

## APPENDIX

### Code Membuat Database dan Tabel

```
1. DROP DATABASE IF EXISTS skripsi;
2. CREATE DATABASE skripsi;
3. USE skripsi;
4.
5. CREATE TABLE tblDataProduk(
6.   id VARCHAR(100),
7.   branch VARCHAR(10),
8.   city VARCHAR(100),
9.   customerType VARCHAR(10),
10.  Gender VARCHAR(10),
11.  productLine VARCHAR(100),
12.  unitPrice decimal,
13.  qty int,
14.  tax decimal,
15.  total decimal,
16.  tgl datetime,
17.  payment VARCHAR(20),
18.  cogs decimal,
19.  grossIncome decimal,
20.  rating decimal,
21.  unitPriceFix VARCHAR(10),
22.  taxFix VARCHAR(10),
23.  totalFix VARCHAR(10),
24.  month VARCHAR(10),
25.  cogsFix VARCHAR(10),
26.  grossIncomeFix VARCHAR(10),
27.  ratingFix VARCHAR(5)
28. );
29.
30. CREATE TABLE tblData(
31.   id VARCHAR(100),
32.   branch VARCHAR(10),
33.   city VARCHAR(100),
34.   customerType VARCHAR(10),
35.   Gender VARCHAR(10),
36.   productLine VARCHAR(100),
37.   unitPrice decimal,
38.   qty int,
39.   tax decimal,
40.   total decimal,
41.   tgl datetime,
42.   payment VARCHAR(20),
43.   cogs decimal,
44.   grossIncome decimal,
45.   rating decimal,
46.   unitPriceFix VARCHAR(20),
47.   taxFix VARCHAR(20),
```

```

48.     totalFix VARCHAR(20),
49.     month VARCHAR(10),
50.     cogsFix VARCHAR(20),
51.     grossIncomeFix VARCHAR(20),
52.     ratingFix VARCHAR(5)
53. );
54.
55. CREATE TABLE sampleData(
56.     no int,
57.     branch VARCHAR(10),
58.     city VARCHAR(10),
59.     customerType VARCHAR(10),
60.     gender VARCHAR(10),
61.     productLine VARCHAR(100),
62.     unitPrice VARCHAR(10),
63.     qty int,
64.     tax VARCHAR(10),
65.     total VARCHAR(10),
66.     month VARCHAR(10),
67.     payment VARCHAR(20),
68.     cogs VARCHAR(10),
69.     grossIncome VARCHAR(10),
70.     rating VARCHAR(5)
71. );

72. LOAD DATA INFILE 'E:/Nugas/Project/coding/datasupermarket2.csv'
73. INTO TABLE tblData
74. FIELDS TERMINATED BY ','
75. ENCLOSED BY '"'
76. LINES TERMINATED BY '\n'
77. IGNORE 1 ROWS;
78.
79. LOAD DATA INFILE 'E:/Nugas/Project/coding/datasupermarket2.csv'
80. INTO TABLE tblDataProduk
81. FIELDS TERMINATED BY ','
82. ENCLOSED BY '"'
83. LINES TERMINATED BY '\n'
84. IGNORE 1 ROWS;
85.
86. create table tblHitung(
87.     atribut varchar(20),
88.     informasi varchar(50),
89.     jumlahdata int,
90.     ratingLow int,
91.     ratingHigh int,
92.     nilaiI double,
93.     gain double
94. );
95.
96. create table tblHasil(
97.     nomer int primary key AUTO_INCREMENT,
98.     atribut varchar(20)
99. );
100.
101. create table tblAtribut

```

```

102. (
103.   atribut varchar(50),
104.   informasi varchar(50),
105.   jumlahdata int,
106.   nilaiI double,
107.   gain double
108. );
109.

```

### Procedure C45

```

110. delimiter ??
111. create procedure c45()
112. BEGIN
113.
114. declare a int default 0;
115.
116. looping: WHILE (a <> 1) do
117.
118. delete from tblHitung;
119. delete from tblAtribut;
120.
121. select @jumlahdata:=count(*)
122. from tblData;
123.
124. select @ratingLow:=count(*)
125. from tblData
126. where ratingFix LIKE ('%LR%');
127.
128. select @ratingHigh:=count(*)
129. from tblData
130. where ratingFix LIKE ('%HR%');
131.
132. select @nilaiI:=(-
      (@ratingLow/@jumlahdata)*log2(@ratingLow/@jumlahdata)
133. +
134. (-(@ratingHigh/@jumlahdata)*log2(@ratingHigh/@jumlahdata));
135.
136. select @jumlahdata as TOTALDATA,
137. @ratingLow as LOW_RATING,
138. @ratingHigh as HIGH_RATING,
139. @nilaiI as ENTROPY;
140.
141. insert into tblHitung (atribut, jumlahdata, ratingLow, ratingHigh, Nil
      aiI)
142. values
143. ('TOTAL DATA', @jumlahdata, @ratingLow, @ratingHigh, @nilaiI);
144. select *from tblHitung;
145.
146. insert into tblHitung
147. (informasi, jumlahdata, ratingLow, ratingHigh)
148. select distinct(A.branch) as BRANCH, count(A.branch) as JUMLAHDATA,
149. (
150.   select COUNT(*)

```

```

151.     from tblData as B
152.     where B.ratingFix LIKE ('%LR%') and
153.     B.branch = A.branch
154. )AS RATINGLOW,
155. (
156.     select COUNT(*)
157.     from tblData as C
158.     where C.ratingFix LIKE ('%HR%') and
159.     C.branch = A.branch
160. )as RATINGHIGH
161. from tblData as A
162. group by A.branch;
163.

164. update tblHitung set atribut = 'branch'
165.     where atribut is NULL;
166.
167. insert into tblHitung
168.     (informasi, jumlahdata, ratingLow, ratingHigh)
169. select distinct(A.city) as CITY, count(A.city) as JUMLAHDATA,
170.     (
171.         select COUNT(*)
172.         from tblData as B
173.         where B.ratingFix LIKE ('%LR%') and
174.         B.city = A.city
175.     )AS RATINGLOW,
176.     (
177.         select COUNT(*)
178.         from tblData as C
179.         where C.ratingFix LIKE ('%HR%') and
180.         C.city = A.city
181.     )as RATINGHIGH
182. from tblData as A
183. group by A.city;
184.
185.     update tblHitung set atribut = 'city'
186.     where atribut is NULL;
187.
188. insert into tblHitung
189.     (informasi, jumlahdata, ratingLow, ratingHigh)
190. select distinct(A.customerType) as CUSTOMER_TYPE, count(A.customerTy
191. pe) as JUMLAHDATA,
192.     (
193.         select COUNT(*)
194.         from tblData as B
195.         where B.ratingFix LIKE ('%LR%') and
196.         B.customerType = A.customerType
197.     )AS RATINGLOW,
198.     (
199.         select COUNT(*)
200.         from tblData as C
201.         where C.ratingFix LIKE ('%HR%') and
202.         C.customerType = A.customerType
203.     )as RATINGHIGH
203. from tblData as A

```

```

204. group by A.customerType;
205.
206.     update tblHitung set atribut = 'customerType'
207.     where atribut is NULL;
208.
209. insert into tblHitung
210.     (informasi, jumlahdata, ratingLow, ratingHigh)
211. select distinct(A.gender) as GENDER, count(A.gender) as JUMLAHDATA,
212.     (
213.     select COUNT(*)
214.     from tblData as B
215.     where B.ratingFix LIKE ('%LR%') and
216.     B.gender = A.gender
217.     )AS RATINGLOW,
218.     (
219.     select COUNT(*)
220.     from tblData as C
221.     where C.ratingFix LIKE ('%HR%') and
222.     C.gender = A.gender
223.     )as RATINGHIGH
224. from tblData as A
225. group by A.gender;
226.
227.     update tblHitung set atribut = 'GENDER'
228.     where atribut is NULL;
229.
230. insert into tblHitung
231.     (informasi, jumlahdata, ratingLow, ratingHigh)
232. select distinct(A.productLine) as PRODUCT_LINE, count(A.productLine)
as JUMLAHDATA,
233.     (
234.     select COUNT(*)
235.     from tblData as B
236.     where B.ratingFix LIKE ('%LR%') and
237.     B.productLine = A.productLine
238.     )AS RATINGLOW,
239.     (
240.     select COUNT(*)
241.     from tblData as C
242.     where C.ratingFix LIKE ('%HR%') and
243.     C.productLine = A.productLine
244.     )as RATINGHIGH
245. from tblData as A
246. group by A.productLine;
247.
248. update tblHitung set atribut = 'productLine'
249.     where atribut is NULL;
250.
251. insert into tblHitung
252.     (informasi, jumlahdata, ratingLow, ratingHigh)
253. select distinct(A.unitPriceFix) as UNIT_PRICE, count(A.unitPriceFix)
as JUMLAHDATA,
254.     (
255.     select COUNT(*)
256.     from tblData as B

```

```

257.         where B.ratingFix LIKE ('%LR%') and
258.         B.unitPriceFix = A.unitPriceFix
259.     )AS RATINGLOW,
260.     (
261.         select COUNT(*)
262.         from tblData as C
263.         where C.ratingFix LIKE ('%HR%') and
264.         C.unitPriceFix = A.unitPriceFix
265.     )as RATINGHIGH
266. from tblData as A
267. group by A.unitPriceFix;
268.
269. update tblHitung set atribut = 'unitPriceFix'
270.     where atribut is NULL;
271.
272. insert into tblHitung
273.     (informasi, jumlahdata, ratingLow, ratingHigh)
274. select distinct(A.qty) as QUANTITY, count(A.qty) as JUMLAHDATA,
275.     (
276.         select COUNT(*)
277.         from tblData as B
278.         where B.ratingFix LIKE ('%LR%') and
279.         B.qty = A.qty
280.     )AS RATINGLOW,
281.     (
282.         select COUNT(*)
283.         from tblData as C
284.         where C.ratingFix LIKE ('%HR%') and
285.         C.qty = A.qty
286.     )as RATINGHIGH
287. from tblData as A
288. group by A.qty;
289.
290. update tblHitung set atribut = 'qty'
291.     where atribut is NULL;
292.
293. insert into tblHitung
294.     (informasi, jumlahdata, ratingLow, ratingHigh)
295. select distinct(A.taxFix) as TAX, count(A.taxFix) as JUMLAHDATA,
296.     (
297.         select COUNT(*)
298.         from tblData as B
299.         where B.ratingFix LIKE ('%LR%') and
300.         B.taxFix = A.taxFix
301.     )AS RATINGLOW,
302.     (
303.         select COUNT(*)
304.         from tblData as C
305.         where C.ratingFix LIKE ('%HR%') and
306.         C.taxFix = A.taxFix
307.     )as RATINGHIGH
308. from tblData as A
309. group by A.taxFix;
310.

```

```

311. update tblHitung set atribut = 'taxFix'
312.     where atribut is NULL;
313.

314. insert into tblHitung
315.     (informasi, jumlahdata, ratingLow, ratingHigh)
316.     select distinct(A.totalFix) as TOTAL, count(A.totalFix) as JUMLAHDAT
A,
317.     (
318.         select COUNT(*)
319.         from tblData as B
320.         where B.ratingFix LIKE ('%LR%') and
321.         B.totalFix = A.totalFix
322.     )AS RATINGLOW,
323.     (
324.         select COUNT(*)
325.         from tblData as C
326.         where C.ratingFix LIKE ('%HR%') and
327.         C.totalFix = A.totalFix
328.     )as RATINGHIGH
329.     from tblData as A
330.     group by A.totalFix;
331.
332. update tblHitung set atribut = 'totalFix'
333.     where atribut is NULL;
334.

335. insert into tblHitung
336.     (informasi, jumlahdata, ratingLow, ratingHigh)
337.     select distinct(A.month) as MONTH, count(A.month) as JUMLAHDATA,
338.     (
339.         select COUNT(*)
340.         from tblData as B
341.         where B.ratingFix LIKE ('%LR%') and
342.         B.month = A.month
343.     )AS RATINGLOW,
344.     (
345.         select COUNT(*)
346.         from tblData as C
347.         where C.ratingFix LIKE ('%HR%') and
348.         C.month = A.month
349.     )as RATINGHIGH
350.     from sampleData as A
351.     group by A.month;
352.
353. update tblHitung set atribut = 'month'
354.     where atribut is NULL;
355.

356. insert into tblHitung
357.     (informasi, jumlahdata, ratingLow, ratingHigh)
358.     select distinct(A.payment) as PAYMENT, count(A.payment) as JUMLAHDAT
A,
359.     (
360.         select COUNT(*)

```

```

361.         from tblData as B
362.         where B.ratingFix LIKE ('%LR%') and
363.         B.payment = A.payment
364.     )AS RATINGLOW,
365.     (
366.         select COUNT(*)
367.         from tblData as C
368.         where C.ratingFix LIKE ('%HR%') and
369.         C.payment = A.payment
370.     )as RATINGHIGH
371. from tblData as A
372. group by A.payment;
373.
374. update tblHitung set atribut = 'payment'
375.     where atribut is NULL;
376.
377. insert into tblHitung
378.     (informasi, jumlahdata, ratingLow, ratingHigh)
379. select distinct(A.cogsFix) as COGS, count(A.cogsFix) as JUMLAHDATA
380. /
381.     (
382.         select COUNT(*)
383.         from tblData as B
384.         where B.ratingFix LIKE ('%LR%') and
385.         B.cogsFix = A.cogsFix
386.     )AS RATINGLOW,
387.     (
388.         select COUNT(*)
389.         from tblData as C
390.         where C.ratingFix LIKE ('%HR%') and
391.         C.cogsFix = A.cogsFix
392.     )as RATINGHIGH
393. from tblData as A
394. group by A.cogsFix;
395. update tblHitung set atribut = 'cogsFix'
396.     where atribut is NULL;
397.
398. insert into tblHitung
399.     (informasi, jumlahdata, ratingLow, ratingHigh)
400. select distinct(A.grossIncomeFix) as PAYMENT, count(A.grossIncome
401.     eFix) as JUMLAHDATA,
402.     (
403.         select COUNT(*)
404.         from tblData as B
405.         where B.ratingFix LIKE ('%LR%') and
406.         B.grossIncomeFix = A.grossIncomeFix
407.     )AS RATINGLOW,
408.     (
409.         select COUNT(*)
410.         from tblData as C
411.         where C.ratingFix LIKE ('%HR%') and
412.         C.grossIncomeFix = A.grossIncomeFix

```



```

412.         )as RATINGHIGH
413.     from tblData as A
414.     group by A.grossIncomeFix;
415.
416.     update tblHitung set atribut = 'grossIncomeFix'
417.         where atribut is NULL;
418.

419.         /*hitung nilai entropy*/
420.     update tblHitung set nilaiI =
421.         (- (ratingLow/jumlahdata) *log2 (ratingLow/jumlahdata))
422.         +
423.         (- (ratingHigh/jumlahdata) *log2 (ratingHigh/jumlahdata));
424.     /*salah satu nilai 0 pasti hasilnya 0*/
425.     update tblHitung set nilaiI = 0
426.         where nilaiI is NULL;
427.     /*bulatkan 4 digit di belakang koma*/
428.     update tblHitung set nilaiI = ROUND (nilaiI,4);
429.     select *from tblHitung;
430.
431.     /*menghitung gain*/
432.     drop table if exists tblTampung;
433.     create temporary table tblTampung
434.     (
435.         atribut varchar(20),
436.         gain double
437.     );
438.
439.     insert into tblTampung(atribut, gain)
440.     select atribut, @nilaiI - SUM((jumlahdata/@jumlahdata)*nilaiI) as HITUNGGAIN
441.         from tblHitung
442.         group by atribut;
443.
444.     select *from tblTampung; /*lihat hasil hitungan gain nya*/
445.
446.     /*hasil dari hasil tampung di update kembali ke tabel Hitung
*/
447.     update tblHitung set gain =
448.         ROUND (
449.             (
450.                 select tblTampung.gain
451.                 from tblTampung
452.                 where tblTampung.atribut = tblHitung.atribut
453.             ),4);
454.     select *from tblHitung;
455.     select * from tblHitung order by gain;
456.     -- select * from tblHasil;
457.
458.
459.
460.     insert into tblAtribut(atribut, informasi, jumlahdata, nilaiI, gain)
461.     select atribut, informasi, jumlahdata, nilaiI, gain

```

```

462.         from tblHitung
463.         where gain = (select max(gain) from tblHitung);
464.
465.         select * from tblAtribut;
466.
467.         -- select informasi from tblAtribut;
468.         select distinct @tampungAtribut:= atribut
469.         from tblAtribut
470.         where gain = (select max(gain) from tblAtribut);
471.
472.         select @tampungInformasi:= informasi
473.         from tblAtribut
474.         where nilaiI= (select max(nilaiI) from tblAtribut);
475.
476.         select @tampungTotalData:= jumlahdata
477.         from tblHitung
478.         where atribut = 'TOTAL DATA';
479.
480.         insert into tblHasil (atribut) values(@tampungAtribut);
481.         select * from tblHasil;
482.
483.         -- /*iterasi 2*/
484.
485.
486.         if(@tampungAtribut = 'branch') then
487.
488.         create table tblData2 as
489.             select id from tblData where branch = @tampungInformas
490.         i;
491.
492.         delete from tblData
493.             where id not in (select id from tblData2);
494.
495.         drop table tblData2;
496.
497.         elseif(@tampungAtribut = 'Gender') then
498.
499.         create table tblData2 as
500.             select id from tblData where gender = @tampungInformas
501.         i;
502.
503.         delete from tblData
504.             where id not in (select id from tblData2);
505.
506.         drop table tblData2;
507.
508.         elseif(@tampungAtribut = 'city') then
509.
510.         create table tblData2 as
511.             select id from tblData where city = @tampungInformasi;
512.
513.         delete from tblData
514.             where id not in (select id from tblData2);

```

j

```

515.         drop table tblData2;
516.
517.     elseif(@tampungAtribut = 'customerType') then
518.
519.         create table tblData2 as
520.             select id from tblData where customerType = @tampungIn
formasi;
521.
522.         delete from tblData
523.             where id not in (select id from tblData2);
524.
525.         drop table tblData2;
526.
527.     elseif(@tampungAtribut = 'productLine') then
528.
529.         create table tblData2 as
530.             select id from tblData where productLine = @tampungInf
ormasi;
531.
532.         delete from tblData
533.             where id not in (select id from tblData2);
534.
535.         drop table tblData2;
536.
537.     elseif(@tampungAtribut = 'unitPriceFix') then
538.
539.         create table tblData2 as
540.             select id from tblData where unitPriceFix = @tampungIn
formasi;
541.
542.         delete from tblData
543.             where id not in (select id from tblData2);
544.
545.         drop table tblData2;
546.
547.     elseif(@tampungAtribut = 'qty') then
548.
549.         create table tblData2 as
550.             select id from tblData where qty = @tampungInformasi;
551.
552.         delete from tblData
553.             where id not in (select id from tblData2);
554.
555.         drop table tblData2;
556.
557.     elseif(@tampungAtribut = 'taxFix') then
558.
559.         create table tblData2 as
560.             select id from tblData where taxFix = @tampungInformas
i;
561.
562.         delete from tblData
563.             where id not in (select id from tblData2);
564.
565.         drop table tblData2;

```

```

566.
567.         elseif(@tampungAtribut = 'totalFix') then
568.
569.             create table tblData2 as
570.                 select id from tblData where totalFix = @tampungInform
asi;
571.
572.             delete from tblData
573.                 where id not in (select id from tblData2);
574.
575.             drop table tblData2;
576.
577.         elseif(@tampungAtribut = 'payment') then
578.
579.             create table tblData2 as
580.                 select id from tblData where payment = @tampungInforma
si;
581.
582.             delete from tblData
583.                 where id not in (select id from tblData2);
584.
585.             drop table tblData2;
586.
587.         elseif(@tampungAtribut = 'cogsFix') then
588.
589.             create table tblData2 as
590.                 select id from tblData where cogsFix = @tampungInforma
si;
591.
592.             delete from tblData
593.                 where id not in (select id from tblData2);
594.
595.             drop table tblData2;
596.
597.         elseif(@tampungAtribut = 'grossIncomeFix') then
598.
599.             create table tblData2 as
600.                 select id from tblData where grossIncomeFix = @tampung
Informasi;
601.
602.             delete from tblData
603.                 where id not in (select id from tblData2);
604.
605.             drop table tblData2;
606.         end if;
607.
608.
609. if (@tampungTotalData <= 4) then
610. LEAVE looping;
611. end if;
612. end while looping;
613.
614. END ??
615. delimiter ;

```



**2.38%** PLAGIARISM  
APPROXIMATELY

## Report #13365889

CHAPTER 1 INTRODUCTION 1.2 Background Products sold in stores are a necessity that humans need for everyday life, by selling products or goods can make it easier for people to get the products they need. Of course, there are some products that sell well and often run out of stock because of the high demand. Product buyers also give a rating of how satisfied the buyer is with the product purchased, so that buyers will usually buy the product again or recommend products that are considered to have a high rating. The rating of the product is categorized from 1 (one) to 10 (ten). To predict which products have a high rating, an algorithm is needed to calculate which products have a high rating and a low rating. With this prediction, the seller will prepare more products that are sold out than usual to avoid running out of stock when selling the product. Because usually high rated products have many buyers and require more stock than other products. The collected data will be processed using the C-45 algorithm