

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Penelitian ini menghasilkan penggunaan algoritma apriori dalam aplikasi Point of Sale (POS) berbasis Laravel untuk menganalisis pola pembelian pada UMKM SehaTea untuk menemukan pola hubungan antar item.

4.1.1 Pembahasan Algoritma Apriori

UMKM SehaTea menganalisis pola pembelian pelanggan berdasarkan data transaksi. Tujuan dari analisis ini adalah untuk menemukan hubungan antar produk (frequent itemsets) menggunakan Algoritma Apriori.

Data transaksi melibatkan berbagai jenis minuman seperti Iced Tea Original, Iced Coklat Tea, Iced Green Tea, Iced Lemon Tea, dan Es Tawar. Data transaksi ini akan diolah untuk menemukan pola pembelian yang sering terjadi.

Dataset Transaksi

Dari data yang diberikan, kita mengonversinya menjadi format transaksi dengan produk yang dibeli. Setiap baris mewakili 1 transaksi:

Tabel 4.1 Dataset Transaksi

No	Transaksi	Produk Dibeli
1	T539	Iced Tea Original
2	T540	Iced Coklat Tea
3	T541	Iced Tea Original
4	T542	Es Tawar
5	T543	Iced Tea Original
6	T544	Iced Green Tea
7	T545	Iced Tea Original
8	T546	Iced Coklat Tea
9	T547	Iced Tea Original
10	T548	Iced Green Tea

11	T549	Iced Tea Original
12	T550	Iced Coklat Tea
13	T551	Iced Coklat Tea
14	T552	Iced Tea Original
15	T553	Iced Green Tea
16	T554	Iced Tea Original
17	T555	Iced Lemon Tea
18	T556	Iced Tea Original

Menentukan Support untuk Item Tunggal

Support dihitung sebagai jumlah transaksi yang mengandung item tersebut, dibagi dengan total transaksi (**18 transaksi**).

Tabel 4.2 Suport item tunggal

Item	Jumlah Transaksi	Support
Iced Tea Original	10	$10/18=0.556$ atau 55.6%
Iced Coklat Tea	5	$5/18=0.278$ atau 27.8%
Es Tawar	1	$1/18=0.056$ atau 5.6%
Iced Green Tea	3	$3/18=0.167$ atau 16.7%
Iced Lemon Tea	1	$1/18=0.056$ atau 5.6%

Misalkan minimum support adalah **20%**. Maka, item yang memenuhi threshold support adalah:

1. Iced Tea Original (55.6%)
2. Iced Coklat Tea (27.8%)

Membentuk Pasangan Itemset 2 dan Menghitung Support

Kombinasi dari item yang memenuhi support threshold:

Tabel 4.3 Itemset 2

Itemset	Jumlah Transaksi yang Mengandung Itemset	Support
(Iced Tea Original, Iced Coklat Tea)	0	0/18=0.0 atau 0%

Karena tidak ada kombinasi dengan support \geq minimum support (20%), algoritma berhenti pada langkah ini.

1. **Support:** Mengukur seberapa sering item atau kombinasi item muncul dalam transaksi.
2. **Confidence:** Mengukur seberapa kuat hubungan antara *antecedent* dan *consequent*.

Hasil Akhir

Dari hasil perhitungan **algoritma Apriori**, pola pembelian yang sering muncul adalah:

1. **Iced Tea Original** dengan support **55.6%**
2. **Iced Coklat Tea** dengan support **27.8%**

Tidak ditemukan kombinasi itemset (frequent itemsets 2) yang memenuhi minimum support.

4.1.2 Implementasi Program Dan Database Pada UMKM Sehat

Berikut ini program untuk menentukan pola pembelian dengan menggunakan algoritma apriori. Program dalam gambar tersebut adalah implementasi fungsi analyze dalam bahasa pemrograman PHP. Fungsi ini dirancang untuk menganalisis data transaksi menggunakan algoritma *Apriori* untuk menghasilkan *association rules*.

```

1 public function analyze($startDate, $endDate)
2 {
3     try {
4         // Ambil data transaksi dalam periode
5         $transactions = DetailTransaksi::with(['transaksi', 'produk'])
6             ->whereHas('transaksi', function ($query) use ($startDate, $endDate) {
7                 $query->whereBetween('created_at', [$startDate, $endDate]);
8             })
9             ->get()
10            ->groupBy('transaksi.id')
11            ->map(function ($items) {
12                return $items->pluck('produk.nama')->toArray();
13            })->values()->toArray();
14
15        if (empty($transactions)) {
16            throw new \Exception("Tidak ada transaksi dalam periode yang dipilih");
17        }
18
19        logger()->info('Transactions:', ['data' => $transactions]);
20
21        $frequentItemSet = $this->findFrequentItemSet($transactions);
22
23        if (empty($frequentItemSet)) {
24            throw new \Exception("Tidak ditemukan item yang memenuhi minimum support");
25        }
26
27        $frequentItemSets = $this->generateFrequentItemSets($transactions, $frequentItemSet);
28        $rules = $this->generateAssociationRules($frequentItemSets, $transactions);
29
30        // Simpan Hasil Analisis
31        foreach ($rules as $rule) {
32            AnalisisApriori::create([
33                'itemset' => json_encode([
34                    'antecedent' => $rule['antecedent'],
35                    'consequent' => $rule['consequent']
36                ]),
37                'support' => $rule['support'],
38                'confidence' => $rule['confidence'],
39                'minimum transactions' => count($transactions),
40                'periode' => $endDate
41            ]);
42        }
43
44        return $rules;
45    } catch (\Exception $e) {
46        logger()->error('Apriori Analysis Error:', [
47            'message' => $e->getMessage(),
48            'trace' => $e->getTraceAsString()
49        ]);
50        throw $e;
51    }
52 }
53
54 private function findFrequentItemSet($transactions)
55 {
56     $items = [];
57     $totalTransactions = count($transactions);
58
59     logger()->info('Total Transactions:', ['count' => $totalTransactions]);
60
61     foreach ($transactions as $transaction) {
62         foreach ($transaction as $item) {
63             if (!isset($items[$item])) {
64                 $items[$item] = 1;
65             } else {
66                 $items[$item]++;
67             }
68         }
69     }
70
71     logger()->info('Item Frequencies:', ['data' => $items]);
72
73     $frequentItemSet = [];
74     foreach ($items as $item => $count) {
75         $support = $count / $totalTransactions;
76         logger()->info('Item Support:', [
77             'item' => $item,
78             'count' => $count,
79             'support' => $support,
80             'minSupport' => $this->minSupport
81         ]);
82
83         if ($support >= $this->minSupport) {
84             $frequentItemSet[$item] = $support;
85         }
86     }
87
88     logger()->info('Frequent 1-ItemSet:', ['data' => $frequentItemSet]);
89     return $frequentItemSet;
90 }

```

Gambar 4.1 Algoritma Apriori

Pada gambar 4.1 program startDate, tanggal mulai untuk memfilter data transaksi dan endDate, tanggal akhir untuk memfilter data transaksi.

1. Mengambil Data Transaksi, Data transaksi difilter berdasarkan kolom Tanggal mulai untuk memfilter data transaksi. Dan Tanggal akhir untuk memfilter data transaksi.
2. Proses Analisis Apriori, Fungsi memanggil data bases untuk menentukan *frequent itemsets* berdasarkan transaksi. *Rules* yang dihasilkan berupa informasi tentang hubungan antar item, termasuk nilai *support*, *confidence*, dan *consequent* dari setiap aturan.

Program pada gambar 4.1 dan gambar 4.2 ini merupakan implementasi dari algoritma *Apriori* untuk menganalisis data transaksi dan menemukan pola hubungan antar item

```

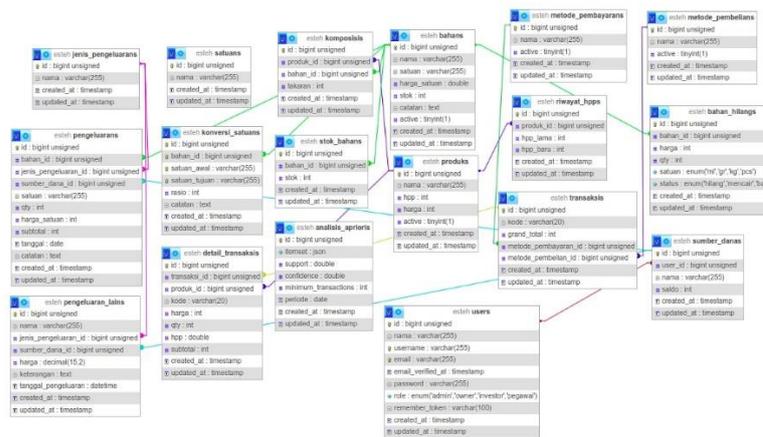
1 private function generateCandidateItemSets($frequentItemSets)
2 {
3     logger()->info('Generating candidates from:', ['itemsets' => $frequentItemSets]);
4     if (empty($frequentItemSets)) {
5         return [];
6     }
7     // Jika ini adalah frequent 1-itemset, ubah formatnya
8     if (!is_array(reset($frequentItemSets))) {
9         $frequentItemSets = array_map(function ($item) {
10             return [$item];
11         }, $frequentItemSets);
12     }
13     $candidates = [];
14     $k = count(reset($frequentItemSets)) + 1; // ukuran itemset berikutnya
15     foreach ($frequentItemSets as $i => $itemSet1) {
16         foreach ($frequentItemSets as $j => $itemSet2) {
17             if ($i < $j) {
18                 $candidate = array_unique(array_merge($itemSet1, $itemSet2));
19                 if (count($candidate) == $k) {
20                     sort($candidate); // Urutkan items untuk konsistensi
21                     $candidates[] = $candidate;
22                 }
23             }
24         }
25     }
26     logger()->info('Generated candidates:', ['candidates' => $candidates]);
27     return $candidates;
28 }
29
30 private function generateFrequentItemSets($transactions, $frequentItemSet)
31 {
32     logger()->info('Starting generateFrequentItemSets', [
33         'frequentItemSet' => $frequentItemSet
34     ]);
35     $allFrequentItemSets = [$frequentItemSet];
36     $k = 1;
37     $currentFrequentSet = array_keys($frequentItemSet);
38     $totalTransactions = count($transactions);
39     while (!empty($currentFrequentSet)) {
40         logger()->info('Processing k=' . $k . ' itemsets');
41         // Generate kandidat
42         $candidates = $this->generateCandidateItemSets($currentFrequentSet);
43         if (empty($candidates)) {
44             break;
45         }
46         // Hitung support untuk setiap kandidat
47         $frequentItemSetsWithSupport = [];
48         foreach ($candidates as $candidate) {
49             $count = 0;
50             foreach ($transactions as $transaction) {
51                 if (count(array_intersect($candidate, $transaction)) == count($candidate)) {
52                     $count++;
53                 }
54             }
55             $support = $count / $totalTransactions;
56             logger()->info('Candidate support:', [
57                 'candidate' => $candidate,
58                 'support' => $support,
59                 'minSupport' => $this->minSupport
60             ]);
61             if ($support >= $this->minSupport) {
62                 $frequentItemSets[] = $candidate;
63                 $frequentItemSetsWithSupport[implode('.', $candidate)] = $support;
64             }
65         }
66         if (empty($frequentItemSets)) {
67             $allFrequentItemSets[] = $frequentItemSetsWithSupport;
68             $currentFrequentSet = $frequentItemSets;
69         } else {
70             break;
71         }
72         $k++;
73     }
74     logger()->info('Final frequent itemsets:', ['data' => $allFrequentItemSets]);
75     return $allFrequentItemSets;
76 }

```

Gambar 4.2 Algoritma Apriori Lanjutan

Tampilan Halaman Database

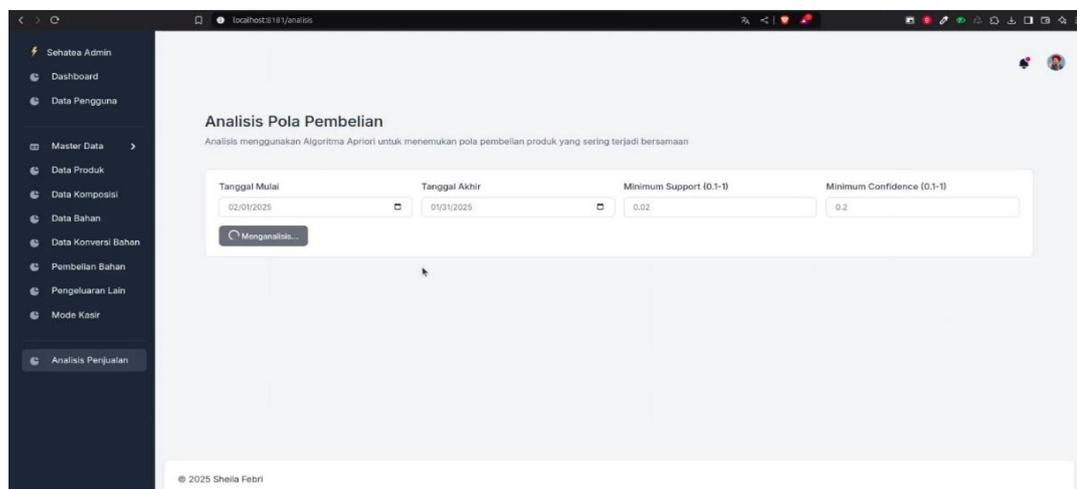
Gambar 4.3 merupakan database dari transaksi, bahan-bahan, metode pembayaran, riwayat, stok bahan pengeluaran dan database user yang di gunakan untuk mengelompokkan dan melakukan analisis yang dilakukan dengan menggunakan algoritma apriori untuk untuk menganalisis data transaksi dan menemukan pola hubungan antar item



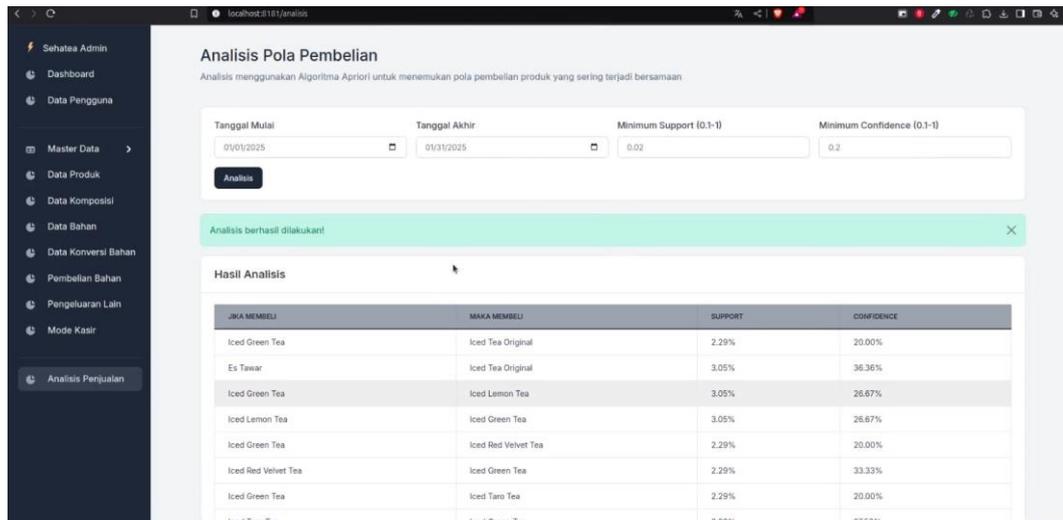
Gambar 4.3 Database

Tampilan Halaman Analisis Pola Pembelian Dengan Algoritma Apriori

Di dalam website umkm sehattea pada menu analisis pola pembelian admin perlu memasukkan tanggal mulai dan tanggal akhir untuk menentukan pola pembelian, Setelah memasukkan tanggal mulai dan tanggal akhir, selanjutnya Sedang menganalisis pola pembelian.



Gambar 4.4 Analisis Pola Pembelian



Gambar 4.5 Hasil Analisis Pola Pembelian

Ketika hasil analisis sudah selesai dilakukan maka, akan muncul hasil analisisnya yang berisikan;

1. Jika pengunjung membeli Iced Green Tea maka pembeli akan membeli Iced Tea Original dengan presentase confidence sebesar 20.00% dan support 2.29%
2. Jika pengunjung membeli Es Tawar maka pembeli akan membeli Iced Tea Original dengan presentase confidence sebesar 36.36% dan support 3.05%
3. Jika pengunjung membeli Iced Green Tea maka pembeli akan membeli Iced Lemon Tea dengan presentase confidence sebesar 26.67% dan support 3.05%
4. Jika pengunjung membeli Iced Lemon Tea maka pembeli akan membeli Iced Green Tea dengan presentase confidence sebesar 26.67% dan support 3.05%
5. Jika pengunjung membeli Iced Green Tea maka pembeli akan membeli Iced Red Velvet Tea dengan presentase confidence sebesar 20.00% dan support 2.29%
6. Jika pengunjung membeli Iced Red Velvet Tea maka pembeli akan membeli Iced Tea Original dengan presentase confidence sebesar 33.33% dan support 2.29%

7. Jika pengunjung membeli Iced Green Tea maka pembeli akan membeli Iced Taro Tea dengan presentase confidence sebesar 20.00% dan support 2.29%
8. Jika pengunjung membeli Iced Taro Tea maka pembeli akan membeli Iced Green Tea dengan presentase confidence sebesar 37.50% dan support 2.29%

4.1.3 Hasil Implementasi Algoritma Apriori Pada UMKM Seatea

Implementasi dari **algoritma Apriori** dalam aplikasi **Point of Sale (POS)** berbasis Laravel untuk menganalisis pola pembelian pada UMKM SehaTea menemukan pola hubungan antar item pada tabel

Tabel 4.4 Hasil Implementasi Algoritma Apriori

No	Jika Membeli	Maka Membeli	Support	Confidence
1	Iced Green Tea	Iced Tea Original	2.29%	20.00%
2	Es Tawar	Iced Tea Original	3.05%	36.36%
3	Iced Green Tea	Iced Lemon Tea	3.05%	26.67%
4	Iced Lemon Tea	Iced Green Tea	3.05%	26.67%
5	Iced Green Tea	Iced Red Velvet Tea	2.29%	20.00%
6	Iced Red Velvet Tea	Iced Green Tea	2.29%	33.33%
7	Iced Green Tea	Iced Taro Tea	2.29%	20.00%
8	Iced Taro Tea	Iced Green Tea	2.29%	37.50%

4.2 Testing (Pengujian)

kegiatan yang dilakukan pada tahap ini yaitu untuk meminimalisir kesalahan dan memastikan output yang dihasilkan sesuai dengan yang diinginkan. Berdasarkan hasil pengujian sistem output dari implementasi algoritma apriori dalam aplikasi pos berbasis laravel untuk analisis pola pembelian umkm seatea sukses dan semua tombol aktivitas berjalan sesuai kebutuhan.

Tabel 4.5 Pengujian *Blackbox* Pada Website Seatea

No	Test Case	Output	Hasil
1	Halaman Login	Tampil halaman login	Sukses
	1. Memasukkan username dan password	- jika login berhasil maka akan tampil notifikasi berhasil dan akan masuk kehalaman dashboard - jika login gagal maka akan muncul notifikasi gagal dan akan Kembali ke halaman login	Sukses
2	Halaman Dashboard	Tampil halaman dashboard	sukses
	3. Terdapat periode data 4. Transaksi bulan ini 5. Nilai transaksi 6. Pengeluaran toko 7. Pengeluaran lain-lain	Menghitung jumlah dari transaksi, nilai transaksi, pengeluaran toko dan tanggalnya	Sukses
3	Halaman Analisis Penjualan	Tampil hasil analisa dari pola pembelian	sukses
4	Halaman Data Barang	Tampil halaman data barang	Sukses
5	Halaman Data komposisi	Tampil halaman data komposisi	Sukses
	1. Halaman bagian tambah komposisi baru	Pada bagian produk dapat memilih nama produknya, bahan-bahan, takaran dan satuanya	Sukses
6	Halaman Data Konversi Satuan	tampil hasil dari satuan bahan, satuan awal, dan satuan akhir	Sukses
7	Halaman Pembelian Bahan	Tampil halaman pembelian bahan	Sukses
	1. Terdapat bahan 2. Jenis pengeluaran 3. Sumber dana 4. Satuan bahan 5. Qty bahan 6. Harga beli 7. Subtotal 8. tanggal	Menghitung jumlah bahan, jenis pengeluaran, sumber dana, satuan bahan, qty bahan, harga beli, subtotal, dan tanggalnya	Sukses
8	Halaman Data Pegeluaran Lain Lain	Tampil halaman data pengeluaran lain-lain	Sukses
9	Halaman data pengguna	Tampil halaman data pengguna	Sukses
	1. Nama lengkap 2. Email 3. Password	Menampilkan nama pengguna, email, dan password.	Sukses
10	Halaman Data Produk	Tampil halaman data produk	Sukses
11	Halaman Kasir Penjualan	Tampil halaman kasir penjualan	Sukses
12	Halaman Data Jenis Pengeluaran	Tampil halaman data jenis pengeluaran	Sukses
13	Halaman Data Sumber Dana	Tampil halaman data sumber dana	Sukses