BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan pengujian serta analisis hasil dari pengujian terhadap sistem pembuat menu paket otomatis yang sudah selesai dikerjakan pada bab implementasi sebelumnya.

6.1 Pengujian Pengaruh Nilai *Minimum Support*

Pengujian nilai *minimum support* untuk mengetahui pengaruh nilai *minimum support* terhadap hasil kombinasi menu paket. Nilai *minimum support* yang digunakan mulai dari nilai 0 sampai nilai 13.

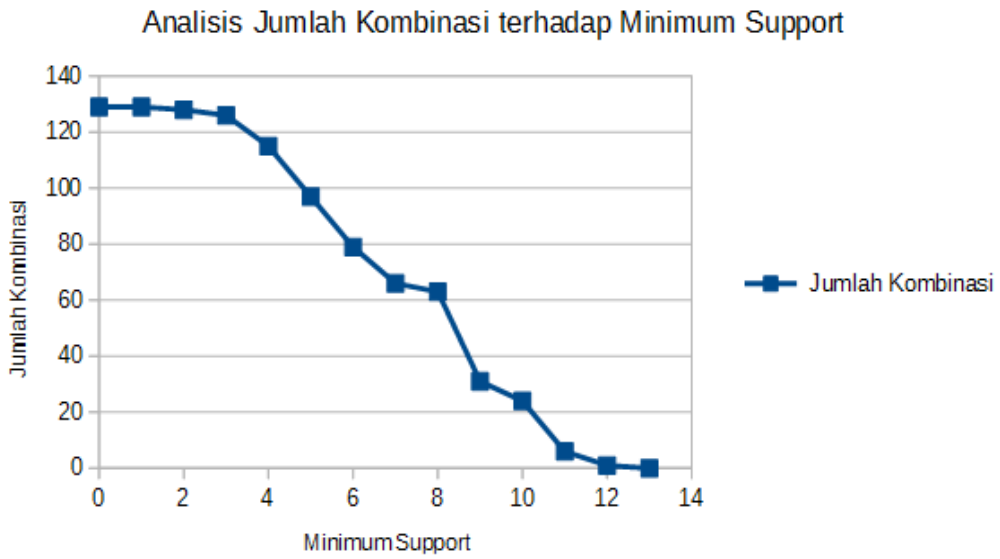
Table 6.1 Hasil Pengujian Jumlah Kombinasi dihasilkan Sistem

No	Nilai *Minimum Support*	Jumlah Kombinasi
1	0	129
2	1	129
3	2	128
4	3	126
5	4	115
6	5	97
7	6	79
8	7	66
9	8	63
10	9	31
11	10	24
12	11	6
13	12	1
14	13	0

Hasil pengujian secara lengkap dapat dilihat pada Lampiran A. Pada kolom nilai *minimum support* adalah nilai uji untuk parameter *minimum support* yang diujikan pada sistem. Pada kolom jumlah kombinasi adalah jumlah variasi menu paket yang dihasilkan dari nilai uji parameter *minimum support* pada sistem.

6.2 Hasil Analisis Pengaruh Nilai *Minimum Support*

6.2.1 Analisis Jumlah Kombinasi terhadap *Minimum Support*



Gambar 6.1 Hasil pengujian perbandingan nilai *minimum support* terhadap jumlah kombinasi

Berdasarkan hasil pengujian, nilai *minimum support* = 0 dan nilai minimum *support* = 1 menghasilkan kombinasi menu paket paling tinggi dengan nilai 129 variasi kombinasi menu paket. Variasi kombinasi menu paket berangsur-angsur berkurang pada nilai *minumum support* yang lebih besar sampai tidak menghasilkan variasi kombinasi menu paket, terlihat pada grafik pada Gambar 6.1. Nilai *minimum support* menentukan *item* menu yang lolos dari eliminasi untuk mengkontruksi *POC-Tree*. Apabila nilai *minimum support* = 0 maka tidak ada *item* menu yang dieliminasi, sedangkan pada nilai *minumum support* = 100 maka semua *item* menu habis dieliminasi. Pada 100 transaksi yang digunakan sebagai data untuk melakukan *mining frequent itemset* menunjukkan bahwa data memiliki sifat divergen, memiliki sedikit pengulangan penggunaan *item* yang sama. Disebabkan jumlah *item* menu yang berukuran besar sebesar 168 *item*.

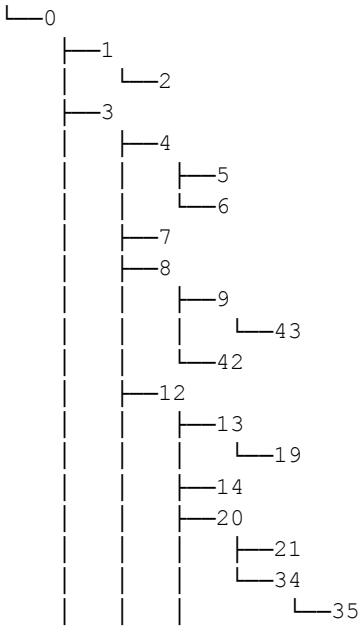
Pada bagian Lampiran Tabel B.1 terlihat bahwa nilai *support* banyak tereliminasi pada sebagian besar *item* menu, apabila digunakan nilai *support* lebih dari 10 untuk melakukan pengujian, disebabkan *item* menu tersebut hanya memiliki nilai *support* kurang dari sama dengan 1. Hanya *id menu* tertentu seperti *id menu* 1, 9, 40, 90, 93, 117, 118, 138, 150, yang mampu bertahan untuk dimasukkan pada *POC-Tree* apabila diuji dengan nilai *support* lebih dari 10. Sehingga menyebabkan hasil *frequent itemset* menggunakan algoritma *FIN* memiliki jumlah variasi kombinasi menu paket yang tinggi pada nilai minimum *support* kurang dari sama dengan 10. Sebaliknya apabila diuji dengan nilai minimal *support* lebih dari 10 maka jumlah variasi hasil kombinasi menu akan menurun drastis. Pada nilai minimal *support* = 10 menghasilkan 24 menu paket, nilai minimal *support* = 11 menghasilkan 6 menu paket, nilai minimal *support* = 12 menghasilkan 1 menu paket. Pada nilai minimal support = 13 sudah tidak ada

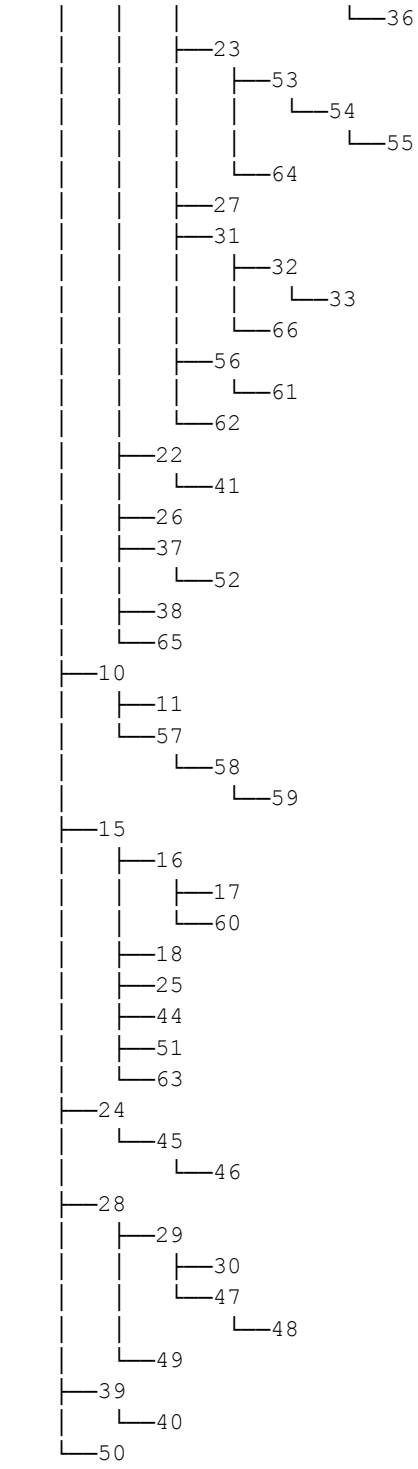
menu paket yang terbentuk, disebabkan *Poc-Tree* yang terbentuk sudah tidak menghasilkan hasil *mining frequent 3-itemset* untuk membentuk menu paket.

Nilai *minimum support* berbanding terbalik dengan jumlah variasi kombinasi menu paket. Apabila nilai *minimum support* semakin besar maka jumlah variasi kombinasi menu paket semakin kecil. Nilai *minimum support* sebagai *eliminator* terhadap *item* menu paket yang memilliki nilai *support* yang lebih kecil dari pada nilai *threshold minimum support*.

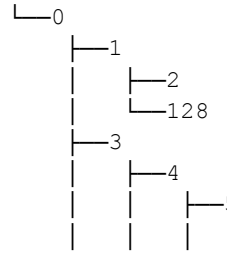
Item penyusun transaksi yang lolos eliminiasi digunakan untuk membentuk *POC-tree*. Variasi menu paket dibentuk oleh *mining frequent 3-itemset* pada *POC-tree*. Berdasarkan hasil variasi menu paket yang terbentuk pada pengujian nilai *minimum suppport* 0 sampai 13. Menunjukkan hasil variasi menu paket menghasilkan kombinasi yang berbeda-beda. Hal ini disebabkan *threshold* nilai *minimum support* sebagai faktor utama pembentukan *POC-tree*. Apabila nilai *minimum support* yang berbeda maka akan menghasilkan bentuk *POC-tree* yang berbeda. Disebabkan variasi *item* penyusun, yang lolos eliminasi, berbeda untuk setiap nilai *minimum support*. Apabila bentuk *POC-tree* berbeda maka akan menghasilkan hasil *mining 3-frequent itemset* yang berbeda pula. Jika hasil *mining 3-itemset* berbeda maka akan menghasilkan variasi menu paket yang berbeda.

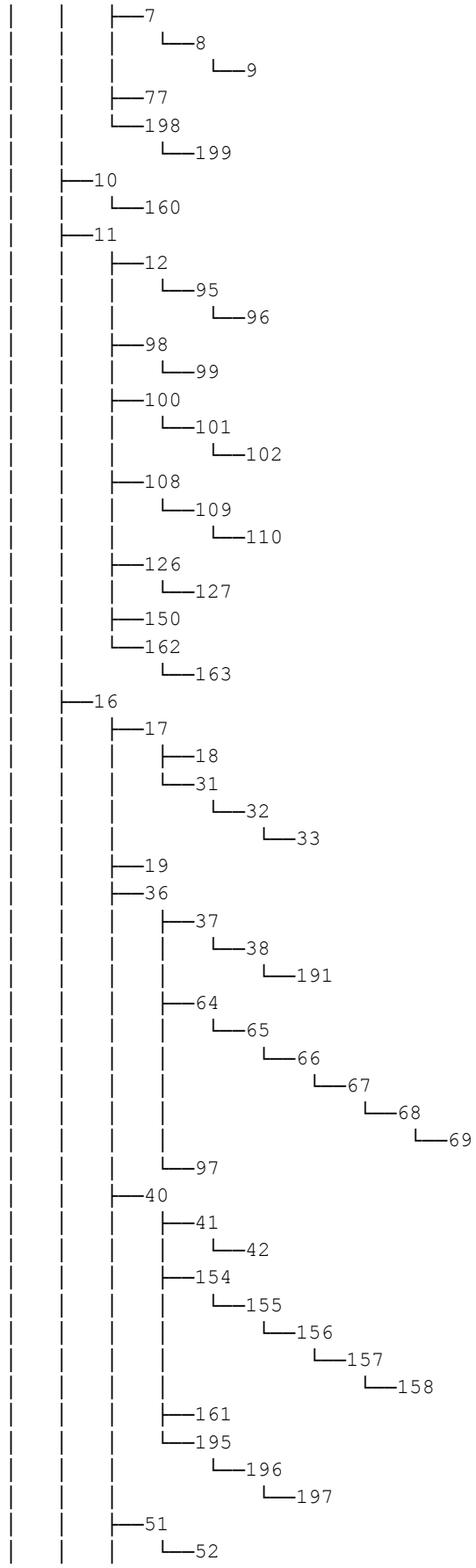
Visualisasi *Poc-Tree* dengan *minimum support* = 11 pada Gambar 6.2 menghasilkan bentuk tree yang berbeda apabila dibandingkan dengan visualisasi *POC-tree* dengan *minimum support* = 5 pada Gambar 6.3. Pada *Poc-Tree* dengan *minimum support* = 5 memiliki anak yang lebih banyak dari pada pada *POC-tree* dengan *minimum support* = 11. Oleh sebab itu pada *Poc-Tree* dengan *minimal support* = 5 menghasilkan hasil irisan lebih banyak, antar 2 hasil *mining frequent 2-itemset* untuk membentuk *frequent 3-itemset*, dari pada pada *Poc-Tree* dengan *minimal support* = 11 yang memiliki ukuran *tree* yang lebih kecil.

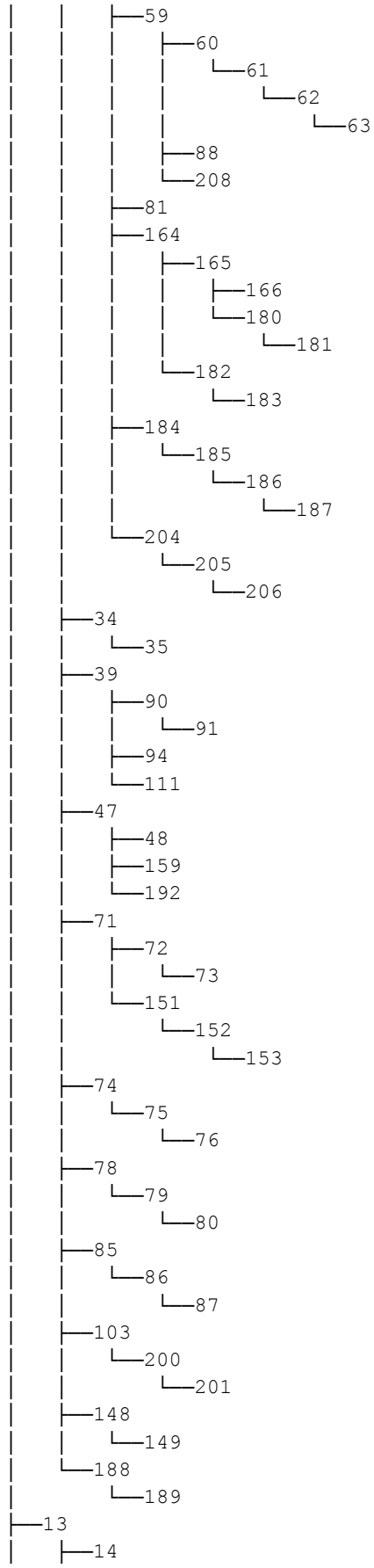


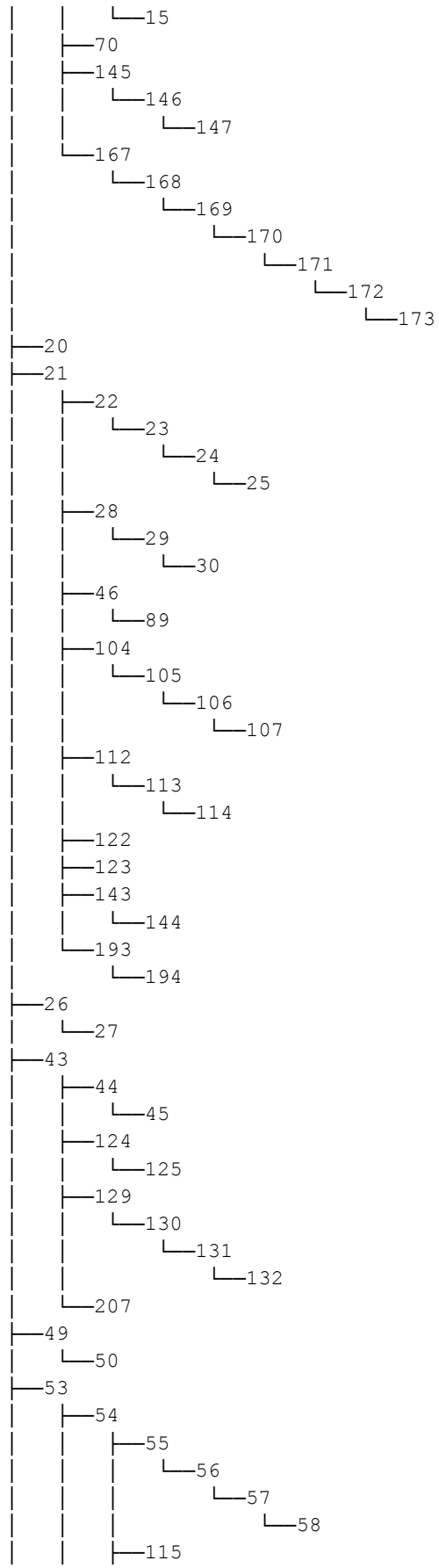


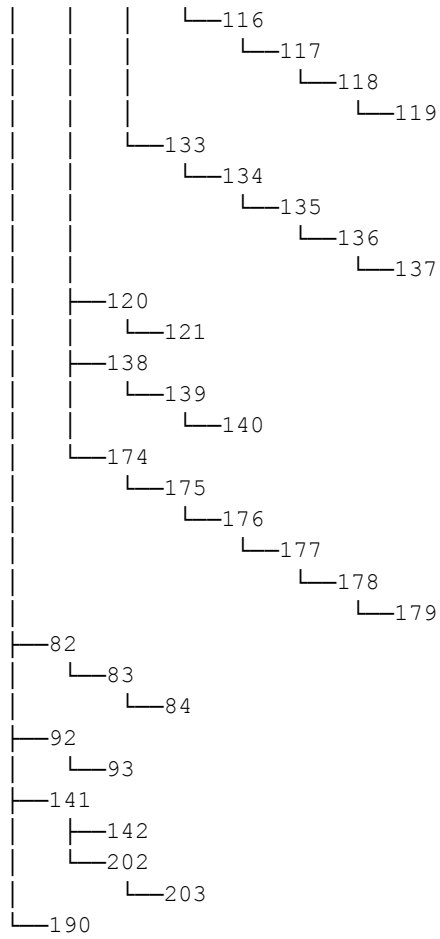
Gambar 6.2 Visualisasi *Poc-Tree* untuk *Minimum Support* = 11




7
8
9
77
198
199
10
160
11
12
95
96
98
99
100
101
102
108
109
110
126
127
150
162
163
16
17
18
31
32
33
19
36
37
38
191
64
65
66
67
68
69
97
40
41
42
154
155
156
157
158
161
195
196
197
51
52


59
60
61
62
63
88
208
81
164
165
166
180
181
182
183
184
185
186
187
204
205
206
34
35
39
90
91
94
111
47
48
159
192
71
72
73
151
152
153
74
75
76
78
79
80
85
86
87
103
200
201
148
149
188
189
13
14


15
70
145
146
147
167
168
169
170
171
172
173
20
21
22
23
24
25
28
29
30
46
89
104
105
106
107
112
113
114
122
123
143
144
193
194
26
27
43
44
45
124
125
129
130
131
132
207
49
50
53
54
55
56
57
58
115



Gambar 6.3 Visualisasi *Poc-Tree* untuk *Minimum Support* = 5

Berdasarkan *Poc-Tree* pada Gambar 6.3 dihasilkan hasil variasi menu paket dengan jumlah 97 variasi. Data hasil menu paket dapat dilihat pada Lampiran A.6. Sedangkan *Poc-Tree* pada Gambar 6.2 dihasilkan hasil variasi menu paket dengan jumlah 6 variasi. Data hasil menu paket dapat dilihat pada Lampiran A.12. Berdasarkan hasil pembentukan menu paket diatas menunjukkan bahwa apabila ukuran *POC-tree* yang besar maka akan menghasilkan jumlah variasi menu paket yang lebih besar. Sedangkan apabila ukuran *Poc-Tree* yang kecil akan menghasilkan jumlah variasi menu paket yang lebih kecil. Dapat disimpulkan ukuran *Poc-Tree* berbanding lurus dengan jumlah variasi menu paket yang dihasilkan.

Untuk variasi menu paket pada *minimum support* = 11 telah menghasilkan jumlah proporsional menu paket yaitu 6 variasi menu paket. Variasi menu paket yang sudah merepresentasikan kebiasaan konsumen dalam memilih *item* menu makanan atau *item* menu minuman. Penulis dapat merekomendasikan kepada pihak rumah makan untuk menggunakan 6 variasi menu paket tersebut sebagai pilihan untuk membuat menu paket. Hasil rekomendasi menu paket oleh sistem masih dalam bentuk menu paket mentah yang perlu untuk dipilih kembali. Hasil dari 6 variasi menu paket dapat dilihat pada Lampiran A.12 . Pada Tabel 6.2 menampilkan hasil *mining frequent 3-itemset* dengan nilai *threshold minimum support* = 11. Kolom frekuensi *item* menunjukkan frekuensi *item* penyusun menu

paket d*item*ukan pada 100 data transaksi. Kolom nilai *support item* penyusun menunjukkan nilai *support* tiap *item* penyusun menu paket. Untuk nilai *support* tiap penyusun menu paket memiliki nilai lebih dari sama dengan nilai minimum *support* yaitu 0.11. 0.11 didapatkan dari nilai *minimum support* masukan user 11 dibagi dengan banyaknya transaksi yaitu 100. Variasi menu paket pada nomor 1, 2, 3 menunjukkan bahwa variasi menu paket masih mentah. Disebabkan variasi menu tersebut tidak terkombinasi dengan *item* menu minuman. Untuk variasi menu paket pada nomor 4, 5, 6 sudah menunjukkan variasi menu paket yang lengkap terdiri dari *item* menu makanan pokok, *item* minuman, dan *item* menu makanan lauk pauk.

Hasil variasi menu paket pada Tabel 6.2 tidak dapat dipaksakan harus tersusun dari kombinasi *item* makanan pokok, *item* minuman, dan *item* makanan lauk pauk. Disebabkan menyesuikan dengan teorema algoritma *FIN* pada proses pembentukan menu paket melalui *mining frequent 3-itemset*. Untuk menghasilkan *frequent 3-itemset* tersebut terdapat proses menemukan irisan *id Node* pada dua frequent *2-itemset* yang dihasilkan dari proses mining *frequent 2-itemset* pada *POC-tree*. Menu paket yang dihasilkan sangat tergantung terhadap bentuk *POC-tree* yang terbentuk. Bentuk *Poc-Tree* sangat tergantung pada *item-item* yang menyusun suatu transaksi serta *eliminator item-item* pada suatu transaksi yaitu nilai *minimum support*.

Tabel 6.2 Hasil Menu Paket dengan *minimum support* = 11

No	Kombinasi Menu Paket	Nilai *Support Item* Penyusun	Frekuensi *Item*
1	Nasi Putih, Ayam Goreng / biji, Kentang Goreng	{0,61; 0,14; 0,11}	{61, 14, 11}
2	Nila Bakar / biji, Ayam Goreng / biji, Kentang Goreng	{0,14; 0,14; 0,11}	{14, 14, 11}
3	Nasi Putih, Jamur Goreng Tepung, Nila Bakar / biji	{0,61; 0,14; 0,14}	{61, 14, 14}
4	Nasi Putih, Nila Bakar / biji, Ice Lemon Tea	{0,61; 0,14; 0,11}	{61, 14, 11}
5	Nasi Putih, Nila Bakar / biji, Juice Alpukat	{0,61; 0,14; 0,12}	{61, 14, 12}
6	Nasi Putih, Nila Bakar / biji, Kentang Goreng	{0,61; 0,14; 0,11}	{61, 14, 11}

Untuk hasil *mining frequent 3-itemset* untuk *minimal support* = 10, menghasilkan variasi menu paket yang lebih banyak yaitu 24 variasi menu paket. Apabila hasil rekomendasi oleh sistem pada *minimum support* = 11, memiliki jumlah variasi menu paket yang kurang variatif maka, penulis dapat merekomendasikan hasil minimum support = 10 kepada pihak rumah makan untuk dijadikan pilihan untuk membuat menu paket. Hasil 24 variasi menu paket dapat dilihat pada Lampiran A.11.

Alasan penulis memilih hasil mining minimum support = 11 sebagai pilihan pertama dan hasil mining minimum support = 10 sebagai pilihan kedua adalah nilai *minimum support* tertinggi terdapat pada nilai *minimum support* 10, 11, dan 12. Hasil pembentukan menu paket ditentukan oleh nilai *minimum support*. Nilai

minimum support sebagai *eliminator item-item* yang tidak *infrequent* pada setiap data transaksi. Apabila dipilih *minimum support* dengan nilai tinggi maka mengeliminasi banyak *item infrequent* pada data transaksi. Sehingga menghasilkan menu paket yang representatif dengan kebiasaan konsumen dalam memilih menu. Hasil *mining* pada *minimum support* pada nilai *minimum support* tinggi ini yang terbaik dari pada hasil *mining* pada nilai *minimum support* yang lebih rendah. Disebabkan pada hasil *mining* nilai *minimum support* yang lebih rendah masih menghasilkan *item infrequent* yang menyusun menu paket. Sedangkan hasil *mining minimum support* = 12 tidak dipilih oleh penulis walaupun memiliki nilai *minimum support* tertinggi disebabkan karena hanya menghasilkan 1 variasi menu paket. Sehingga tidak variatif apabila direkomendasikan sebagai menu paket.

Pada Tabel 6.3 menampilkan waktu eksekusi sistem berdasarkan nilai *minimum support* yang berbeda – beda. Untuk melakukan *mining frequent itemset* dengan 100 data transaksi. Waktu eksekusi tercepat terdapat pada *minimum support* = 11. Sedangkan waktu eksekusi terlambat terdapat pada *minimum support* = 0. Rata-rata waktu eksekusi sistem adalah 1,32 detik untuk *minimum support* 0 sampai 12. Pada pengujian penelitian sebelumnya, algoritma *FIN* digunakan untuk *mining frequent itemset* pada 8.124 transaksi. Rata-rata waktu eksekusi pengujian tersebut adalah 0,067 detik (Deng, 2014). Pengujian penelitian sebelumnya menggunakan perangkat keras dengan memori 16 Gb dan *prosessor Intel Xeon* 2.0 GHz. Penelitian yang dilakukan penulis menunjukkan performa algoritma *FIN*, yang diimplementasikan pada sistem pembuat menu paket otomatis, memiliki selisih waktu eksekusi yang kecil dibandingkan dengan penelitian yang dilakukan oleh peneliti sebelumnya.

Tabel 6.3 Hasil waktu eksekusi sistem berdasarkan *minimum support*

No	*Minimum Support*	Waktu Eksekusi
1	0	2,11 detik
2	1	2,30 detik
3	2	1,65 detik
4	3	1,56 detik
5	4	1,91 detik
6	5	1,42 detik
7	6	1,38 detik
8	7	1,24 detik
9	8	1,13 detik
10	9	0,77 detik
11	10	0,80 detik
12	11	0,41 detik
13	12	0,55 detik

Penelitian oleh peneliti sebelumnya menggunakan algoritma *dEclat* untuk melakukan *mining frequent itemset* menggunakan 49.096 transaksi. Rata-rata waktu eksekusi adalah 319, 5 detik (Wang, 2012). Menggunakan perangkat keras dengan memori 2 Gb dan *prosesor Intel Core 2 Duo* 2,66 Ghz. Algoritma *dEclat* adalah algoritma *frequent itemset* yang berdasar pada algoritma *Apriori* (Wang, 2012).

Berdasarkan pengujian yang dilakukan oleh penulis maupun pengujian yang dilakukan oleh peneliti-peneliti sebelumnya, menunjukkan algoritma *FIN* memiliki performa yang lebih baik dibandingkan dengan algoritma yang berdasarkan algoritma *Apriori* seperti algoritma *dEclat*. Algoritma *FIN* memiliki waktu eksekusi yang lebih cepat dibandingkan algoritma *dEclat*. Algorima *dEclat* memiliki efisiensi yang lebih rendah dibandingkan algorima *FIN* disebabkan terdapat proses pembentukan kandidat itemset (Han, 2007). Sedangakan algoritma *FIN* tidak terdapat proses pembentukan kandidat itemset. Pada algoritma *dEclat* dibutuhkan sumber daya yang besar disebabkan harus berulang-ulang melakukan pemindaian terhadap *database* dan harus mencocokan *itemset* pada *dataset* kandidat yang berukuran besar (Han, 2007). Algoritma *FIN* melakukan pendekatan yang lebih efisien dalam melakukan *mining frequent itemset*. Tanpa proses pembentukan kandidat *itemset*. Algoritma *FIN* mengganti proses tersebut dengan menggunakan struktur data terkondensasi tinggi yaitu *POC-tree* untuk menyimpan *database*. Serta *nodeset* digunakan untuk merepresentasikan *itemset*. Algoritma *FIN* melakukan teknik pendekatan *divide and conquer* untuk melakukan mining frequent itemset (Deng, 2014).

6.3 Pengujian Akurasi Menu Paket

Menu paket lengkap terdiri dari item makanan pokok, makanan lauk pauk, dan minuman. Sedangkan menu paket tidak lengkap adalah menu paket yang tidak memiliki salah satu komponen menu paket lengkap. Pada Tabel 6.3 ditampilkan hasil pengujian akurasi menu paket yang dihasilkan oleh sistem.

Tabel 6.4 Hasil pengujian akurasi menu paket

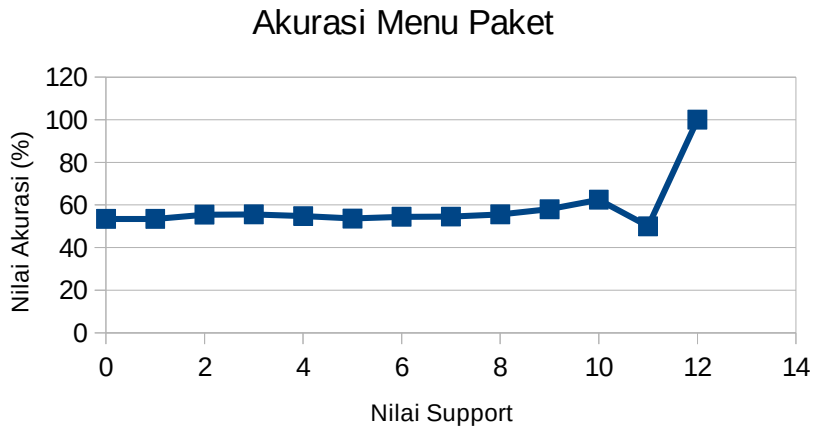
Minimal Support	**Menu Paket Lengkap (%)**	**Menu Paket Tidak Lengkap (%)**
0	53,49	46,51
1	53,49	46,51
2	55,47	44,53
3	55,56	44,44
4	54,78	45,22
5	53,61	46,39
6	54,43	45,57
7	54,55	45,45
8	55,56	44,44
9	58	42

Tabel 6.4 Hasil pengujian akurasi menu paket (lanjutan)

Minimal Support	Menu Paket Lengkap (%)	Menu Paket Tidak Lengkap (%)
10	62,5	37,5
11	50	50
12	100	0

Rata-rata untuk akurasi menu lengkap adalah 63,45 %. Sedangkan rata-rata untuk akurasi menu paket tidak lengkap adalah 44,88 %. Menu paket yang dihasilkan oleh sistem memiliki nilai akurasi dengan nilai 63,45 %, berdasarkan pada rata-rata nilai akurasi menu paket lengkap. Menu paket lengkap dapat digunakan sebagai menu paket oleh pihak rumah makan. Sedangkan menu paket tidak lengkap kurang tepat apabila dijadikan sebagai menu paket.

6.4 Analisis Hasil Akurasi Menu Paket



Gambar 6.4 Grafik hasil akurasi menu paket

Berdasarkan hasil pengujian pada Tabel 6.3, sistem telah menghasilkan menu paket secara lengkap pada nilai presentase 63,45 %. Sedangkan sistem masih membuat kesalahan dalam menghasilkan menu paket secara lengkap pada nilai presentase 44,88 %. Apabila sistem diimplementasikan oleh pihak rumah makan pada proses bisnisnya maka pihak rumah makan dapat mempercayai hasil rekomendasi oleh sistem sebesar 63,45 %. Pada sistem belum terdapat mekanisme memilih menu paket lengkap secara otomatis. Sehingga pihak rumah makan masih menggunakan cara manual yaitu melihat satu per satu hasil variasi menu paket yang dihasilkan oleh sistem. Untuk mendapatkan menu paket lengkap yang dapat digunakan untuk menyusun menu paket rumah makan.