM tblTempData AS a
GROUP BY a.departure_arrival_time;

UPDATE tblCount set atribut = 'DEPARTURE ARRIVAL TIME'
WHERE atribut IS NULL;

INSERT INTO tblCount (informasi, jumlahdata, satisfied, dissatisfied)
SELECT DISTINCT (a.easy_of_booking) AS 'EASY OF BOOKING',
COUNT(a.easy_of_booking) AS JUMLAHDATA,
(
  SELECT COUNT(*) FROM tblTempData AS b
  WHERE b.classification = 'satisfied' AND
  b.easy_of_booking = a.easy_of_booking
) AS satisfied,
(

```

```

SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'neutral or dissatisfied' AND
b.easy_of_booking = a.easy_of_booking
) AS dissatisfied
FROM tblTempData AS a
GROUP BY a.easy_of_booking;

```

```

UPDATE tblCount set atribut = 'EASY OF BOOKING'
WHERE atribut IS NULL;

```

```

INSERT INTO tblCount (informasi, jumlahdata, satisfied, dissatisfied)
SELECT DISTINCT (a.gate_location) AS 'GATE LOCATION',
COUNT(a.gate_location) AS JUMLAHDATA,
(
SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'satisfied' AND
b.gate_location = a.gate_location
) AS satisfied,
(
SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'neutral or dissatisfied' AND
b.gate_location = a.gate_location
) AS dissatisfied
FROM tblTempData AS a
GROUP BY a.gate_location;

```

```
UPDATE tblCount set atribut = 'GATE LOCATION'
```

```
WHERE atribut IS NULL;
```

```
INSERT INTO tblCount (informasi, jumlahdata, satisfied, dissatisfied)
```

```
SELECT DISTINCT (a.food_and_drink) AS 'FOOD AND DRINK',
```

```
COUNT(a.food_and_drink) AS JUMLAHDATA,
```

```
(
```

```
SELECT COUNT(*) FROM tblTempData AS b
```

```
WHERE b.classification = 'satisfied' AND
```

```
b.food_and_drink = a.food_and_drink
```

```
) AS satisfied,
```

```
(
```

```
SELECT COUNT(*) FROM tblTempData AS b
```

```
WHERE b.classification = 'neutral or dissatisfied' AND
```

```
b.food_and_drink = a.food_and_drink
```

```
) AS dissatisfied
```

```
FROM tblTempData AS a
```

```
GROUP BY a.food_and_drink;
```

```
UPDATE tblCount set atribut = 'FOOD AND DRINK'
```

```
WHERE atribut IS NULL;
```

```
INSERT INTO tblCount (informasi, jumlahdata, satisfied, dissatisfied)
```

```
SELECT DISTINCT (a.online_boarding) AS 'ONLINE BOARDING',
```

```
COUNT(a.online_boarding) AS JUMLAHDATA,
```

```
(
```

```

SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'satisfied' AND
b.online_boarding = a.online_boarding
) AS satisfied,
(
SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'neutral or dissatisfied' AND
b.online_boarding = a.online_boarding
) AS dissatisfied
FROM tblTempData AS a
GROUP BY a.online_boarding;

UPDATE tblCount set atribut = 'ONLINE BOARDING'
WHERE atribut IS NULL;

INSERT INTO tblCount (informasi, jumlahdata, satisfied, dissatisfied)
SELECT DISTINCT (a.seat_comfort) AS 'SEAT COMFORT',
COUNT(a.seat_comfort) AS JUMLAHDATA,
(
SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'satisfied' AND
b.seat_comfort = a.seat_comfort
) AS satisfied,
(
SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'neutral or dissatisfied' AND

```

```
b.seat_comfort = a.seat_comfort
```

```
) AS dissatisfied
```

```
FROM tblTempData AS a
```

```
GROUP BY a.seat_comfort;
```

```
UPDATE tblCount set atribut = 'SEAT COMFORT'
```

```
WHERE atribut IS NULL;
```

```
INSERT INTO tblCount (informasi, jumlahdata, satisfied, dissatisfied)
```

```
SELECT DISTINCT (a.inflight_entertainment) AS 'INFLIGHT  
ENTERTAINMENT',
```

```
COUNT(a.inflight_entertainment) AS JUMLAHDATA,
```

```
(
```

```
SELECT COUNT(*) FROM tblTempData AS b
```

```
WHERE b.classification = 'satisfied' AND
```

```
b.inflight_entertainment = a.inflight_entertainment
```

```
) AS satisfied,
```

```
(
```

```
SELECT COUNT(*) FROM tblTempData AS b
```

```
WHERE b.classification = 'neutral or dissatisfied' AND
```

```
b.inflight_entertainment = a.inflight_entertainment
```

```
) AS dissatisfied
```

```
FROM tblTempData AS a
```

```
GROUP BY a.inflight_entertainment;
```

```
UPDATE tblCount set atribut = 'INFLIGHT ENTERTAINMENT'
```



```
WHERE atribut IS NULL;
```

```
INSERT INTO tblCount (informasi, jumlahdata, satisfied, dissatisfied)
SELECT DISTINCT (a.on_board_service) AS 'ON BOARD SERVICE',
COUNT(a.on_board_service) AS JUMLAHDATA,
```

```
(
SELECT COUNT(*) FROM tblTempData AS b
```

```
WHERE b.classification = 'satisfied' AND
```

```
b.on_board_service = a.on_board_service
```

```
) AS satisfied,
```

```
(
SELECT COUNT(*) FROM tblTempData AS b
```

```
WHERE b.classification = 'neutral or dissatisfied' AND
```

```
b.on_board_service = a.on_board_service
```

```
) AS dissatisfied
```

```
FROM tblTempData AS a
```

```
GROUP BY a.on_board_service;
```

```
UPDATE tblCount set atribut = 'ON BOARD SERVICE'
```

```
WHERE atribut IS NULL;
```

```
INSERT INTO tblCount (informasi, jumlahdata, satisfied, dissatisfied)
SELECT DISTINCT (a.leg_room_service) AS 'LEG ROOM SERVICE',
COUNT(a.leg_room_service) AS JUMLAHDATA,
```

```
(
SELECT COUNT(*) FROM tblTempData AS b
```

```

WHERE b.classification = 'satisfied' AND

b.leg_room_service = a.leg_room_service

) AS satisfied,

(

SELECT COUNT(*) FROM tblTempData AS b

WHERE b.classification = 'neutral or dissatisfied' AND

b.leg_room_service = a.leg_room_service

) AS dissatisfied

FROM tblTempData AS a

GROUP BY a.leg_room_service;

UPDATE tblCount set atribut = 'LEG ROOM SERVICE'

WHERE atribut IS NULL;

INSERT INTO tblCount (informasi, jumlahdata, satisfied, dissatisfied)

SELECT DISTINCT (a.baggage_handling) AS 'BAGGAGE HANDLING',

COUNT(a.baggage_handling) AS JUMLAHDATA,

(

SELECT COUNT(*) FROM tblTempData AS b

WHERE b.classification = 'satisfied' AND

b.baggage_handling = a.baggage_handling

) AS satisfied,

(

SELECT COUNT(*) FROM tblTempData AS b

WHERE b.classification = 'neutral or dissatisfied' AND

b.baggage_handling = a.baggage_handling

```

```

) AS dissatisfied
FROM tblTempData AS a
GROUP BY a.baggage_handling;

```

```

UPDATE tblCount set atribut = 'BAGGAGE HANDLING'
WHERE atribut IS NULL;

```

```

INSERT INTO tblCount (informasi, jumlahdata, satisfied, dissatisfied)
SELECT DISTINCT (a.checking_service) AS 'CHECKING SERVICE',
COUNT(a.checking_service) AS JUMLAHDATA,

```

```

(
SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'satisfied' AND
b.checking_service = a.checking_service
) AS satisfied,

```

```

(
SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'neutral or dissatisfied' AND
b.checking_service = a.checking_service

```

```

) AS dissatisfied
FROM tblTempData AS a
GROUP BY a.checking_service;

```

```

UPDATE tblCount set atribut = 'CHECKING SERVICE'
WHERE atribut IS NULL;

```

```

INSERT INTO tblCount (informasi, jumlahdata, satisfied, dissatisfied)
SELECT DISTINCT (a.inflight_service) AS 'INFLIGHT SERVICE',
COUNT(a.inflight_service) AS JUMLAHDATA,
(
SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'satisfied' AND
b.inflight_service = a.inflight_service
) AS satisfied,
(
SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'neutral or dissatisfied' AND
b.inflight_service = a.inflight_service
) AS dissatisfied
FROM tblTempData AS a
GROUP BY a.inflight_service;

UPDATE tblCount set atribut = 'INFLIGHT SERVICE'
WHERE atribut IS NULL;

INSERT INTO tblCount (informasi, jumlahdata, satisfied, dissatisfied)
SELECT DISTINCT (a.cleanliness) AS 'CLEANLINESS',
COUNT(a.cleanliness) AS JUMLAHDATA,
(
SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'satisfied' AND
b.cleanliness = a.cleanliness

```

```

) AS satisfied,
(
SELECT COUNT(*) FROM tblTempData AS b
WHERE b.classification = 'neutral or dissatisfied' AND
b.cleanliness = a.cleanliness
) AS dissatisfied
FROM tblTempData AS a
GROUP BY a.cleanliness;

UPDATE tblCount set atribut = 'CLEANLINESS'
WHERE atribut IS NULL;

UPDATE tblCount set nilaiI =
(- (satisfied/jumlahdata)*log2(satisfied/jumlahdata))
+ (- (dissatisfied/jumlahdata)*log2(dissatisfied/jumlahdata))
WHERE satisfied > 0 AND dissatisfied > 0;

UPDATE tblCount set nilaiI = 0
WHERE nilaiI IS NULL;

UPDATE tblCount set nilaiI = round(nilaiI,4);

```

```

DROP TABLE IF EXISTS tblSementara;

```

```

CREATE TABLE tblSementara

```

```

(
atribut varchar(30),

```

Gain double

);

INSERT INTO tblSementara(atribut, gain)

SELECT atribut,

entropyval - SUM((jumlahdata/totaldata)*nilai)

AS HITUNGGAIN

FROM tblCount

GROUP BY atribut;

UPDATE tblCount set gain =

ROUND ((

SELECT tblSementara.gain

FROM tblSementara

WHERE tblSementara.atribut = tblCount.atribut

), 4);

set bestatt := (SELECT atribut FROM tblCount

WHERE gain = (SELECT max(gain) FROM tblCount)

GROUP BY atribut LIMIT 1);

if rootval = -1 and position = 0 then

set rootval := (select count(informasi) from tblCount

where atribut = bestatt);

elseif branch1val = 0 and position = 1 then

```
set branch1val := (select count(informasi) from tblCount
where atribut = bestatt);
```

```
elseif branch2val = 0 and position = 2 then
set branch2val := (select count(informasi) from tblCount
where atribut = bestatt);
```

```
elseif branch3val = 0 and position = 3 then
set branch3val := (select count(informasi) from tblCount
where atribut = bestatt);
```

```
elseif branch4val = 0 and position = 4 then
set branch4val := (select count(informasi) from tblCount
where atribut = bestatt);
```

```
elseif branch5val = 0 and position = 5 then
set branch5val := (select count(informasi) from tblCount
where atribut = bestatt);
```

```
elseif branch6val = 0 and position = 6 then
set branch6val := (select count(informasi) from tblCount
where atribut = bestatt);
```

```
elseif branch7val = 0 and position = 7 then
set branch7val := (select count(informasi) from tblCount
where atribut = bestatt);
```

```
elseif branch8val = 0 and position = 8 then  
    set branch8val := (select count(informasi) from tblCount  
    where atribut = bestatt);
```

```
elseif branch9val = 0 and position = 9 then  
    set branch9val := (select count(informasi) from tblCount  
    where atribut = bestatt);
```

```
elseif branch10val = 0 and position = 10 then  
    set branch10val := (select count(informasi) from tblCount  
    where atribut = bestatt);
```

```
elseif branch11val = 0 and position = 11 then  
    set branch11val := (select count(informasi) from tblCount  
    where atribut = bestatt);
```

```
elseif branch12val = 0 and position = 12 then  
    set branch12val := (select count(informasi) from tblCount  
    where atribut = bestatt);
```

```
elseif branch13val = 0 and position = 13 then  
    set branch13val := (select count(informasi) from tblCount  
    where atribut = bestatt);
```

```
end if;
```



```
SELECT * FROM tblCount;
```

```
DROP TABLE IF EXISTS tblInfo;
```

```
CREATE TABLE tblInfo
```

```
(  
    no int not null auto_increment,  
    informasi int,  
    primary key (no)  
);
```

```
INSERT INTO tblInfo (informasi)
```

```
SELECT informasi FROM tblCount
```

```
WHERE atribut = bestatt;
```

```
set totalinfo := (SELECT COUNT(*) FROM tblInfo);
```

```
if position = 13 then
```

```
    set info := (select informasi from tblInfo
```

```
    where no = (totalinfo - branch13val + 1) );
```

```
elseif position = 12 then
```

```
    set info := (select informasi from tblInfo
```

```
    where no = (totalinfo - branch12val + 1) );
```

```
elseif position = 11 then
```

```
    set info := (select informasi from tblInfo
```

```
    where no = (totalinfo - branch11val + 1) );
```

```
elseif position = 10 then
    set info := (select informasi from tblInfo
    where no = (totalinfo - branch10val + 1) );
elseif position = 9 then
    set info := (select informasi from tblInfo
    where no = (totalinfo - branch9val + 1) );
elseif position = 8 then
    set info := (select informasi from tblInfo
    where no = (totalinfo - branch8val + 1) );
elseif position = 7 then
    set info := (select informasi from tblInfo
    where no = (totalinfo - branch7val + 1) );
elseif position = 6 then
    set info := (select informasi from tblInfo
    where no = (totalinfo - branch6val + 1) );
elseif position = 5 then
    set info := (select informasi from tblInfo
    where no = (totalinfo - branch5val + 1) );
elseif position = 4 then
    set info := (select informasi from tblInfo
    where no = (totalinfo - branch4val + 1) );
elseif position = 3 then
    set info := (select informasi from tblInfo
    where no = (totalinfo - branch3val + 1) );
elseif position = 2 then
    set info := (select informasi from tblInfo
```

```

where no = (totalinfo - branch2val + 1) );
elseif position = 1 then
  set info := (select informasi from tblInfo
  where no = (totalinfo - branch1val + 1) );
elseif position = 0 then
  set info := (select informasi from tblInfo
  where no = (totalinfo - rootval + 1) );
end if;

set position := position + 1;

if bestatt = 'INFLIGHT WIFI SERVICE' then
  delete from tblTempData
  where inflight_wifi_service <> info;
end if;

if bestatt = 'DEPARTURE ARRIVAL TIME' then
  delete from tblTempData
  where departure_arrival_time <> info;
end if;

if bestatt = 'EASY OF BOOKING' then
  delete from tblTempData
  where easy_of_booking <> info;
end if;

```

```
if bestatt = 'GATE LOCATION' then  
    delete from tblTempData  
    where gate_location <> info;  
end if;
```

```
if bestatt = 'FOOD AND DRINK' then  
    delete from tblTempData  
    where food_and_drink <> info;  
end if;
```

```
if bestatt = 'ONLINE BOARDING' then  
    delete from tblTempData  
    where online_boarding <> info;  
end if;
```

```
if bestatt = 'SEAT COMFORT' then  
    delete from tblTempData  
    where seat_comfort <> info;  
end if;
```

```
if bestatt = 'INFLIGHT ENTERTAINMENT' then  
    delete from tblTempData  
    where inflight_entertainment <> info;  
end if;
```

```
if bestatt = 'ON BOARD SERVICE' then
```

```
delete from tblTempData  
where on_board_service <> info;  
end if;
```

```
if bestatt = 'LEG ROOM SERVICE' then  
delete from tblTempData  
where leg_room_service <> info;  
end if;
```

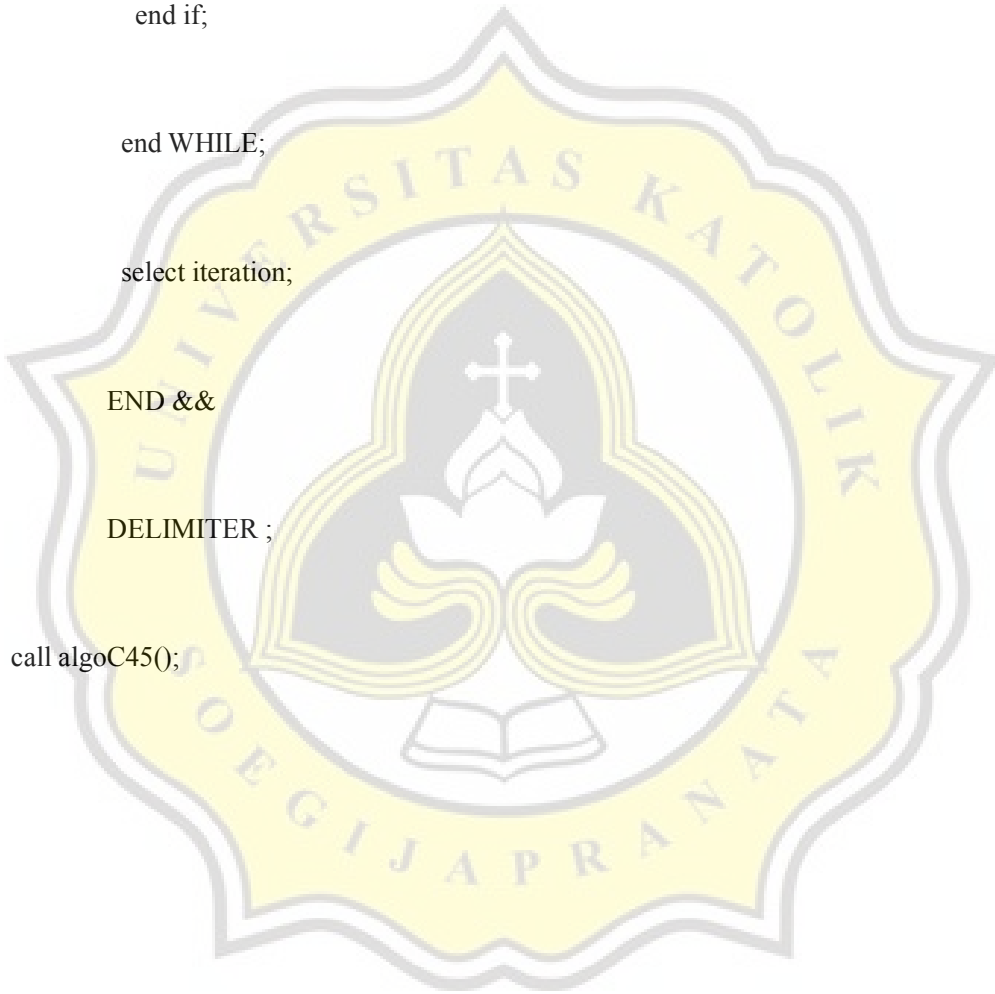
```
if bestatt = 'BAGGAGE HANDLING' then  
delete from tblTempData  
where baggage_handling <> info;  
end if;
```

```
if bestatt = 'CHECKING SERVICE' then  
delete from tblTempData  
where checking_service <> info;  
end if;
```

```
if bestatt = 'INFLIGHT SERVICE' then  
delete from tblTempData  
where inflight_service <> info;  
end if;
```

```
if bestatt = 'CLEANLINESS' then  
delete from tblTempData
```

```
where cleanliness <> info;  
end if;  
  
select iteration,bestatt,info;  
  
end if;  
  
end WHILE;  
  
select iteration;  
  
END &&  
  
DELIMITER ;  
  
call algoC45());
```





2% PLAGIARISM
APPROXIMATELY

Report #14093329

Introduction BackgroundThe form of aircraft passenger satisfaction is an assessment given by airplane passengers to airlines based on the facilities provided to its passengers.

There are various kinds of airline facilities for which an assessment can be carried out by passengers to the airline.

With this program, it aims to determine whether or not passengers are satisfied based on the facilities provided by the airline in order to provide positive benefits for passengers and airlines as well. The positive advantages that aircraft passengers get are the comfort when traveling on an airplane by choosing a good airline, as well as the benefits that airlines get is that they can be better in the future based on the assessment given by passengers. As well as prospective passengers who want to use the airline for travel can find out the best airline to use. For this reason, the use of this decision tree algorithm is very suitable and precise. This program is designed so that airlines can input passenger assessment data. This action is