BAB V
SIMPULAN DAN SARAN

5.1 Kesimpulan

1. Menghindari Menu Fisik

Dengan sistem menu digital, restoran dapat menghindari kekurangan menu fisik pada saat banyak tamu datang sekaligus, sehingga pemesanan tetap berjalan lancar dan efisien.

2. Menonaktifkan Menu

Sistem ini menguntungkan karena menu yang tidak tersedia secara otomatis dinonaktifkan pada menu digital, tanpa perlu adanya komunikasi manual antara *waitress* dan pelanggan, yang mempercepat proses pemesanan dan mengurangi kebingungan pelanggan.

3. Mendorong Penjualan

Dengan sistem yang mendorong penjualan menu tambahan atau pendamping, restoran dapat menjual menu *best seller* lain yang dapat menaikkan omset penjualan.

4. Stock Bahan Baku

Hal ini sangat memudahkan restoran dalam membeli bahan baku karena restoran akan berfokus pada bahan yang cepat habis dan lebih berani untuk stok bahan baku dalam jumlah besar.

5. Penerapan metode apriori dan pengujian SUS

Sistem yang telah dibuat menghasilkan respon yang positif dari para pihak restoran yang diwawancarai, sistem apriori yang digunakan pada pemesanan menu secara digital dapat berdampak pada penjualan. Selain itu, pada pengujian SUS skor rata-rata yang didapat adalah 72,4. Dari hasil tersebut, maka sistem pemesanan makanan dan minuman mendapat *grade* C+, peringkat persentilnya berada pada kisaran 63,8%, masuk dalam kategori *Good* untuk sifatnya (*Adjective*) dan masuk dalam kategori marginal untuk tingkat penerimaannya (*Acceptable*) dan untuk nilai NPS-nya sediri tergolong *passive*.

5.2 Saran

Berdasarkan penelitian ini, maka saran yang dapat diberikan adalah sebagai berikut:

1. **Pengembangan fitur ulasan menu**

   Sistem perlu mengembangkan fitur ulasan menu supaya pelanggan dapat lebih yakin dalam memilih menu, karena terdapat ulasan yang memungkinkan pelanggan untuk mempertimbangkan menu tersebut.

2. **Pengembangan fitur pemilihan 2 menu terlaris**

   Sistem perlu mengembangkan fitur pencarian menu terlaris dengan 2 itemset atau menu A dan menu B, sehingga proses perhitungan apriori dapat dilakukan secara otomatis mengambil data dari penjualan secara langsung tanpa perlu menghitung apriori secara manual.

DAFTAR PUSTAKA

Adensa … Azizah. (2024). PENGEMBANGAN WEB DINAS PERPUSTAKAAN DAN ARSIP BERBASIS LARAVEL FRAMEWORK PADA DPAD KOTA TANGERANG. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(6). https://doi.org/10.36040/jati.v7i6.7840

Adler, & Dika. (2022). Sistem Informasi Pemesanan Menu Makanan dan Minuman Berbasis Web Sebagai Penentu Nilai Menu Terbaik. *Majalah Ilmiah UNIKOM*, *20*(1), 33–43. https://doi.org/10.34010/miu.v20i1.7712

Aldisa … Furqon. (2022). Penerapan Metode Joint Application Design (JAD) dalam Pengembangan Sistem Informasi Penjualan Jaket Hoodie Berbasis *Website*. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *6*(1). https://doi.org/10.30865/mib.v6i1.3438

Anjali Prabhu, Anvitha H M, Bhavya S. (2022). E-Food Ordering System for AIET. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, *06*(03). https://doi.org/10.55041/ijsrem11970

Apriliyanti … Alam. (2023). Perancangan UI/UX Aplikasi Media Pembelajaran Pengenalan Budaya Indonesia Untuk Siswa Sekolah Dasar Menggunakan Metode *User* Centered Design (UCD). *Jurnal Teknologi Informatika Dan Komputer*, *9*(2). https://doi.org/10.37012/jtik.v9i2.1665

Bist, & Prambudi. (2021). Implementation Of Data Mining On Glasses Sales Using The Apriori Algorithm. *International Journal of Cyber and IT Service Management (IJCITSM)*, *1*(2), 159–172. https://doi.org/10.34306/ijcitsm.v1i1.46

Daryanti … Astuti. (2022). Prediksi Harga Cabai Menggunakan Fuzzy Time Series Model Chen. *Jurnal Rekayasa Teknologi Dan Komputasi*, *1*(2).

Dirgahinta, & Anwar. (2018). Aplikasi E-Commerce Penjualan Sepatu Dengan Metode Cross Selling Pada Toko Pantes. *Prosiding SINTAK 2018*, 164–170.

Handayani … Taufiq. (2020). Rancang Bangun Sistem Informasi Pemesanan Menu Makanan Berbasis Web (Studi Kasus: Restoran Bukit Randu Bandara). *Jurnal SITECH : Sistem Informasi Dan Teknologi*, *3*(1), 21–28.

https://doi.org/10.24176/sitech.v3i1.4837

Juniar, & Daniawan. (2024). Optimasi Sistem Informasi Pembelian, Persediaan, dan Penjualan Barang dengan Penerapan Metode Algoritma Apriori. *Jurnal Teknik Informatika Dan Sistem Informasi*, *10*(1). https://doi.org/10.28932/jutisi.v10i1.7647

Liansyah, & Destiana. (2020). The Use of Apriori Algorithm in the Formation of Association Rule at Lotteria Cibubur. *SinkrOn*, *4*(2), 76. https://doi.org/10.33395/sinkron.v4i2.10526

Marbun. (2021). PERANCANGAN SISTEM INFORMASI PEMESANAN MENU MAKANAN DI TWIN SEAFOOD & RESTORAN RANTAUPRAPAT BERBASIS WEB. *INFORMATIKA*, *9*(2), 71–76. https://doi.org/10.36987/informatika.v9i2.1950

Muhammad Jibril … Zulrahmadi. (2023). SISTEM INFORMASI PEMESANAN PADA WARKOP PAK DE BERBASIS WEB. *JURNAL PERANGKAT LUNAK*, *5*(2), 86–96. https://doi.org/10.32520/jupel.v5i2.2566

Nugraha, & Abdussallam. (2022). DESIGN OF THE POPULATION INFORMATION SYSTEM IN THE VILLAGE OF PAJAJARAN. *Journal of Applied Engineering and Technological Science*, *4*(1). https://doi.org/10.37385/jaets.v4i1.1012

Oktavia … Ong. (2023). DIGITAL MENU TRANSFORMATION: USABILITY TESTING APPROACH FOR THE FOOD AND BEVERAGE INDUSTRY'S. *Journal of Theoretical and Applied Information Technology*, *101*(10).

Pratama, & Dewi. (2025). Analysis of Consumer Purchasing Patterns Using the Apriori Algorithm on Sales Transaction Data from Anak Panah Kopi Salatiga. *International Journal Software Engineering and Computer Science (IJSECS)*, *5*(1), 1–10. https://doi.org/10.35870/ijsecs.v5i1.3275

Qamal. (2021). SISTEM INFORMASI WARUNG MAKAN SATE APALEH KECAMATAN GANDAPURA KABUPATEN BIREUEN BERBASIS WEB. *Jurnal Teknologi Terapan and Sains 4.0*, *1*(3), 289. https://doi.org/10.29103/tts.v1i3.3269

Reddy, & Kishore. (2024). Network For Ordering Food Online. *International Scientific*

Refereed Research Journal Available Online. https://doi.org/10.32628/SHISRRJ

Santoso. (2021). Application of Association Rule Method Using Apriori Algorithm to Find Sales Patterns Case Study of Indomaret Tanjung Anom. *Brilliance: Research of Artificial Intelligence*, *1*(2), 54–66. https://doi.org/10.47709/brilliance.v1i2.1228

Saputri … Kota. (n.d.). *RANCANG BANGUN SISTEM INFORMASI PEMESANAN MAKANAN BERBASIS WEB PADA CAFE SURABIKU*.

Saputri … Suherman. (2019). RANCANG BANGUN SISTEM INFORMASI PEMESANAN MAKANAN BERBASIS WEB PADA CAFE SURABIKU. *Jurnal Teknologi Dan Informasi*, *9*(1), 66–77. https://doi.org/10.34010/jati.v9i1.1378

Septiani … Astuti. (2023). Implementasi Metode Pieces Untuk Menganalisis Tingkat Kepuasan Pengguna Aplikasi Peduli Lindungi. *JIKI (Jurnal Ilmu Komputer Dan Informatika)*, *4*(1).

Zahara, & Nunsina. (2022). Perancangan Aplikasi Pemesanan Makanan Online Berbasis Web (E-del). *DEVICE : JOURNAL OF INFORMATION SYSTEM, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY*, *3*(2), 1–8. https://doi.org/10.46576/device.v3i2.2695

DAFTAR RIWAYAT HIDUP

Data Pribadi

NIM	: 20200700049
Nama	: Patricia Fransiska
Tempat Tanggal Lahir	: Tangerang, 13 Maret 2002
Alamat	: Griya Sangiang Mas Jl Elang 1, Tangerang

Pendidikan Formal

2008-2014	: SD Strada st. Aloysius 1
2014-2017	: SMP Penerus Bangsa
2017-2020	: SMK Maria Mediatrix
2020-2025	: Universitas Buddhi Dharma

Pengalaman Kerja

2020-2024	: Admin Finance, PT Tanindo Sejahtera Mandiri
2024-sekarang	: Manager, Bum Kitchen

Tangerang, 05-02-2025

Patricia Fransiska

20200700049

LAMPIRAN

Lampiran A- 1 Program (Coding)

Lampiran Coding Apriori

```
<?php

namespace App\Http\Controllers;

use App\Models\Customer;
use App\Models\Order;
use App\Models\OrderDetail;
use App\Models\Product;
use Exception;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;

class OrderController extends Controller
{
    protected $order;
    protected $product;
    protected $orderDetail;
    protected $customer;

    public function __construct()
    {
        $this->order = new Order();
        $this->product = new Product();
        $this->orderDetail = new OrderDetail();
        $this->customer = new Customer();
    }
```

```
public function getOrders(Request $request)
{
    try {
        $search = $request->input('search', '');
        $perPage = $request->input('perPage', 10);
        $page = $request->input('page', 1);

        $getOrders = $this->order->getOrders($search, $perPage, $page);

        return response()->json([
            'success'   => true,
            'orders'    => $getOrders,
        ]);
    } catch (Exception $error) {
        return response()->json([
            'success'   => false,
            'message'   => $error->getMessage()
        ], 500);
    }
}

public function getOrdersData()
{
    try {
        $getOrders = $this->order->getOrdersData();

        return response()->json([
            'success'   => true,
            'orders'    => $getOrders,
        ]);
    } catch (Exception $error) {
        return response()->json([
            'success'   => false,
            'message'   => $error->getMessage()
```

```
        ], 500);
    }
}

public function getOrderById($orderId)
{
    try {
        $getOrders = $this->order->where('id', $orderId)->first();

        return response()->json([
            'success'   => true,
            'orders'    => $getOrders,
        ]);
    } catch (Exception $error) {
        return response()->json([
            'success'   => false,
            'message'   => $error->getMessage()
        ], 500);
    }
}

public function getOrderDetailById($orderId)
{
    try {
        $getOrderDetails = $this->orderDetail->getOrderDetailById($orderId);

        return response()->json([
            'success'   => true,
            'orders'    => $getOrderDetails,
        ]);
    } catch (Exception $error) {
        return response()->json([
            'success'   => false,
            'message'   => $error->getMessage()
```

```
        ], 500);
    }
}

public function storeOrder(Request $request)
{
    try {
        $validation = Validator::make($request->all(), [
            'order_date'    => 'required|date',
            'customer'      => 'required',
        ], [
            'order_date.required'   => 'Tanggal pesanan tidak boleh kosong.',
            'order_date.date'       => 'Format tanggal pesanan tidak sesuai.',
            'customer.required'     => 'Nama pelanggan tidak boleh kosong.'
        ]);

        if($validation->fails()) {
            return response()->json([
                'success'   => false,
                'message'   => 'Something went wrong with your input.',
                'errors'    => $validation->errors()
            ], 400);
        }

        $orderNo = 'ORDER#' . date('mdHis');

        $this->order->create([
            'order_date'    => $request->order_date,
            'order_no'      => $orderNo,
            'customer'      => $request->customer
        ]);

        return response()->json([
            'success'   => true,
```

```
            'message'   => 'Pesanan baru berhasil dibuat.'
        ]);
    } catch (Exception $error) {
        return response()->json([
            'success'   => false,
            'message'   => $error->getMessage()
        ], 500);
    }
}

public function updateOrder(Request $request, $orderId)
{
    try {
        $this->order->where('id', $orderId)->update([
            'status'    => $request->status
        ]);

        return response()->json([
            'success'   => true,
            'message'   => 'Pesanan telah selesai dibuat, mohon diantarkan ke pelanggan.'
        ]);
    } catch (Exception $error) {
        return response()->json([
            'success'   => false,
            'message'   => $error->getMessage()
        ], 500);
    }
}

// For customer endpoint API
public function addProductToOrder(Request $request)
{
    try {
        foreach ($request->order_detail as $item) {
```

```
            $getProducts = $this->product->where('id', $item['product_id'])->get();

            if ($getProducts->isEmpty() || $getProducts->first()->stock < $item['qty']) {
                return response()->json([
                    'success' => false,
                    'message' => 'Stok produk tidak cukup untuk item ' . $item['product_id']
                ], 400);
            }

            $getProduct = $getProducts->first();

            $getProduct->stock -= $item['qty'];
            $getProduct->save();

            $this->orderDetail->create([
                'order_id'      => $request->order_id,
                'product_id'    => $item['product_id'],
                'qty'           => $item['qty'],
                'price'         => $item['price'],
            ]);
        }

        return response()->json([
            'success'   => true,
            'message'   => 'Pesanan mu berhasil berhasil ditambahkan.'
        ]);
    } catch (Exception $error) {
        return response()->json([
            'success'   => false,
            'message'   => $error->getMessage()
        ], 500);
    }
}
```

```
public function updateOrderData(Request $request, $orderId)
{
    try {
        $validation = Validator::make($request->all(), [
            'person'    => 'required|numeric',
            'table_no'  => 'required|numeric',
            'phone'     => 'required|numeric|digits_between:10,13'
        ], [
            'person.required'   => 'Jumlah orang tidak boleh kosong.',
            'person.numeric'    => 'Jumlah orang hanya dapat diisi angka.',
            'table_no.required' => 'Nomor meja tidak boleh kosong.',
            'table_no.numeric'  => 'Nomor meja hanya dapat diisi angka.',
            'phone.required'    => 'Nomor handphone tidak boleh kosong.',
        ]);

        if($validation->fails()) {
            return response()->json([
                'success'   => false,
                'message'   => 'Something went wrong with your input.',
                'errors'    => $validation->errors()
            ], 400);
        }

        $getOrderById = $this->order->select('customer')->where('id', $orderId)->first();
        $checkIfCustomerExist = $this->customer->where('phone', $request->phone)-
>first();

        if(empty($checkIfCustomerExist)) {
            $this->customer->create([
                'customer_name' => $getOrderById->customer,
                'phone'         => $request->phone,
                'address'       => null
            ]);
        }
```

```
        $this->order->where('id', $orderId)->update([
            'person'    => $request->person,
            'table_no'  => $request->table_no,
        ]);

        return response()->json([
            'success'   => true,
            'message'   => 'Berhasil, silahkan melakukan pesanan.'
        ]);
    } catch (Exception $error) {
        return response()->json([
            'success'   => false,
            'message'   => $error->getMessage()
        ], 500);
    }
}

public function checkOrderStatus($orderId)
{
    try {
        $getOrderById = $this->order->select('id', 'order_no', 'status')
                                    ->where('id', $orderId)
                                    ->first();

        if(empty($getOrderById)) {
            return response()->json([
                'success'   => false,
            ], 404);
        } else {
            if($getOrderById->status === 'served') {
                return response()->json([
                    'success'   => false,
                ], 400);
```

```
            }
        }

        return response()->json([
            'success'   => true,
        ]);
    } catch (Exception $error) {
        return response()->json([
            'success'   => false,
            'message'   => $error->getMessage()
        ], 500);
    }
}

public function showRecommendation(Request $request)
{
   try {
    $productId = $request->query('productId');
    $thresholdSupport = 0.10;
    $thresholdConfidence = 0.70;

    $transactions = $this->order->getTransactions();

    if ($transactions->isEmpty()) {
        return response()->json([
            'success' => false,
            'message' =>
            'No transactions found.'
        ], 400);
    }

    // preparing for item set 1
    $itemSet1 = $transactions->pluck('product_id')->toArray();
```

```
        // counting support for item set 1
        $support1 = $this->calculateSupport($itemSet1, $transactions, $thresholdSupport);

        // filtering item with support below threshold
        $filteredItemSet1 = array_filter($support1, fn($support) => $support >=
$thresholdSupport);

        if (count($filteredItemSet1) < 2) {
            return response()->json([
                'success' => false,
                'message' => 'Not enough item pairs for recommendation.'
            ], 400);
        }

        // make item pairs (item set 2)
        $itemSet2 = $this->generateItemPairs(array_keys($filteredItemSet1));

        // count support for item set 2
        $support2 = $this->calculateSupport($itemSet2, $transactions, $thresholdSupport);

        // filter item set 2 based on support
        $filteredItemSet2 = array_filter($support2, function ($support) use
($thresholdSupport) {
            return $support >= $thresholdSupport;
        });

        // make sure filteredItemSet2 contains pairs product, instead string
        $filteredItemSet2 = $this->formatItemPairs($filteredItemSet2);

        // counting confidence for item set 2
        $confidence2 = $this->calculateConfidence($filteredItemSet2, $transactions);

        // filtering based on confidence threshold
```

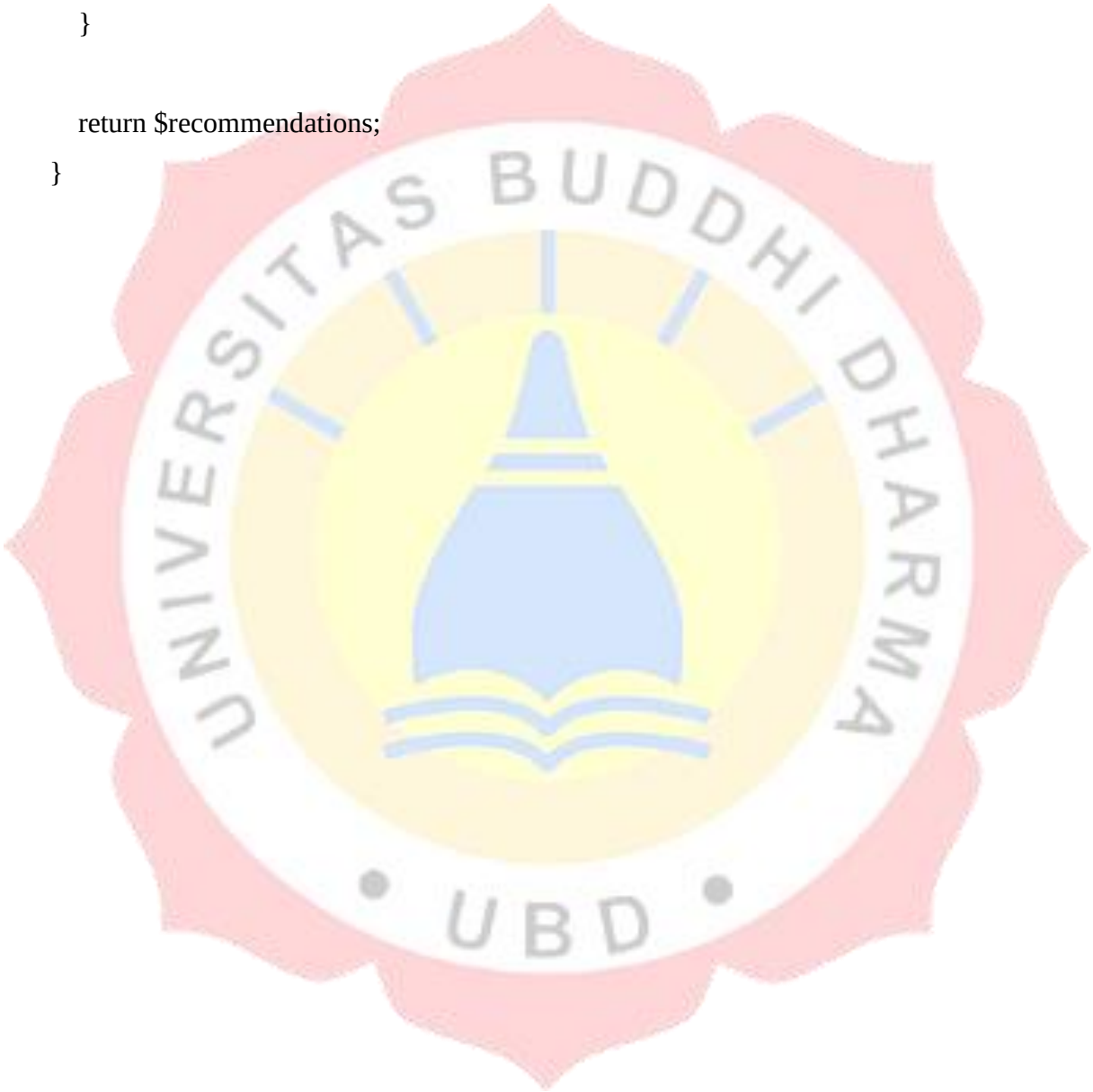
```
$filteredConfidence = array_filter($confidence2, fn($confidence) => $confidence >= $thresholdConfidence);

// get recommendation based on confidence
$recommendations = $this->getRecommendations($filteredConfidence);

// set array recommendations
$detailedRecommendations = [];

foreach ($recommendations as $recommendation) {
    $itemPair = explode(',', $recommendation['item_pair']);

    if (in_array((string)$productId, $itemPair)) {
        foreach ($itemPair as $item) {
            $product = $this->product->find($item);
            if ($product) {
                $detailedRecommendations[] = [
                    'id' => $product->id,
                    'product_name' => $product->product_name,
                    'price' => $product->price,
                    'catalogue' => $product->catalogue,
                    'confidence' => $recommendation['confidence'],
                ];
            }
        }
    }
}

return response()->json([
    'success'          => true,
    'productId'        => $productId,
    'apriori'          => $recommendations,
    'recommendations'   => collect($detailedRecommendations)->unique('id')->values()->all(),
```

```
        ]);
       } catch (Exception $error) {
        return response()->json([
            'success'   => false,
            'message'   => $error->getMessage()
        ], 500);
       }
    }

    private function formatItemPairs($filteredItemSet)
    {
        // set data to pairs product (array)
        $formattedPairs = [];
        foreach ($filteredItemSet as $pair => $confidenceValue) {
            $products = explode(',', $pair);
            $formattedPairs[] = $products;
        }

        return $formattedPairs;
    }

    private function calculateSupport($itemSets, $transactions)
    {
        $support = [];
        $totalTransactions = count(array_unique(array_column($transactions->toArray(),
'order_id')));

        if ($totalTransactions == 0) return [];

        $groupedTransactions = $transactions->groupBy('order_id')->map(function
($transactionGroup) {
            return $transactionGroup->pluck('product_id')->toArray();
        });
```

```
        foreach ($itemSets as $itemSet) {
            $count = 0;
            $itemSet = (array) $itemSet;
            foreach ($groupedTransactions as $productsInTransaction) {
                if (count(array_intersect($itemSet, $productsInTransaction)) ==
count($itemSet)) {
                    $count++;
                }
            }

            $support[implode(',', $itemSet)] = $count / $totalTransactions;
        }

        return $support;
    }

    private function generateItemPairs($itemSet)
    {
        $pairs = [];
        for ($i = 0; $i < count($itemSet); $i++) {
            for ($j = $i + 1; $j < count($itemSet); $j++) {
                if ($itemSet[$i] !== $itemSet[$j]) {
                    $pairs[] = [$itemSet[$i], $itemSet[$j]];
                }
            }
        }
        return $pairs;
    }

    private function calculateConfidence($itemPairs, $transactions)
    {
        $confidence = [];

        $groupedTransactions = [];
```

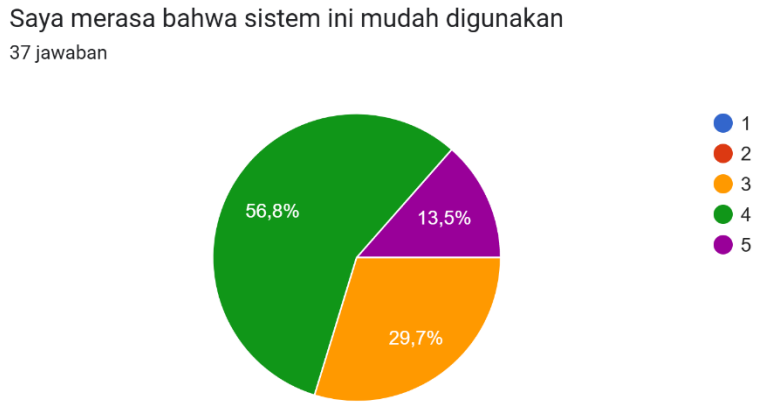
```
    foreach ($transactions as $transaction) {
        $groupedTransactions[$transaction->order_id][] = $transaction->product_id;
    }

    foreach ($itemPairs as $pair) {
        $itemA = $pair[0];
        $itemB = $pair[1];

        $countAB = 0;
        $countA = 0;

        foreach ($groupedTransactions as $productsInTransaction) {
            if (in_array($itemA, $productsInTransaction)) {
                $countA++;
                if (in_array($itemB, $productsInTransaction)) {
                    $countAB++;
                }
            }
        }

        if ($countA > 0) {
            $confidence[implode(',', $pair)] = $countAB / $countA;
        } else {
            $confidence[implode(',', $pair)] = 0;
        }
    }

    return $confidence;
}

private function getRecommendations($confidence)
{
    arsort($confidence);
```

```
    $recommendations = [];
    foreach ($confidence as $pair => $conf) {
        $recommendations[] = [
            'item_pair' => $pair,
            'confidence' => $conf
        ];
    }

    return $recommendations;
}
```



Lampiran A- 2 Grafik Jawaban Kuesioner

Jawaban Pertanyaan 1



Jawaban Pertanyaan 2

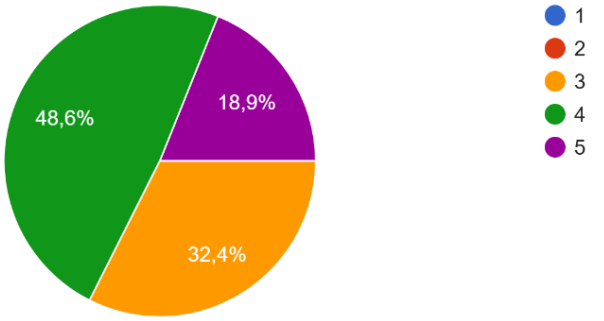
Jawaban Pertanyaan 3

Jawaban Pertanyaan 4

Jawaban Pertanyaan 5

Saya dapat memahami dan mengingat setiap fitur yang terdapat dalam sistem ini

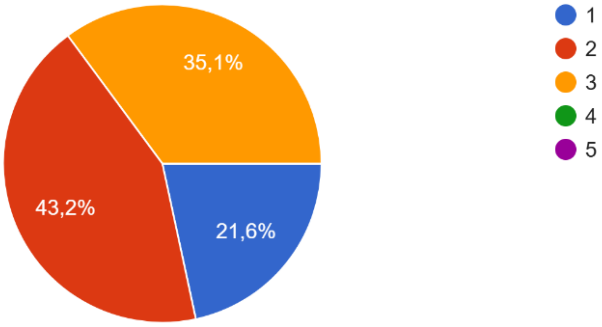
37 jawaban



Jawaban Pertanyaan 6

Saya merasa kesulitan dalam menggunakan sistem ini

37 jawaban



Jawaban Pertanyaan 7

Jawaban Pertanyaan 8

Jawaban Pertanyaan 9

Jawaban Pertanyaan 10

No	Analisis Kebutuhan Sistem Saya Ingin Sistem Dapat :	Keterangan
1	menonaktifkan semua menu yang tidak tersedia	
2	menampilkan data 3x transaksi	
3	menampilkan detail ingredients menu	
4	menerapkan metode Cross - Selling	
5	Tampilan yang interaktif	
6	menampilkan menu 3x Best Sellor	
7		
8		
9		
10		

Tangerang, 29-10-2024

Pembimbing	Responden	Mahasiswa
Benny Daniawan	Pio. S.	Patricia Fransiska
NDIN : 0424049006	Supervisor	20200700049

No	Analisis Kebutuhan Sistem Saya Ingin Sistem Dapat :	Keterangan
1	Menampilkan menu best seller	
2	Dapat mengubah detail menu	
3	Menampilkan harga menu	
4	Menampilkan informasi mengenai menu yang dishon	
5	Menampilkan menu terbaru	
6	Dapat mengupdate menu	
7		
8		
9		
10		

Tangerang, 29-10-2024

Pembimbing	Responden	Mahasiswa
Benny Daniawan	AWAL	Patricia Fransiska
NDIN : 0424049006	Kepala Koki	20200700049

No	Analisis Kebutuhan Sistem Saya Ingin Sistem Dapat :	Keterangan
1	DAPAT MENGUPDATE MENU	
2	TERDAPAT ULASAN MENGENAI MENU	
3	DAPAT MENAMPILKAN INFORMASI PROMO	
4	MENDAFTARKAN IDENTITAS CUSTOMER	
5	MENDAPATKAN POIN	
6		
7		
8		
9		
10		

Tangerang, 29-10-2024

Pembimbing

Benny Daniawan

NDIN : 0424049006

Responden

Intan Rusmawati

Kepala Barista

Mahasiswa

Patricia Fransiska

20200700049

UNIVERSITAS BUDDHI DHARMA
Jl. Imam Bonjol No. 41 Karawaci Ilir, Tangerang
021 5517853 / 021 5586822 admin@buddhidharma.ac.id

KARTU BIMBINGAN TA/SKRIPSI

NIM	:20200700049
Nama Mahasiswa	:PATRICIA FRANSISKA
Fakultas	:Sains dan Teknologi
Program Studi	:Sistem Informasi
Jenjang	:Strata Satu
Tahun Akademik/Semester	:2024/2025 Ganjil
Dosen Pembimbing	:Benny Daniawan, M.Kom
Judul Skripsi	:ANALISIS DAN PERANCANGAN SISTEM INFORMASI PEMESANAN MENU MAKANAN DAN MINUMAN BERBASIS WEB MENGGUNAKAN METODE APRIORI

Tanggal	Catatan	Paraf
2024-09-13	Breifing awal	
2024-09-27	Perbaikan berdasar panduan Skripsi	
2024-10-04	Perbaikan tabel RE	
2024-10-18	Perancangan Sistem	
2024-10-18	Desain Blue Print Perancangan Sistem	
2024-10-25	Perbaikan tampilan perancangan sistem	
2024-11-08	Desain Database	
2024-11-21	Bab 3 usecase, Activity, dan Sequence	
2024-11-29	Perbaikan Bab 3	
2024-12-06	Perbaikan bab 4	
2024-12-13	Pengujian Sistem SUS	
2024-12-19	Penyelesaian Bab 4 dan 5	
2024-12-20	Acc maju sidang	

Mengetahui
Ketua Program Studi

Benny Daniawan, M.Kom

Tangerang, 14 January 2025
Pembimbing

Benny Daniawan, M.Kom

Lampiran Dokumentasi Hasil Observasi

Restoran Srasi

Restoran Oniku

Restoran Bum