

## BAB V

### SIMPULAN DAN SARAN

#### 5.1 Simpulan

Hasil analisis dari penelitian tentang Aplikasi Metode Asosiasi dengan menggunakan *Algoritma* Apriori untuk mengidentifikasi pola transaksi di Teratai Café menghasilkan simpulan sebagai berikut:

1. Dapat mengimplementasikan paket menu penjualan berdasarkan pola pembelian yang berlaku di Teratai Café. Hal ini terbukti dari hasil yang memiliki *Best Recommendation Menu*.
2. Manajemen Teratai Café dapat menganalisis semua data pola penjualan untuk mengatur dan mengubah bahan baku yang tersedia sesuai kebutuhan, dengan menggunakan *data itemset* yang memenuhi nilai *confidence*.
3. Aplikasi yang diamati dapat menyajikan rekomendasi paket menu combo berdasarkan pola penjualan pelanggan di Teratai Café, sehingga membantu dalam meningkatkan penjualan dan menarik perhatian konsumen yang signifikan.

#### 5.2 Saran

Saran untuk penelitian tambahan yang bertujuan untuk meningkatkan dan memajukan sistem ini adalah:

1. Penelitian selanjutnya bertujuan untuk menggunakan algoritma dan metodologi tambahan untuk memfasilitasi analisis komparatif, menghasilkan perbedaan hasil yang beragam dan menarik.
2. Aplikasi yang telah dibuat masih berbasis tampilan *web desktop*, sehingga harapannya dapat dikembangkan sehingga mempunyai aplikasi berbasis *mobile* sehingga pengaksesan aplikasi dapat dilakukan secara fleksibel.

## DAFTAR PUSTAKA

- Achmad, F., Nurdiawan, O., & Arie Wijaya, Y. (2023). Analisa Pola Transaksi Pembelian Konsumen Pada Toko Ritel Kesehatan Menggunakan Algoritma Fp-Growth. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(1), 168–175. <https://doi.org/10.36040/jati.v7i1.6210>
- Anjelita, P., & Rosiska, E. (2019). Rancang Bangun Sistem Informasi E-Learning Pada Smk Negeri 3 Batam. *Computer and Science Industrial Engineering (COMASIE)*, 1(1), 132–141.
- Arhami, Muhammad; Nasir, M. (2020). *Data Mining - Algoritma dan Implementasi*. Penerbit Andi. <https://books.google.co.id/books?id=AtcCEAAAQBAJ&hl=id>
- Arhami, M., & Nasir, M. (2020). *Data Mining, Algoritma Dan Implementasi*. Andi Offset. <https://www.gramedia.com/products/data-mining-algoritma-dan-implementasi>
- Chandra, M. D., Irawan, E., Saragih, I. S., Windarto, A. P., & Suhendro, D. (2021). Penerapan Algoritma K-Means dalam Mengelompokkan Balita yang Mengalami Gizi Buruk Menurut Provinsi. *BIOS: Jurnal Teknologi Informasi Dan Rekayasa Komputer*, 2(1), 30–38. <https://doi.org/10.37148/bios.v2i1.19>
- Johan, Ragil Andika; Himilda, Rispani; Auliza, N. (2019). *J-TIFA*. 2617(2), 1–7.
- M. Afdal, M. A., & Rosadi, M. (2019). Penerapan Association Rule Mining Untuk Analisis Penempatan Tata Letak Buku Di Perpustakaan Menggunakan Algoritma Apriori. *Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi*, 5(1), 99. <https://doi.org/10.24014/rmsi.v5i1.7379>
- Manullang, F. (2020). *Penerapan Data Mining Penjualan Rumah dengan Menggunakan Algoritma Apriori*. <http://repository.upbatam.ac.id/2381/1/cover> s.d bab III.pdf
- Marisma, M., Wibowo, F., Purwanto, L. A., & Fauzan, A. (2024). Desain Pengembangan Aplikasi Kasir Apotek Berbasis Web. *Journal of Intelligent Systems and Information Technology*, 1(1), 6–15.
- Martin Halomoan Lumbangaol, M. R. R. (2020). Rancang Bangun Sistem Informasi Penjualan dan Penyewaan Properti Berbasis WEB Di Kota Batam. *Jurnal Comasie*, 01(03), 83–92.

- Muhyidin, M. A., Sulhan, M. A., & Sevtiana, A. (2020). Perancangan Ui/Ux Aplikasi My Cic Layanan Informasi Akademik Mahasiswa Menggunakan Aplikasi Figma. *Jurnal Digit*, 10(2), 208. <https://doi.org/10.51920/jd.v10i2.171>
- Munir, R., & Leony, L. (2016). *ALGORITMA DAN PEMROGRAMAN DALAM BAHASA PASCAL, C, DAN C++*. Informatika Bandung. <https://perpustakaan.binadarma.ac.id/opac/detail-opac?id=96>
- Muslihudin, M. O. (2016). *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*. Andi Offset Yogyakarta. [https://books.google.co.id/books?hl=en&lr=&id=2SU3DgAAQBAJ&oi=fnd&pg=PR3&dq=info:C5SUxhXrpOcJ:scholar.google.com&ots=T0110vUXPT&sig=Mqlaj-m54XTTOPBEi01UN56Ej9o&redir\\_esc=y#v=onepage&q&f=false](https://books.google.co.id/books?hl=en&lr=&id=2SU3DgAAQBAJ&oi=fnd&pg=PR3&dq=info:C5SUxhXrpOcJ:scholar.google.com&ots=T0110vUXPT&sig=Mqlaj-m54XTTOPBEi01UN56Ej9o&redir_esc=y#v=onepage&q&f=false)
- Nada, B. K., & Ridho, M. R. (2021). Jurnal Comasie. *Comasie*, 6(2), 107–118.
- Nofitri, R., & Irawati, N. (2019). Analisis Data Hasil Keuntungan Menggunakan Software Rapidminer. *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, 5(2), 199–204. <https://doi.org/10.33330/jurtekxi.v5i2.365>
- Pradipta, N. A., & Untari N, R. D. H. (2024). Implementasi Algoritma Apriori Untuk Analisis Pola Pembelian Produk Donat Bolong. *Jutisi : Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi*, 13(1), 268. <https://doi.org/10.35889/jutisi.v13i1.1778>
- Rahardjo, S. S. (2020). *Akuntansi : Suatu Pengantar*.
- Rahmatiyah, H., Salkin, L., & Assaf, A. (2023). Jurnal Informatika dan Komputer. *Membuat Website UPTD Puskesmas Batumarta II Menggunakan PHP & MySQL, volume 1*(Vol 14, No.1 (2023)), 1. <https://garuda.ristekbrin.go.id/documents/detail/1064467>
- Rizky, F., Sudiarti, S., & Atika. (2023). Strategi Customer Relationship Management Dalam Mempertahankan Pelanggan Pada Coffee Shop Grama Sphere. *ECo-Fin*, 5(3), 247–259. <https://doi.org/10.32877/ef.v5i3.951>
- Rosmayati, I., Wahyuningsih, W., Harahap, E. F., & Hanifah, H. S. (2023). Implementasi Data Mining pada Penjualan Kopi Menggunakan Algoritma Apriori. *Jurnal Algoritma*, 20(1), 99–107. <https://doi.org/10.33364/algoritma/v.20-1.1259>
- Simanjuntak, A., Difa Wijaya, K., Permatasari, A., Astuti, A. L., & Zy, A. T. (2024). Pengujian Dan Penjaminan Kualitas Software Point Of Sale Penjualan Motor PT. XZY Dengan

Metode Black Box. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 8(1), 108–119.

Sugiyono. (2018). *METODE PENELITIAN KUANTITATIF, KUALITATIF DAN R&D*.

Tarigan, P. M. S., Hardinata, J. T., Qurniawan, H., Safii, M., & Winanjaya, R. (2022). Implementasi Data Mining Menggunakan Algoritma Apriori Dalam Menentukan Persediaan Barang. *Jurnal Janitra Informatika Dan Sistem Informasi*, 2(1), 9–19. <https://doi.org/10.25008/janitra.v2i1.142>

Tukino, T. (2020). Rancang Bangun Sistem Informasi E-Marketing Pada Pt Pulau Cahaya Terang. *Computer Based Information System Journal*, 8(1), 25–34. <https://doi.org/10.33884/cbis.v8i1.1680>

Wahyudi, D. M., & Ridho, R. M. (2019). Sistem Informasi Penjualan Mobil Bekas Berbasis Web Pada CV Phutu Oil Club di Kota Batam. *Computer and Science Industrial Engineering (COMASIE)*, 1(01), 102–111. <https://ejournal.upbatam.ac.id/index.php/comasiejournal/article/view/1565>

Yudhanto, Y., & Prasetyo, H. A. (2019). *Mudah Menguasai Framework Laravel*. Elex Media Komputindo. <https://books.google.co.id/books?id=8tKdDwAAQBAJ&printsec=frontcover&hl=id#v=onepage&q&f=false>



**LAMPIRAN-LAMPIRAN**

## DAFTAR RIWAYAT HIDUP



### Data Pribadi

Nama Lengkap : Aron Tansavero  
Tempat, Tanggal Lahir : Tangerang, 10 Mei 2001  
Jenis Kelamin : Laki-laki  
Agama : Buddha  
Alamat : Duta Graha Blok A No.33 – Tangerang  
No.Telp/HP : 08558299388  
Email : [arontansav@gmail.com](mailto:arontansav@gmail.com)

### PENDIDIKAN FORMAL

SD (2007-2013) : SD Strada Yosudarso - Curug  
SMP (2013-2016) : SMP Citra Berkat – Citra Raya  
SMK (2016-2019) : SMK Atisa Dipamkara – Karawaci  
Strata 1 ( 2019-Sekarang) : Universitas Buddhi Dharma Strata 1 Teknik Informatika

Tangerang, 4 Februari 2025  
Yang membuat pernyataan,

Aron Tansavero

## KUESIONER PENGGUNAAN APLIKASI

Ringkasan      Pertanyaan      **Individual**

< 1 dari 18 >       

Jawaban tidak dapat diedit

### KUESIONER PENILAIAN APLIKASI DATA MINING TERATA CAFE

Perkenalkan nama saya Aron Tansavero mahasiswa Teknik Informatika, Fakultas Sains Dan Teknologi di Universitas Buddhi Dharma, Tangerang. Saat ini saya sedang melakukan penelitian untuk skripsi saya yang berjudul **"IMPLEMENTASI DATA MINING DENGAN ALGORITMA APRIORI UNTUK MENGHASILKAN POLA PENJUALAN PADA CV TERATAI CAFE"**. Sehubungan dengan penelitian yang saya lakukan, saya bermaksud meminta kesediaan teman-teman dari Teratai Cafe untuk meluangkan waktu mengisi kuesioner ini dengan lengkap dan benar. Semua informasi yang diterima sebagai hasil kuesioner ini bersifat rahasia dan hanya akan digunakan untuk kepentingan akademis. Untuk kelancaran analisis data, terdapat beberapa kriteria untuk bisa mengisi kuesioner penelitian ini, yaitu(KUISIONER INI TIDAK DIPESERBARLUASKAN):

1. Pihak yang terlibat dalam staff Teratai Cafe
2. Pihak manajerial hingga owner Teratai Cafe
3. Pihak manajerial yang memiliki bisnis serupa (FnB)

Dimohon untuk dapat melihat aplikasi berbasis website ini terlebih dahulu setelah itu kembali memberikan penilaian terhadap aplikasi ini

Link AKSES :  
<https://teratai-cafe.shop/login>

id : admin@gmail.com  
password : password





1. Apakah aplikasi ini mudah digunakan? \*

- Ya
- Cukup
- Tidak

2. Apakah menu yang tersedia dalam aplikasi tersebut lengkap sesuai kebutuhan cafe? \*

- Ya
- Cukup
- Tidak

3. Apakah menu yang tersedia pada aplikasi tersebut dapat dipahami dengan baik? \*

- Ya
- Cukup
- Tidak



UBD

4. Apakah aplikasi tersebut dapat mengetahui pola transaksi pembelian konsumen dan menemukan aturan asosiasi dan paket penjualan? \*

- Ya
- Cukup
- Tidak

5. Apakah aplikasi tersebut dapat membantu dalam memberikan sebuah rekomendasi menu paket penjualan? \*

- Ya
- Cukup
- Tidak

6. Apakah aplikasi tersebut memiliki design yang menarik? \*

- Ya
- Cukup
- Tidak

• U B D •



7. Apakah dengan menggunakan sistem ini pada sebuah toko akan memberikan perubahan dalam penjualan maupun pekerjaan? \*

- Ya
- Cukup
- Tidak

8. Apakah aplikasi tersebut dapat berjalan dengan lancar? \*

- Ya
- Cukup
- Tidak

9. Apakah anda tertarik untuk menggunakan sistem ini? \*

- Ya
- Cukup
- Tidak



UBD

Masukkan Nama Anda	1. Apakah aplikasi ini mudah digunakan?	2. Apakah menu yang tersedia dalam aplikasi?	3. Apakah menu yang tersedia pada aplikasi?
William Albert	Ya	Ya	Cukup
Vany	Ya	Cukup	Cukup
Calvin	Ya	Ya	Ya
Melly	Cukup	Cukup	Cukup
Sintia	Ya	Ya	Cukup
Achiu	Ya	Ya	Ya
Sardy	Cukup	Ya	Ya
Rian	Ya	Cukup	Ya
Aheng	Ya	Cukup	Cukup
Tini	Ya	Ya	Ya
Ananda	Ya	Cukup	Ya
Ferdinan Sinaga	Ya	Ya	Cukup
Setiawan Chandra	Ya	Cukup	Ya
Christ	Ya	Ya	Ya
Edward	Ya	Ya	Ya
Wahyuda	Ya	Ya	Cukup
Fernandes Lado	Ya	Cukup	Cukup
Felix	Cukup	Ya	Ya

4. Apakah aplikasi tersebut dapat mengetahui?	5. Apakah aplikasi tersebut dapat membantu?	6. Apakah aplikasi tersebut memiliki design yang?
Ya	Ya	Ya
Ya	Ya	Ya
Ya	Ya	Ya
Cukup	Ya	Ya
Ya	Ya	Ya
Ya	Ya	Ya
Ya	Cukup	Ya
Cukup	Cukup	Ya
Cukup	Cukup	Ya
Ya	Ya	Ya
Cukup	Ya	Ya
Cukup	Cukup	Tidak
Ya	Ya	Cukup
Ya	Ya	Cukup
Ya	Ya	Ya
Ya	Ya	Cukup
Cukup	Ya	Cukup
Cukup	Cukup	Cukup

7. Apakah dengan menggunakan sistem ini p	8. Apakah aplikasi tersebut dapat berjalan de	9. Apakah anda tertarik untuk menggunakan
Cukup	Ya	Ya
Ya	Ya	Ya
Ya	Ya	Ya
Tidak	Ya	Cukup
Ya	Ya	Ya
Ya	Ya	Ya
Cukup	Cukup	Cukup
Tidak	Ya	Ya
Cukup	Ya	Ya
Cukup	Ya	Ya
Ya	Ya	Ya
Cukup	Cukup	Ya
Ya	Ya	Cukup
Ya	Ya	Ya
Ya	Ya	Tidak
Ya	Ya	Ya
Ya	Cukup	Ya
Ya	Ya	Cukup



## KARTU BIMBINGAN



**UNIVERSITAS BUDDHI DHARMA**

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

### KARTU BIMBINGAN TA/SKRIPSI

NIM : 20191000030  
Nama Mahasiswa : ARON TANSAVERO  
Fakultas : Sains dan Teknologi  
Program Studi : Teknik Informatika  
Jenjang : Strata Satu  
Tahun Akademik/Semester : 2024/2025 Ganjil  
Dosen Pembimbing : Hartana Wijaya, M.Kom  
Judul Skripsi : IMPLEMENTASI DATA MINING DENGAN ALGORITMA APRIORI  
UNTUK MENGHASILKAN POLA PENJUALAN PADA CV TERATAI  
CAFE

Tanggal	Catatan	Paraf
2024-09-11	Penulisan skripsi	
2024-09-25	Judul skripsi, Bab I	
2024-10-09	Revisi Bab I, Review Bab II	
2024-10-24	Revisi Bab I, Bab II, Lanjut Bab III	
2024-11-18	Revisi Bab II dan Bab III	
2024-12-02	ACC Bab I dan II, Revisi Bab III dan IV	
2024-12-05	Demo Aplikasi dan Revisi	
2024-12-12	Acc Bab III dan Bab IV, Revisi Program dan Bab V	
2024-12-13	Revisi Bab V, Daftar Pustaka. Rekomendasi Maju Sidang	
2024-12-16	Acc Bab V dan Daftar Pustaka	

Mengetahui  
Ketua Program Studi



Hartana Wijaya, M.Kom

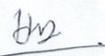
Tangerang, 15 January 2025  
Pembimbing



Hartana Wijaya, M.Kom

## REQUIREMENT ELICITATION

### Requirement Elicitation

No.	Analisa Kebutuhan Sistem	Keterangan
	Saya ingin sistem dapat :	
1)	Sistem dengan tampilan bagus / menarik	
2)	Integrasi online	
3)	Aplikasi mudah digunakan	
4)	Aplikasi dapat berjalan dengan lancar	
5)	Memiliki hasil laporan	

Tangerang, 16 Desember 2024

Pembimbing

Responden

Mahasiswa



Hartana Wijaya, M.Kom  
NIDN : 0412058102

Aron Tansavero  
20191000030

## CODE ALGORITMA APRIORI

### DataProcessController.php

```
<?php

namespace App\Http\Controllers\User;

use App\Http\Controllers\Controller;
use App\Models\DataProcessModel;
use App\Models\TransactionsModel;
use App\Models\BestSellerProduct;
use App\Models\Results;
use Illuminate\Http\Request;

class DataProcessController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        return inertia('User/Dashboard/DataProcess/Index');
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        //
    }
}
```

```

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $validatedData = $request->validate([
        'start_date' => 'required|date',
        'end_date' => 'required|date|after_or_equal:start_date',
        'minimum_support' => 'required|min:1|max:100',
        'minimum_confidence' => 'required|min:1|max:100',
    ]);
    $validatedData['minimum_support'] = (int) str_replace('%', '', $validatedData['minimum_support']);
    $validatedData['minimum_confidence'] = (int) str_replace('%', '', $validatedData['minimum_confidence']);

    /// Get input from request
    $start_date = $validatedData['start_date'];
    $end_date = $validatedData['end_date'];
    $minimum_support = $validatedData['minimum_support'];
    $minimum_confidence = $validatedData['minimum_confidence'];

    $checkDate = TransactionsModel::whereBetween('date', [$start_date, $end_date])->first();
    if ($checkDate == null) {
        return back()->with([
            'message' => "There is no data in the range date you selected please choose another date",
            'type' => "danger"
        ]);
    }

    // Get transactions within date range
    $transactionModels = \App\Models\TransactionsModel::whereBetween('date', [$start_date, $end_date])->get();

    // Get count transactions within date range

```

```
$totalTransactions = $transactionModels->count();
```

```
// *** Start Itemset1 ***
```

```
//Separate the variant column into rows by comma and get the unique values
```

```
$itemset1 = \DB::table('transactions_models')
```

```
->select('variant')
```

```
->whereBetween('date', [$start_date, $end_date])
```

```
->get()
```

```
->map(function($item) {
```

```
    return explode(",", $item->variant);
```

```
});
```

```
->flatten()
```

```
->unique()
```

```
->map(function($item) {
```

```
    return (object) [
```

```
        'variant' => trim($item)
```

```
    ];
```

```
});
```

```
// Get frequency of each variant
```

```
$itemset1Frequency = [];
```

```
foreach ($itemset1 as $variant) {
```

```
    $itemset1Frequency[$variant->variant] = 0;
```

```
    foreach ($transactionModels as $transaction) {
```

```
        $transactionVariants = array_map('trim', explode(",", $transaction->variant));
```

```
        if (in_array($variant->variant, $transactionVariants)) {
```

```
            $itemset1Frequency[$variant->variant]++;
```

```
        }
```

```
    }
```

```
}
```

```
// Calculate minimum support for each variant
```

```
$supportItemset1 = [];
```

```
foreach ($itemset1Frequency as $itemset1frequency => $frequency) {
```

```
    $supportItemset1[$itemset1frequency] = round($frequency / $totalTransactions * 100, 2);
```

```

}

// Filter items that pass and fail the minimum support
$passMinimumSupportItemset1 = array_filter($supportItemset1, function($support) use
($minimum_support) {
    return $support >= $minimum_support;
});

$failMinimumSupportItemset1 = array_filter($supportItemset1, function($support) use
($minimum_support) {
    return $support < $minimum_support;
});

arsort($passMinimumSupportItemset1);
arsort($failMinimumSupportItemset1);
// *** End Itemset1 ***

// *** Start Itemset2 ***
// Itemset2 - Get all possible combinations of 2 items
$Itemset2Combination = [];
$passKeys = array_keys($passMinimumSupportItemset1);

for ($i = 0; $i < count($passKeys); $i++) {
    for ($j = $i + 1; $j < count($passKeys); $j++) {
        $item = $passKeys[$i];
        $item2 = $passKeys[$j];
        $Itemset2Combination["$item,$item2"] = 0;
    }
}

$Itemset2 = array_keys($Itemset2Combination);

// Get frequency of each itemset2
$Itemset2Frequency = [];
foreach ($Itemset2 as $Itemset) {
    $Itemset = explode(",", $Itemset);
    $Item1 = trim($Itemset[0]);

```

```

$item2 = trim($itemset[1]);

$query = \DB::table('transactions_models')
    ->where('date', '>=', $start_date)
    ->where('date', '<=', $end_date);

$query->where(function($subquery) use ($item1, $item2) {
    $subquery->where('variant', 'LIKE', '%'.$item1.'%')
    ->where('variant', 'LIKE', '%'.$item2.'%');
});

$count = $query->count();

// Add itemset to $itemset2Frequency array with its frequency
$itemset2Frequency[implode(",", $itemset)] = $count;
}

// Calculate minimum support for each itemset2
$minimumSupportItemset2 = [];
foreach ($itemset2Frequency as $itemset => $frequency) {
    $minimumSupportItemset2[$itemset] = round($frequency / $totalTransactions * 100, 2);
}

// Filter items that pass and fail the minimum support
$passMinimumSupportItemset2 = array_filter($minimumSupportItemset2, function($support) use ($minimum_support) {
    return $support >= $minimum_support;
});

$failMinimumSupportItemset2 = array_filter($minimumSupportItemset2, function($support) use ($minimum_support) {
    return $support < $minimum_support;
});

// *** End Itemset2 ***

// *** Start Itemset3 ***
// Itemset3 - Get all possible combinations of 3 items

```

```

Itemset3Combination = [];
$passKeys = array_keys($passMinimumSupportItemset2);

for ($i = 0; $i < count($passKeys); $i++) {
    for ($j = $i + 1; $j < count($passKeys); $j++) {
        $item1 = explode(',', $passKeys[$i]);
        $item2 = explode(',', $passKeys[$j]);

        // Only generate combinations if the first item is the same
        if ($item1[0] === $item2[0]) {
            $intersect = array_intersect($item1, $item2);

            if (count($intersect) == 1) {
                $diff1 = array_diff($item1, $intersect);
                $diff2 = array_diff($item2, $intersect);
                $newCombination = array_merge($intersect, $diff1, $diff2);
                //dd($newCombination);
                $itemset3Combination[implode(',', $newCombination)] = 0;
            }
        }
    }
}

Itemset3 = array_keys($itemset3Combination);

// Get frequency of each itemset3
Itemset3Frequency = [];
foreach ($itemset3 as $itemset) {
    $itemset = explode(",", $itemset);
    $item1 = trim($itemset[0]);
    $item2 = trim($itemset[1]);
    $item3 = trim($itemset[2]);

    $query = \DB::table('transactions_models')
        ->where('date', '>=', $start_date)
        ->where('date', '<=', $end_date);
}

```

```

$query->where(function($subquery) use ($item1, $item2, $item3) {
    $subquery->where('variant', 'LIKE', '%'. $item1. '%')
        ->where('variant', 'LIKE', '%'. $item2. '%')
        ->where('variant', 'LIKE', '%'. $item3. '%');
});

$count = $query->count();

// Add itemset to $itemset3Frequency array with its frequency
$itemset3Frequency[implode(",", $itemset)] = $count;
}
// Calculate minimum support for each itemset3
$minimumSupportItemset3 = [];
foreach ($itemset3Frequency as $itemset => $frequency) {
    $minimumSupportItemset3[$itemset] = round($frequency / $totalTransactions * 100, 2);
}

// Filter items that pass and fail the minimum support
$passMinimumSupportItemset3 = array_filter($minimumSupportItemset3, function($support) use ($minimum_support) {
    return $support >= $minimum_support;
});

$failMinimumSupportItemset3 = array_filter($minimumSupportItemset3, function($support) use ($minimum_support) {
    return $support < $minimum_support;
});

// *** End Itemset3 ***

// *** Start Calculate Confidence 2 Itemset ***
$itemsetConfCombination2 = [];
$itemABFrequency2 = [];
$itemAFrequency2 = [];
$valueConfidence2 = [];
$itemBFrequency2 = [];
$valueLift2 = [];

```

```

$confidenceItemset2=[];
$passMinimumConfidenceItemset2 = [];
$failMinimumConfidenceItemset2 = [];

foreach ($passMinimumSupportItemset2 as $itemset => $support) {
    $items = explode(",", $itemset);
    $item1 = trim($items[0]);
    $item2 = trim($items[1]);

    // Create combination of 2 items and Get frequency of item1
    $itemsetConfCombination2[] = [$item1, $item2];
    $itemABFrequency2[] = $itemset2Frequency[$itemset];
    $itemAFrequency2[] = $itemset1Frequency[$item1];
    $valueConfidence2[] = round($itemset2Frequency[$itemset] / $itemset1Frequency[$item1] * 100 , 2);
    $itemBFrequency2[] = $itemset1Frequency[$item2];
    $valueLift2[] = round(($itemset2Frequency[$itemset]/$itemset1Frequency[$item1]) /
($itemset1Frequency[$item2]/$totalTransactions), 2);

    $itemsetConfCombination2[] = [$item2, $item1];
    $itemABFrequency2[] = $itemset2Frequency[$itemset];
    $itemAFrequency2[] = $itemset1Frequency[$item2];
    $valueConfidence2[] = round($itemset2Frequency[$itemset] / $itemset1Frequency[$item2] * 100 , 2);
    $itemBFrequency2[] = $itemset1Frequency[$item1];
    $valueLift2[] = round(($itemset2Frequency[$itemset]/$itemset1Frequency[$item2]) /
($itemset1Frequency[$item1]/$totalTransactions), 2);
}

$confidenceItemset2 = array_map(function($itemset, $abFreq, $afreq, $confidence2) {
    return array_merge($itemset, array($abFreq, $afreq, $confidence2));
}, $itemsetConfCombination2, $itemABFrequency2, $itemAFrequency2, $valueConfidence2);

// Filter items that pass and fail the minimum confidence
$passMinimumConfidenceItemset2 = array_values(array_filter($confidenceItemset2,
function($confidence2) use ($minimum_confidence) {
    return $confidence2[4] >= $minimum_confidence;
}));
}));

```

```

    $failMinimumConfidenceItemset2 = array_values(array_filter($confidenceItemset2,
function($confidence2) use ($minimum_confidence) {
    return $confidence2[4] < $minimum_confidence;
}));

$valueLiftItemset2 = array_map(function($itemset, $confidence2, $bFreq, $lift2) {
    return array_merge($itemset, array($confidence2, $bFreq, $lift2));
}, $itemsetConfCombination2, $valueConfidence2, $itemBFrequency2, $valueLift2);

$passLiftItemset2 = array_values(array_filter($valueLiftItemset2, function($confidence2) use
($minimum_confidence) {
    return $confidence2[2] >= $minimum_confidence && $confidence2[4] >= 1;
}));

$failLiftItemset2 = array_values(array_filter($valueLiftItemset2, function($confidence2) use
($minimum_confidence) {
    return $confidence2[2] >= $minimum_confidence && $confidence2[4] < 1;
}));

//sort highest to lowest $passLiftItemset2 and $failLiftItemset2
usort($passLiftItemset2, function($a, $b) {
    if ($b[2] === $a[2]) {
        return $b[3] <=> $a[3];
    }
    return $b[2] <=> $a[2];
});

usort($failLiftItemset2, function($a, $b) {
    if ($b[2] === $a[2]) {
        return $b[3] <=> $a[3];
    }
    return $b[2] <=> $a[2];
});

// *** End Calculate Confidence 2 Itemset ***

// *** Start Calculate Confidence 3 Itemset ***

```

```

ItemsetConfCombination3 = [];
ItemABFrequency3 = [];
ItemAFrequency3 = [];
SvalueConfidence3 = [];
ItemBFrequency3 = [];
SvalueLift3 = [];

SconfidenceItemset3=[];
$passMinimumConfidenceItemset3 = [];
$failMinimumConfidenceItemset3 = [];

foreach ($passMinimumSupportItemset3 as $Itemset => $support) {
    $Items = explode(",", $Itemset);
    $Item1 = trim($Items[0]);
    $Item2 = trim($Items[1]);
    $Item3 = trim($Items[2]);

    // Create combination of 3 items and Get frequency of item1
    $ItemsetConfCombination3[] = [$Item1, $Item2, $Item3];
    $ItemABFrequency3[] = $Itemset3Frequency[$Itemset];
    $ItemAFrequency3[] = $Itemset2Frequency[$Item1 . ' . ' . $Item2];
    $valueConfidence3[] = round($Itemset3Frequency[$Itemset] / $Itemset2Frequency[$Item1 . ' . ' . $Item2] *
100 , 2);
    //lift itemset3
    $ItemBFrequency3[] = $Itemset1Frequency[$Item3];

    $valueLift3[] = round(($Itemset3Frequency[$Itemset]/$Itemset2Frequency[$Item1 . ' . ' . $Item2]) /
($Itemset1Frequency[$Item3]/$totalTransactions), 2);

    $ItemsetConfCombination3[] = [$Item1, $Item3, $Item2];
    $ItemABFrequency3[] = $Itemset3Frequency[$Itemset];
    $ItemAFrequency3[] = $Itemset2Frequency[$Item1 . ' . ' . $Item3];
    $valueConfidence3[] = round($Itemset3Frequency[$Itemset] / $Itemset2Frequency[$Item1 . ' . ' . $Item3] *
100 , 2);
    //lift itemset3
    $ItemBFrequency3[] = $Itemset1Frequency[$Item2];

    $valueLift3[] = round(($Itemset3Frequency[$Itemset]/$Itemset2Frequency[$Item1 . ' . ' . $Item3]) /
($Itemset1Frequency[$Item2]/$totalTransactions), 2);
}

```

```

ItemsetConfCombination3[] = [Item2, Item3, Item1];
ItemABFrequency3[] = Itemset3Frequency[Itemset];
ItemAFrequency3[] = Itemset2Frequency[Item2 . ' ' . Item3];
$valueConfidence3[] = round(Itemset3Frequency[Itemset] / Itemset2Frequency[Item2 . ' ' . Item3] *
100 , 2);
//lift itemset3
ItemBFrequency3[] = Itemset1Frequency[Item1];
$valueLift3[] = round((Itemset3Frequency[Itemset]/Itemset2Frequency[Item2 . ' ' . Item3]) /
(Itemset1Frequency[Item1]/$totalTransactions), 2);
}

$confidenceItemset3 = array_map(function($itemset, $abFreq3, $aFreq3, $confidence3) {
    return array_merge($itemset, array($abFreq3, $aFreq3, $confidence3));
}, ItemsetConfCombination3, ItemABFrequency3, ItemAFrequency3, $valueConfidence3);

// Filter items that pass and fail the minimum confidence
$passMinimumConfidenceItemset3 = array_values(array_filter($confidenceItemset3,
function($confidence3) use ($minimum_confidence) {
    return $confidence3[5] >= $minimum_confidence;
}));

$failMinimumConfidenceItemset3 = array_values(array_filter($confidenceItemset3,
function($confidence3) use ($minimum_confidence) {
    return $confidence3[5] < $minimum_confidence;
}));

$valueLiftItemset3 = array_map(function($itemset, $confidence3, $bFreq, $lift3) {
    return array_merge($itemset, array($confidence3, $bFreq, $lift3));
}, ItemsetConfCombination3, $valueConfidence3, ItemBFrequency3, $valueLift3);

$passLiftItemset3 = array_values(array_filter($valueLiftItemset3, function($confidence3) use
($minimum_confidence) {
    return $confidence3[3] >= $minimum_confidence && $confidence3[5] >= 1;
}));

$failLiftItemset3 = array_values(array_filter($valueLiftItemset3, function($confidence3) use
($minimum_confidence) {

```

```

return $confidence3[3] >= $minimum_confidence && $confidence3[5] < 1;
});

//sort highest to lowest $passLiftItemset3 and $failLiftItemset3
usort($passLiftItemset3, function($a, $b) {
    if ($b[3] === $a[3]) {
        return $b[5] <=> $a[5];
    }
    return $b[3] <=> $a[3];
});

usort($failLiftItemset3, function($a, $b) {
    if ($b[3] === $a[3]) {
        return $b[5] <=> $a[5];
    }
    return $b[3] <=> $a[3];
});
// *** End Calculate Confidence 3 Itemset ***

//save data to database
$data = DataProcessModel::create([
    'user_id' => auth()->user()->id,
    'start_date' => $validatedData['start_date'],
    'end_date' => $validatedData['end_date'],
    'min_support' => $validatedData['minimum_support'],
    'min_confidence' => $validatedData['minimum_confidence']
]);

$itemsetResult = $this->resultItemset($passMinimumSupportItemset1,
    $failMinimumSupportItemset1, $passLiftItemset2, $passLiftItemset3);

return inertia('User/Dashboard/DataProcess/Index', [
    'minimum_support' => $minimum_support,
    'minimum_confidence' => $minimum_confidence,
    'totalTransactions' => $totalTransactions,
    'itemset1Frequency' => $itemset1Frequency,

```

```

'passMinimumSupportItemset1' => $passMinimumSupportItemset1,
'failMinimumSupportItemset1' => $failMinimumSupportItemset1,
'itemset2Frequency' => $itemset2Frequency,
'passMinimumSupportItemset2' => $passMinimumSupportItemset2,
'failMinimumSupportItemset2' => $failMinimumSupportItemset2,
'itemset3Frequency' => $itemset3Frequency,
'passMinimumSupportItemset3' => $passMinimumSupportItemset3,
'failMinimumSupportItemset3' => $failMinimumSupportItemset3,
'passMinimumConfidenceItemset2' => $passMinimumConfidenceItemset2,
'failMinimumConfidenceItemset2' => $failMinimumConfidenceItemset2,
'passMinimumConfidenceItemset3' => $passMinimumConfidenceItemset3,
'failMinimumConfidenceItemset3' => $failMinimumConfidenceItemset3,
'passLiftItemset2' => $passLiftItemset2,
'failLiftItemset2' => $failLiftItemset2,
'passLiftItemset3' => $passLiftItemset3,
'failLiftItemset3' => $failLiftItemset3,
]);
}

private function resultItemset($passMinimumSupportItemset1, $failMinimumSupportItemset1,
$passLiftItemset2, $passLiftItemset3)
{
    //Get data_process_model_id
    $data_process_id = DataProcessModel::where('user_id', auth()->user()->id)->orderBy('id', 'desc')->first()-
>id;

    // *** Start Save Result ***
    // Best Seller Product
    //dd($passMinimumSupportItemset1);
    if (!empty($passMinimumSupportItemset1)) {
        foreach ($passMinimumSupportItemset1 as $variant => $support) {
            $dataItemset1 = new BestSellerProduct();
            $dataItemset1->user_id = auth()->user()->id;
            $dataItemset1->data_process_id = $data_process_id;
            $dataItemset1->itemset = $variant;
            $dataItemset1->support = $support;
            $dataItemset1->status = 'Pass';
        }
    }
}

```

```

        $dataItemset1->save();
    }
}

if (empty($passMinimumSupportItemset1)) {
    foreach ($failMinimumSupportItemset1 as $variant => $support) {
        $dataItemset1 = new BestSellerProduct();
        $dataItemset1->user_id = auth()->user()->id;
        $dataItemset1->data_process_id = $data_process_id;
        $dataItemset1->itemset = $variant;
        $dataItemset1->support = $support;
        $dataItemset1->status = 'Fail';
        $dataItemset1->save();
    }
}

if (!empty($passLiftItemset2)) {
    foreach ($passLiftItemset2 as $itemset) {
        $item1 = trim($itemset[0]);
        $item2 = trim($itemset[1]);
        $confidence = $itemset[2];
        $freqb = $itemset[3];
        $lift = $itemset[4];

        $data = new Results();
        $data->user_id = auth()->user()->id;
        $data->data_process_id = $data_process_id;
        $data->itemset = $item1 . ' ' . $item2;
        $data->confidence = $confidence;
        $data->freqb = $freqb;
        $data->lift = $lift;
        $data->status = 'Valid';
        $data->save();
    }
}
}

```

```

if (!empty($passLiftItemset3)) {
    foreach ($passLiftItemset3 as $itemset) {
        $item1 = trim($itemset[0]);
        $item2 = trim($itemset[1]);
        $item3 = trim($itemset[2]);
        $confidence = $itemset[3];
        $freqb = $itemset[4];
        $lift = $itemset[5];

        $data = new Results();
        $data->user_id = auth()->user()->id;
        $data->data_process_id = $data_process_id;
        $data->itemset = $item1 . ' . ' . $item2 . ' . ' . $item3;
        $data->confidence = $confidence;
        $data->freqb = $freqb;
        $data->lift = $lift;
        $data->status = 'Valid';
        $data->save();
    }
}

/**
 * Display the specified resource.
 *
 * @param \App\Models\DataProcessModel $dataProcessModel
 * @return \Illuminate\Http\Response
 */
public function show(DataProcessModel $dataProcessModel)
{
    //
}

/**
 * Show the form for editing the specified resource.

```

```

*
* @param \App\Models\DataProcessModel $dataProcessModel
* @return \Illuminate\Http\Response
*/
public function edit(DataProcessModel $dataProcessModel)
{
    //
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Models\DataProcessModel $dataProcessModel
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, DataProcessModel $dataProcessModel)
{
    //
}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Models\DataProcessModel $dataProcessModel
 * @return \Illuminate\Http\Response
 */
public function destroy(DataProcessModel $dataProcessModel)
{
    //
}
}

```



## Teratai Café & Resto

Jl. Gatot Subroto No.10, RT.003/RW.001, Jatiuwung, Kec. Cibodas, Kota  
Tangerang, Banten 15138

Perihal : Surat Persetujuan Penelitian Skripsi

Dengan Hormat,

Yang bertanda tangan dibawah ini adalah Pemilik dari Teratai Cafe & Resto  
memberitahukan bahwa :

Nama : Sheng Chin

Jabatan : Pemilik Teratai Cafe & Resto

Menyatakan bahwa benar adanya mahasiswa/i Universitas Buddhi Dharma Tangerang  
dengan data :

Nama : Aron Tansavero

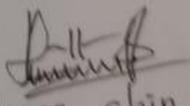
NIM : 20191000030

Jurusan : Teknik Informatika

Nama tersebut adalah benar mahasiswa Universitas Buddhi Dharma Tangerang dan telah  
melakukan penelitian di Bengkel kami dengan judul " *IMPLEMENTASI DATA MINING  
DENGAN ALGORITMA APRIORI UNTUK MENGHASILKAN POLA PENJUALAN PADA CV  
TERATAI CAFE*" untuk memenuhi salah satu syarat mengikuti skripsi dan mencapai gelar  
sarjana Program Studi Teknik Informatika di Universitas Buddhi Dharma.

Demikian surat keterangan ini kami buat dengan sebenar-benarnya agar dapat  
dipergunakan sebagaimana mestinya.

Tangerang 20 November 2024

  
Sheng chin