



Implementasi Algoritma *Improve Apriori* Terhadap Keluarga Beresiko *Stunting*

Muhammad Habib Nazlis¹, Fitri Insani^{*2}, Alwis Nazir³, Iis Afrianty⁴

Email: ¹11950115129@students.uin-suska.ac.id, ²fitri.insani@uin-suska.ac.id, ³alwis.nazir@uin-suska.ac.id,

⁴iis.afrianty@uin-suska.ac.id

^{1,2,3,4}Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau

Diterima: 19 Desember 2024 | Direvisi: - | Disetujui: 27 Desember 2024

©2020 Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Muhammadiyah Riau, Indonesia

Abstrak

Stunting merupakan masalah kesehatan serius di Indonesia, terutama pada keluarga dengan kondisi sosial-ekonomi rendah. Namun dengan tidak adanya ukuran kondisi sosial atau kriteria keluarga yang pasti dalam mengakibatkan keluarga yang beresiko sulit untuk diprediksi. Penelitian ini bertujuan untuk menemukan pola hubungan antar 17 kriteria yang memengaruhi risiko *stunting*, seperti usia ibu, jumlah anak, jenis lantai rumah, hingga akses terhadap air bersih, dengan meningkatkan efisiensi proses melalui penggunaan teknik *hash-based* pada algoritma apriori. Data penelitian diperoleh dari keluarga di Kecamatan Tuah Madani, Pekanbaru, dan dianalisis menggunakan preprocessing serta transformasi data. Implementasi algoritma ini dalam sistem informasi berbasis web memungkinkan analisis cepat dan efisien untuk mengidentifikasi risiko *stunting* berdasarkan kombinasi kriteria yang relevan. Hasil analisis menunjukkan bahwa beberapa kriteria, seperti usia ibu di atas 35 tahun, status sebagai pasangan usia subur (PUS), dan jumlah anak lebih dari tiga, memiliki keterkaitan signifikan terhadap risiko *stunting* dengan nilai *support* 37.54% dan *confidence* 83.16%. Penelitian ini berkontribusi dalam penyediaan metode yang efisien untuk analisis risiko *stunting* serta memberikan dasar bagi intervensi kesehatan yang lebih tepat sasaran. Bagi peneliti selanjutnya, disarankan untuk memperluas cakupan data dengan melibatkan lebih banyak wilayah dan periode waktu yang berbeda untuk meningkatkan generalisasi hasil. Selain itu, penambahan variabel lain seperti status gizi ibu atau tingkat pendidikan kepala keluarga dapat memberikan *insight* lebih mendalam dalam memahami pola risiko *stunting*.

Kata kunci: *Stunting; Improve Apriori; Hash-based; Data mining; Association rule*

Implementation of *Improve Apriori* Algorithm for Families at Risk of *Stunting*

Abstract

Stunting is a serious health issue in Indonesia, particularly among families with low socio-economic conditions. However, the lack of precise criteria or measurements of social conditions contributing to at-risk families makes prediction challenging. This study aims to identify patterns of relationships among 17 criteria influencing *stunting* risk, such as maternal age, number of children, type of flooring in the house, and access to clean water, by enhancing the efficiency of the Apriori algorithm through hash-based techniques. Data were obtained from families in Tuah Madani District, Pekanbaru, and analyzed using data preprocessing and transformation methods. The implementation of this algorithm within a web-based information system enables rapid and efficient analysis to identify *stunting* risks based on relevant combinations of criteria. The analysis results indicate that certain criteria, such as maternal age above 35 years, status as a couple of childbearing age (PUS), and having more than three children, are significantly associated with *stunting* risk, with a support value of 37.54% and a confidence level of 83.16%. This study contributes to the development of efficient methods for *stunting* risk analysis and provides a foundation for more targeted health interventions. Future researchers are advised to expand the data scope by including additional regions and different time periods to improve result generalization. Furthermore, incorporating other variables, such as maternal nutritional status or the education level of household heads, may offer deeper insights into understanding *stunting* risk patterns.

Keywords: *Stunting; Improve Apriori; Hash-based; Data mining; Association rule*

1. PENDAHULUAN

Permasalahan gizi sudah menjadi masalah serius yang harus segera ditangani setiap negara. Terutama negara kita Indonesia, negara berkembang dengan angka kemiskinan yang termasuk kedalam 100 negara paling miskin di dunia menurut *Gross National Income* (GNI) pada tahun 2020, yang dapat mendukung naiknya juga resiko keluarga untuk terkena permasalahan pada gizi. Salah satu permasalahan gizi tersebut merupakan *stunting*. Menurut WHO (2015), *stunting* adalah hambatan pada pertumbuhan anak yang disebabkan oleh kekurangan gizi kronis dan infeksi berulang. *Stunting* juga merupakan kondisi gangguan pada anak yang diakibatkan oleh gizi yang buruk, sehingga menyebabkan terhambatnya perkembangan secara tidak maksimal pada kognitif, motorik, dan verbal. *Stunting* ini dapat berdampak jangka panjang jika tidak ditangani dengan baik pada usia 23 bulan atau 1000 hari pertama kelahiran bayi [1]. Diantara gangguan-gangguan yang dapat disebabkan seperti menurunkan produktivitas, mengganggu tingkat kecerdasan, dan rentan terhadap penyakit. *Stunting* berdampak besar pada perkembangan jangka pendek balita, seperti gangguan perkembangan kognitif, motorik, dan verbal sehingga tidak menjadi optimal [1]. Dampak *stunting* menjadi lebih tinggi terhadap anak-anak di masa mendatang sehingga mudah mengalami obesitas, rentan terhadap penyakit, serta kapasitas belajar dan performa anak menjadi tidak optimal. Selain itu dampak buruk *stunting* bisa juga berimbas pada kesehatan reproduksi.

Menurut Kementerian Kesehatan Republik Indonesia, hasil Survei Status Gizi Indonesia mengumumkan bahwa prevalensi *stunting* di Indonesia pada tahun 2022 mencapai 21.6%, tetapi angka ini masih berada diatas standar Organisasi Kesehatan Dunia (WHO) dengan prevalensi *stunting* dibawah 20% [2]. Sementara itu di Riau, prevalensi *stunting* Kota Pekanbaru sebesar 16.8% yaitu sebanyak 12.166 balita berdasarkan data e-PPGBM Agustus 2022 [3]. Salah satu kampung yang mendukung percepatan penurunan *stunting* yaitu Kampung KB Tunas Harapan terletak di Kecamatan Tuah Madani Kota Pekanbaru. Kampung ini merupakan kampung yang gencar dalam pengentasan *stunting* dengan salah satu inovasi masyarakat berupa Cemara *Stunting* (Celengan Masyarakat Cegah *Stunting*) dengan pemberian makanan bergizi kepada Ibu Hamil, baduta, balita, dan ibu setelah melahirkan [3].

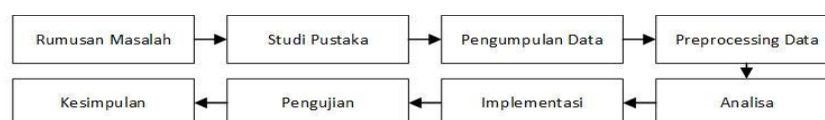
Salah satu solusi untuk mengetahui bagaimana perhitungan kuatnya hubungan antar kriteria keluarga beresiko *stunting* yang perlu diprioritaskan adalah dengan menghitung menggunakan algoritma apriori. Algoritma apriori merupakan salah satu algoritma yang melakukan pencarian *frequent itemset* dengan menggunakan teknik *association rule*. *Association rule* melakukan dua tahap dalam proses miningnya, yaitu membangkitkan *large itemsets*, dan membentuk *rule* asosiasi. Pada tahapan pembentukan *large itemsets* merupakan tahapan yang paling berat karena dalam prosesnya dibutuhkan *scanning* data yang berkali-kali dan pengolahan data yang sangat besar. Oleh karena itu algoritma apriori dapat dilakukan *improvement* dengan menggunakan teknik *hash-based* yang dapat membuat efisiensi dalam pembangkitan *large itemsets* dan efektif dalam mereduksi jumlah transaksi dari database.

Dalam melakukan penelitian ini, ada beberapa penelitian terkait yang dijadikan penulis sebagai referensi terkait topik permasalahan yang diangkat. Penelitian yang dilakukan oleh Ulva Rizky Ananda dan timnya pada tahun 2021 [4], penelitian yang dilakukan oleh Agung Triyudi dengan timnya pada tahun 2022 [5], penelitian yang dilakukan Ike Septi Nindyaa dan timnya pada tahun 20223 [6], dan yang terakhir penelitian yang dilakukan oleh Manuel Wilson dengan timnya pada tahun 2023 [7].

Berdasarkan paparan masalah di atas, penelitian ini bertujuan untuk mencari hubungan asosiasi kriteria keluarga beresiko *stunting* dengan menggunakan *Improve Apriori Algorithm* / algoritma apriori yang telah dioptimasi menggunakan *hash-based techniques* untuk mendapatkan hasil yang lebih efisien, penelitian ini juga bertujuan untuk mengimplementasikan algoritma apriori dengan teknik *hash-based* kedalam sebuah sistem informasi berbentuk aplikasi web. Oleh karena itulah fokus pada penelitian ini adalah pengimplementasian *association rule* menggunakan algoritma *improve apriori* dengan teknik *hash-based* tersebut kedalam sebuah sistem informasi.

2. METODE PENELITIAN

Proses tahapan yang perlu dilakukan adalah sebagai tahapan yang terdapat pada Gambar 1.



Gambar 1. Tahapan Metode Penelitian

a. Rumusan masalah

Tahap awal awal penelitian ini mencari permasalahan yang akan diteliti dimana rumusan masalah penelitian ini mengenai implementasi algoritma *improve apriori* dengan teknik *hash-based* kedalam sebuah sistem informasi web, dengan tujuan mendapatkan *association rule* dari kriteria keluarga beresiko *stunting* di Kecamatan Tuah Madani.

b. Studi pustaka

Pada tahapan ini penulis mendefinisikan permasalahan serta mengumpulkan literatur atau penelitian yang relevan. Penulis mempelajari teori-teori atau metode-metode yang berkaitan dengan *stunting*, algoritma apriori, teknik *hash-based*, dan improvisasi dari kedua metode tersebut.

c. Pengumpulan data

Pada tahapan ini pengumpulan data dilakukan dengan mengambil data Balai Penyuluhan KB di Kecamatan Tuah Madani. Data yang digunakan adalah data sekunder dari pendataan keluarga tahun 2021 dari Balai Penyuluhan KB Kecamatan Tuah Madani. 6 dari 20 parameter untuk menilai keluarga beresiko *stunting* ditunjukkan pada Tabel 1 sebagai berikut.

Tabel 1. Data Keluarga Beresiko *Stunting* Kecamatan Tuah Madani 2021

Data Ke-	Kode keluarga	NIK Kepala Keluarga	Nama Kepala Keluarga	Keluarga memiliki anak baduta	Keluarga memiliki anak balita	...	Kategori Keluarga beresiko <i>stunting</i>
1	147...	321...	A... Mun..	X	X	...	X
2	147...	140...	A.. Musli..	V	V	...	V
3	147...	140...	A.. Sup..	X	X	...	X
4	147...	147...	Ab..	X	V	...	V
5	147...	147...	Aba.. Pel..	X	X	...	V
...
11107	147...	147...	Zun..	V	V	...	X

Data asli yang terdapat pada penelitian ini berjumlah 20 parameter diantaranya data-data berkaitan dengan kode keluarga, NIK kepala keluarga, nama kepala keluarga, punya anak baduta (0-23 bulan), punya anak balita (24-59 bulan), PUS (pasangan usia subur), PUS hamil, ada anak 7-15 tahun tidak sekolah, tidak ada anggota keluarga memiliki sumber penghasilan untuk memenuhi kebutuhan pokok per bulan, jenis lantai tanah, tidak setiap anggota keluarga makan ”makanan beragam” paling sedikit 2 (dua) kali sehari, keluarga pra sejahtera, keluarga tidak mempunyai sumber air minum utama yang layak, keluarga tidak punya rumah layak huni, pendidikan terakhir ibu di bawah SLTP, terlalu muda (umur istri < 20 tahun), terlalu tua (umur istri > 35 tahun), terlalu dekat (< 2 tahun), terlalu banyak (≥ 3 anak), dan kategori keluarga berpotensi risiko *stunting*.

d. Preprocessing data

Pada tahapan ini, data yang telah dikumpulkan akan dipersiapkan dan dibersihkan untuk digunakan dalam analisis menggunakan *improve apriori algorithm*. Tahapan ini melibatkan berbagai proses seperti penghapusan data yang tidak relevan, penanganan missing value, dan transformasi data ke dalam format yang sesuai untuk dianalisis. Tujuan dari *preprocessing* data adalah untuk memastikan bahwa data yang akan dianalisis berkualitas tinggi dan siap untuk digunakan dalam penerapan algoritma.

e. Analisa

Pada tahapan ini, terdapat analisa algoritma dan juga analisa perancangan terhadap data yang telah dipreprocessing. Data dianalisis untuk menemukan pola-pola dan insight yang relevan. Peneliti akan menggunakan algoritma apriori dengan kombinasi teknik *hash-based* untuk menemukan *association rule* yang dapat mengungkapkan pola-pola kriteria *stunting*. Analisis ini bertujuan untuk mendapatkan pemahaman yang lebih dalam tentang data dan mendukung temuan penelitian.

f. Implementasi

Algoritma yang digunakan yaitu algoritma apriori dengan kombinasi teknik *hash-based*. Tahap ini peneliti akan merancang dan mengimplementasikan algoritma kedalam sistem informasi berbasis web yang telah dirancang untuk mendapatkan *association rule* yang relevan dengan permasalahan penelitian.

g. Pengujian

Pada tahapan ini dilakukan evaluasi terhadap implementasi sistem dan juga algoritma yang digunakan. Pengujian ini bertujuan untuk memastikan bahwa sistem berjalan sesuai fungsi yang dirancang dan juga sesuai dengan kinerja algoritma yang diterapkan.

h. Kesimpulan

Tahapan ini merangkum seluruh proses dari penelitian yang dapat digunakan untuk menjawab tujuan dari penelitian.

3. HASIL DAN PEMBAHASAN

3.1. Preprocessing Data

Preprocessing data yang dilakukan yaitu *cleaning data*, merubah *missing value*, dan juga melakukan transformasi data pada dataset yang akan diolah.

3.1.1. Cleaning data

Pada tahap ini, parameter yang tidak diperlukan dalam proses asosiasi dibersihkan atau difilter. Parameter yang dihapus yaitu kode keluarga, nomor induk kependudukan, nama kepala keluarga, dan kategori keluarga. Untuk data keluarga yang tidak beresiko terkena *stunting* juga akan dilakukan proses *cleaning* karena data yang dibutuhkan hanya keluarga yang beresiko *stunting* untuk mencari hubungan yang kuat antara parameter dalam proses asosiasi, sehingga parameter yang digunakan dalam penelitian ini berjumlah 17 parameter. Hasil *cleaning data* pada Tabel 2 berikut.

Tabel 2. Hasil Cleaning Data

Data ke-	Punya anak baduta	Punya anak balita	PUS	PUS Hamil	Anak 7-15 tahun tidak sekolah	Tidak ada sumber penghasilan	Jenis lantai tanah	...	Terlalu banyak anak (≥3 anak)
1	V	V	V	X	X	X	X	...	X
2	X	V	V	X	X	X	X	...	X
3	X	X	V	X	X	X	X	...	X
4	V	X	V	X	X	X	X	...	V
5	X	X	V	X	X	X	X	...	X
...
5000	X	X	V	X	X	X	X	...	V

3.1.2. Missing value

Nilai *missing value* digantikan dengan **X**, yang dianggap mewakili kondisi **TIDAK ADA**. Pendekatan ini sesuai dengan prinsip imputasi konservatif, di mana nilai hilang digantikan dengan nilai yang paling aman untuk menghindari bias besar. Schafer dan Graham menyarankan penggunaan imputasi konservatif untuk mencegah distorsi hasil analisis, dengan mengganti *missing value* dengan nilai yang tidak memperkenalkan kesalahan interpretasi, seperti **X** untuk data binomial [8].

3.1.3. Transformasi data

Untuk mempermudah proses data pada website menggunakan algoritma *improve* apriori, maka nilai kelas dari paratemeternya yang berupa X dan V diubah menjadi bentuk numerik. Transformasi data dapat dilihat pada Tabel 3 berikut.

Tabel 3. Hasil Transformasi Data

Data ke-	Punya anak baduta	Punya anak balita	PUS	PUS Hamil	Anak 7-15 tahun tidak sekolah	Tidak ada sumber penghasilan	Jenis lantai tanah	...	Terlalu banyak anak (≥3 anak)
1	1	1	1	0	0	0	0	...	0
2	0	1	1	0	0	0	0	...	0
3	0	0	1	0	0	0	0	...	0
4	1	0	1	0	0	0	0	...	1
5	0	0	1	0	0	0	0	...	0
...
5000	0	0	1	0	0	0	0	...	1

3.2. Analisa

Terdapat dua analisa yang akan dilakukan untuk penelitian ini yaitu analisa algoritma yang digunakan dan juga analisa perancangan dari sistem informasi.

3.2.1. Analisa Algoritma

a. Algoritma Apriori

Apriori merupakan algoritma yang sangat sederhana untuk menemukan frequent itemset dari transaksi database yang besar. Namanya berasal dari fakta bahwa algoritma ini menggunakan pengetahuan sebelumnya dari frequent itemset untuk proses iterasi berikutnya. Algoritma Apriori adalah salah satu algoritma yang melakukan pencarian frekuensi itemset dengan menggunakan teknik *association rule*. Algoritma Apriori menggunakan pengetahuan frekuensi atribut yang telah diketahui sebelumnya untuk memproses informasi selanjutnya. Pada algoritma Apriori menentukan kandidat yang mungkin muncul dengan cara memperhatikan minimum *support* dan minimum *confidence*. *Support* adalah nilai pengujung atau persentase kombinasi sebuah item dalam database[9].

Adapun dua tolak ukur dalam bentuk *rules* atau aturan dalam penerapan algoritma apriori adalah sebagai berikut [10]:

Pertama, *Support* atau bisa juga disebut nilai penunjang adalah persentase dari laporan atau record yang didalamnya mengandung kombinasi item. Persamaan (1) adalah rumus untuk mendapatkan nilai *support*.

$$Support(A) = \frac{Jumlah\ Transaksi\ Mengandung\ A}{Total\ Transaksi} \tag{1}$$

Persamaan (2) adalah rumus untuk mendapatkan nilai *support* dari suatu kombinasi item.

$$Support(A, B) = \frac{Jumlah\ Transaksi\ Mengandung\ A\ dan\ B}{Total\ Transaksi} \tag{2}$$

Kedua, *Confidence* atau biasa disebut nilai kepastian adalah Kuatnya hubungan antar item dalam aturan asosiasi. Adapun rumus untuk mendapatkan nilai *confidence* ialah:

$$Confidence (A \Rightarrow B) = \frac{Support(A,B)}{Support(A)} \times 100\% \quad (3)$$

Langkah-langkah pada proses algoritma apriori adalah sebagai berikut:

Langkah pertama, *scan database* guna menemukan kandidat 1-itemset (C1) dan juga menghitung nilai *support*-nya. Setelah itu bandingkan antara nilai *support* dengan *minimum support* yang sebelumnya telah ditentukan, apabila nilai *support* lebih besar atau nilainya sama dengan *minimum support*, *itemset* terhitung dalam *large-itemset* set 1 (L1). **Langkah kedua**, *Itemset* yang tidak terhitung dalam *large-itemset* tidak dipakai untuk melakukan iterasi berikutnya. (Proses *pruning*). **Langkah ketiga**, *Large-itemset* set 1 (L1) digunakan untuk proses iterasi yang berikutnya. Pada *large-itemset* set 1 (L1) dilakukan proses join pada dirinya sendiri untuk menghasilkan kandidat 2-itemset (C2). Setelah itu bandingkan nilai *support* dari semua item yang ada pada C2 dengan *minimum support*, jika nilainya lebih atau sama dengan *minimum support* maka akan masuk kedalam *large-itemset* L2. Ulangi langkah yang sama seperti mencari *large-itemset* yang sebelumnya. **Langkah keempat**, Pembentukan kandidat (*joining*) dan pembentukan *large-itemset* (*Pruning*) dilakukan secara terus-menerus sampai tidak ada lagi kandidat yang bisa terbentuk. **Langkah kelima**, yaitu untuk semua *large-itemset* yang terbentuk atau memenuhi nilai *minimum support* akan dibentuk *association rule* setelah itu dicari juga nilai *confidence*-nya. Nantinya seluruh aturan yang terbentuk jika nilai *confidence*-nya kurang dari nilai *minimum confidence* yang ditetapkan, maka aturan tersebut tidak akan dipakai atau tidak termasuk dalam *association rule* yang dipakai.

b. Teknik Hash-Based

Teknik hashing digunakan untuk meningkatkan efisiensi algoritma apriori. Teknik ini bekerja dengan membuat *dictionary* (tabel hash) yang menyimpan kandidat item set sebagai *key*, dan jumlah kemunculannya sebagai nilai. Inisialisasi dimulai dengan nol dan bertambah untuk setiap set item yang berada didalam data. Algoritma *Hash Based* menggunakan teknik *hashing* untuk menyaring keluar *itemset* yang tidak penting untuk pembangkitan *itemset* selanjutnya. Ketika *support count* untuk kandidat *k-itemset* dihitung dengan menelusuri basis data, algoritma *hash based* mengumpulkan informasi mengenai (k+1)-*itemset* dengan cara seluruh kemungkinan (k+1)-*itemset* dihash ke dalam *hash table* dengan menggunakan fungsi *hash* (menggunakan sebuah bilangan prima untuk operasi *modulo*)[4].

Algoritma *Hash Based* terbagi menjadi tiga bagian utama yang masing-masing bagian melakukan proses yang berbeda.

Bagian pertama, akan menghasilkan kandidat 1-itemset yang disebut C1 dan *large 1-itemset* yang disebut L1 dari basis data. Untuk kandidat 1-itemset, seluruh transaksi ditelusuri untuk menghitung *support count* dari *itemset* ini. Pada tahap ini *hash tree* untuk C1 dibangun dengan tujuan mengefisienkan penghitungan *support count*. Pada bagian ini juga algoritma akan membangun *hash table* (dengan fungsi *hash*) untuk 2 *itemset* yang akan berguna mengurangi banyaknya kandidat 2-itemset C2. **Bagian kedua**, kumpulan kandidat *itemset* Ck dibangkitkan berdasarkan *hash table* yang telah dibuat pada iterasi sebelumnya. Lalu ditentukan *large itemset* Lk dan mengurangi ukuran basis data untuk pembangkitan *itemset* selanjutnya. Bagian algoritma ini terbagi menjadi dua fase. Fase pertama untuk membangkitkan kandidat *k-itemset* berdasarkan *hash table*. Fase kedua akan menghitung *support* pada kandidat *itemset* dan mengurangi ukuran dari setiap transaksi. **Bagian ketiga**, sama seperti bagian kedua tetapi tidak menggunakan *hash table* sehingga mirip dengan algoritma apriori. Bagian kedua dilakukan selama nilai *hash* buket lebih besar dari *minimum support*. Setelah batasan ini terlewati, algoritma *hash based* diganti dengan algoritma apriori.

Tujuan dari teknik *hash-based* ini sebagai efisiensi dalam pembangkitan *large itemsets* dan efektif dalam mereduksi jumlah transaksi dari database. Algoritma *hash-based* efektif digunakan tepatnya pada *large itemsets* (L2) keatas. Kandidat *itemsets* (L) yang dibangkitkan lebih kecil dibandingkan dengan teknik sebelumnya yang menggunakan algoritma apriori. Berikut penjelasan dari rumus *hash table* (*hash bucket*):

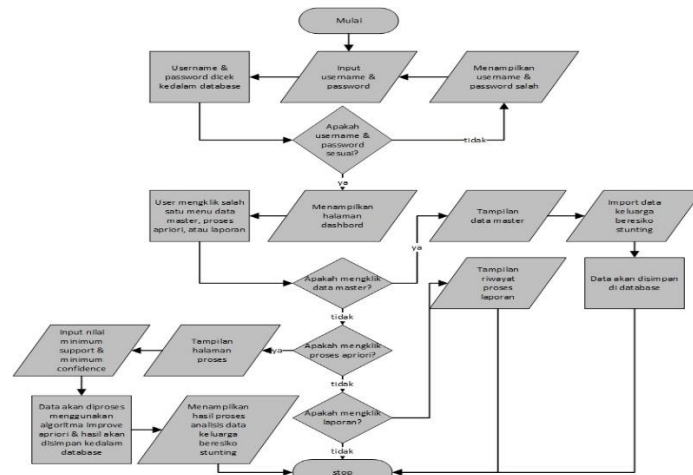
$$H(X, Y) = [(order\ of\ X) \times\ penambahan\ ctr\ hash\ table + (order\ of\ y)]\ mod\ prima \quad (4)$$

dengan *Order of X* adalah perwakilan nilai X, *Penambahan ctr hash table* adalah nilai modulus bilangan prima yang apabila terjadi *collision* nilai tersebut ditambah 1 (+ 1) sampai tidak terjadi *collision*, *Order of Y* adalah perwakilan nilai Y, dan *Prima* adalah bilangan prima yang terdekat dan yang lebih besar dari jumlah kombinasi 2-itemset (C2).

3.2.2. Analisa Perancangan

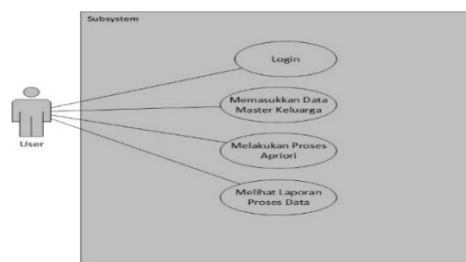
a. Flowchart Aplikasi

Dalam perancangan sistem informasi web untuk implementasi algoritma *improve apriori* dengan teknik *hash-based* pada penelitian ini, penulis membutuhkan diagram alir atau *flowchart* bagaimana sistem akan berjalan. *Flowchart* dapat dilihat pada Gambar 2. *Flowchart* pada Gambar 2 yang disajikan dalam penelitian ini memberikan gambaran visual alur kerja dari sebuah aplikasi yang dikembangkan untuk membantu dalam pengelolaan data dan analisis risiko keluarga yang berisiko *stunting*. Alur kerja ini mencakup beberapa fitur utama seperti autentikasi pengguna, pengelolaan data master, analisis data menggunakan algoritma *improve apriori*, dan pembuatan laporan.



Gambar 2. Flowchart sistem

b. Use Case Diagram



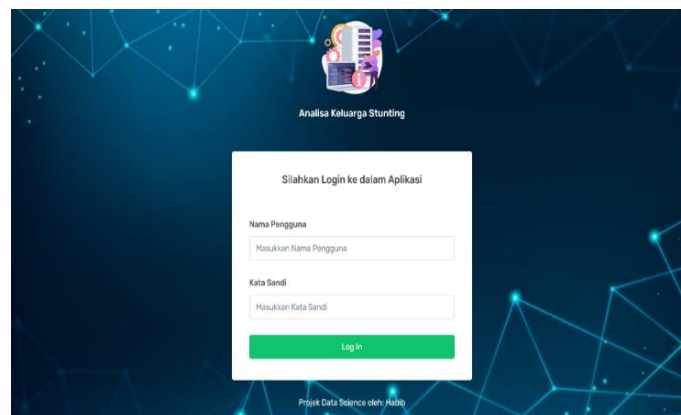
Gambar 3. Use Case Diagram

Pada Gambar 3 menjelaskan *use case* dari website yang dirancang hanya dapat diakses oleh satu *user*. Saat membuka website *user* akan dihadapkan langsung ke tampilan *login* untuk dapat mengakses menu-menu lainnya seperti data master, proses apriori, dan laporan. Pertama, pada bagian Data Master, pengguna dapat melihat dan mengelola data keluarga berisiko *stunting* serta memasukkan data baru melalui input yang disediakan. Selanjutnya, pada bagian Proses Apriori, pengguna dapat melakukan analisis menggunakan algoritma Apriori yang telah diimprovisasi dengan mengatur nilai *minimum support* dan *minimum confidence* untuk memproses data keluarga tersebut. Terakhir, pada bagian Laporan, aplikasi ini menyediakan tampilan yang menampilkan informasi terkait riwayat laporan dari proses analisis yang telah dilakukan sebelumnya, memberikan kemudahan untuk melacak hasil analisis dan progres yang telah dicapai.

3.3. Hasil Implementasi

Implementasi dilakukan setelah tahapan analisa kebutuhan sistem dilakukan sesuai dengan *flowchart* sistem yang dirancang. Berikut merupakan tampilan *interface* hasil dari implementasi ke sistem informasi web menggunakan *improve apriori algorithm*.

3.3.1. Halaman Login

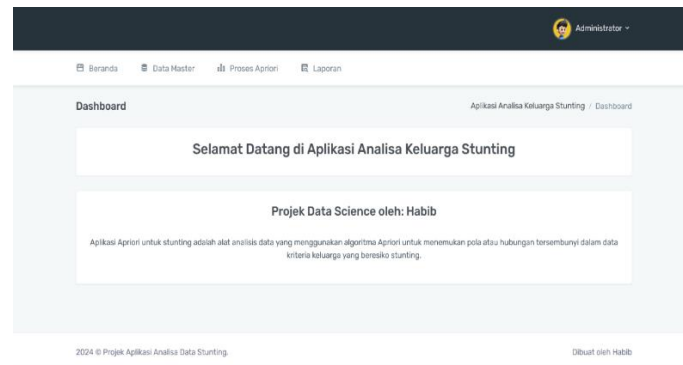


Gambar 4. Halaman Login

Gambar 4 merupakan tampilan halaman login untuk masuk kedalam sistem analisa keluarga *stunting*. Pada halaman ini hanya terdapat form untuk mengisi nama pengguna, kata sandi dan 1 *button* untuk melakukan proses login.

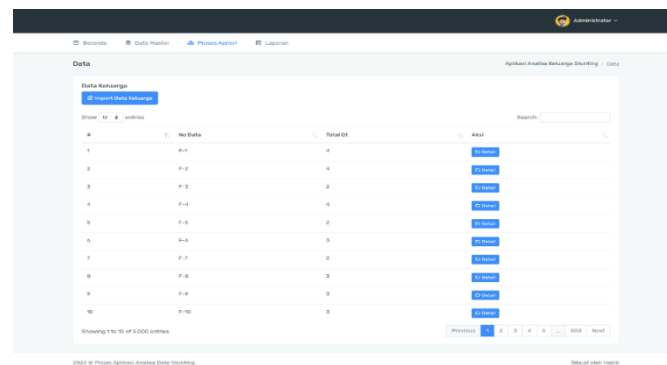
3.3.2. Halaman Beranda/ Dashboard

Halaman dashboard merupakan tampilan utama yang akan terlihat oleh pengguna segera setelah berhasil melakukan proses login ke dalam sistem. Tampilan ini dapat dilihat pada Gambar 5 yang menunjukkan interface awal dari aplikasi. Setelah pengguna berhasil melakukan *login*, mereka akan diarahkan ke halaman dashboard, yang berfungsi sebagai pusat navigasi utama dalam aplikasi diantaranya menu beranda, data master, proses apriori, dan laporan.



Gambar 5. Halaman Beranda/Dashboard

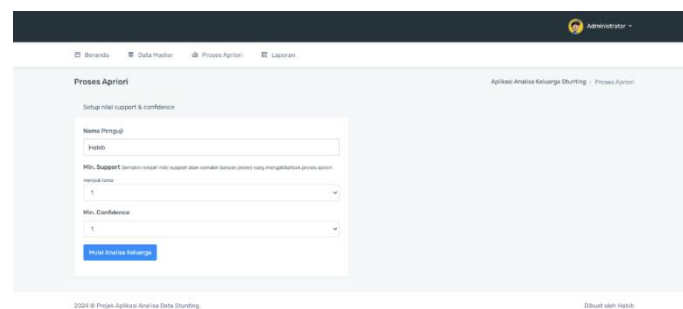
3.3.3. Halaman Data Master



Gambar 6. Halaman Data Master

Gambar 6 menunjukkan tampilan halaman Data Master yang berisi data keluarga dalam aplikasi Analisa Keluarga *Stunting*. Halaman ini berfungsi sebagai pusat pengelolaan data keluarga yang berpotensi berisiko *stunting*, dimana data keluarga akan dianalisis lebih lanjut berdasarkan kriteria yang relevan. Pada halaman data master terdapat 1 *button* yang berfungsi untuk melakukan *import* data yang berformat excel kedalam sistem. Kemudian pada setiap item data keluarga beresiko *stunting* terdapat 1 *button* yaitu detail untuk menampilkan kriteria keluarga.

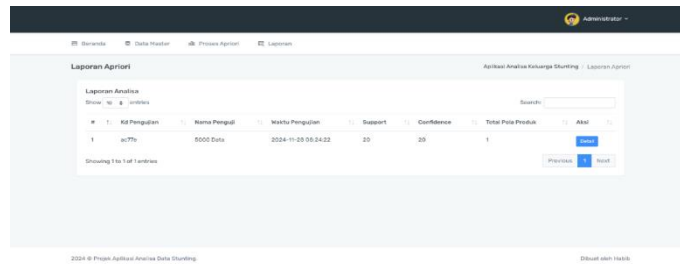
3.3.4. Halaman Proses Apriori



Gambar 7. Halaman Proses Apriori

Gambar 7 menunjukkan tampilan halaman Proses Apriori dalam aplikasi Analisa Keluarga *Stunting*. Halaman ini muncul ketika pengguna memilih menu Proses Apriori pada navbar di bagian atas aplikasi. Pada halaman ini, pengguna dapat mengatur parameter yang diperlukan untuk memulai analisis data menggunakan algoritma *Apriori* yang telah ditingkatkan (*improved Apriori algorithm*). Terdapat 3 form input untuk nama penguji, *minimum support*, dan *minimum confidence* serta 1 *button* mulai analisa keluarga yang berfungsi untuk memulai proses analisa keluarga beresiko *stunting* dengan algoritma *improve apriori*.

3.3.5. Halaman Laporan



Gambar 8. Halaman Laporan

Gambar 8 menunjukkan tampilan halaman Laporan dalam aplikasi Analisa Keluarga *Stunting*. Halaman ini menampilkan riwayat laporan analisis algoritma *Improved Apriori* yang telah dijalankan sebelumnya untuk keluarga yang berisiko *stunting*. Terdapat 1 *button* detail yang berfungsi untuk melihat riwayat analisa sesuai dengan *minimum support* dan *confidence*.

3.4. Hasil Analisa

Pada percobaan pertama analisa untuk *improve apriori alogrithm* pada keluarga beresiko *stunting* dimasukkan data sekitar 5000 keluarga dengan beragam kriteria sesuai dengan parameter yang ditentukan. *Minimum support* dan *minimum confidence* yang diinputkan sebesar 20% masing-masingnya sehingga diperoleh hasil sebagai berikut.

Tabel 4. Tabel Nilai *Support*

No	Kriteria	Frekuensi	Support	Persen
1	Punya Anak Baduta (0-23 Bulan)	377	377/5000 = 0.0754	7.54%
2	Punya Anak Balita (24-59 Bulan)	939	939/5000 = 0.1878	18.78%
3	PUS	4984	4984/5000 = 0.9968	99.68%
4	PUS Hamil	198	198/5000 = 0.0396	3.96%
5	Ada Anak 7-15 Tahun Tidak Sekolah	327	327/5000 = 0.0654	6.54%
6	Tidak Ada Keluarga Memiliki Penghasilan Untuk Memenuhi Kebutuhan Pokok PerBulan	183	183/5000 = 0.0366	3.66%
7	Jenis Lantai Tanah	4	4/5000 = 0.0008	0.08%
8	Tidak Setiap Anggota Keluarga Makan Makanan Beragam Paling Sedikit 2 (Dua) Kali Sehari	65	65/5000 = 0.013	1.3%
9	Keluarga Pra Sejahtera	530	530/5000 = 0.106	10.6%
10	Keluarga Tidak Mempunyai Sumber Air Minum Utama Yang Layak	5	5/5000 = 0.001	0.1%
11	Keluarga Tidak Mempunyai Jamban Yang Layak	33	33/5000 = 0.0066	0.66%
12	Keluarga Tidak Mempunyai Rumah Layak Huni	129	129/5000 = 0.0258	2.58%
13	Pendidikan Terakhir Ibu Di Bawah SLTP	373	373/5000 = 0.0746	7.46%
14	Terlalu Muda (Umur Istri < 20 Tahun)	14	14/5000 = 0.0028	0.28%
15	Terlalu Tua (Umur Istri > 35 Tahun)	4079	4079/5000 = 0.8158	81.58%
16	Terlalu Dekat (< 2 Tahun)	292	292/5000 = 0.0584	5.84%
17	Terlalu Banyak Anak (>= 3 Anak)	2257	2257/5000 = 0.4514	45.14%

Dari total 17 kategori kriteria keluarga beresiko *stunting*, hanya ada tiga kategori yang memenuhi syarat *minimum support* sebesar 20% berdasarkan Tabel 4. Data menunjukkan bahwa kategori Pasangan Usia Subur (PUS) memiliki *support* sebesar 99,68%, yang berarti dari 5000 transaksi, sebanyak 4984 transaksi melibatkan keluarga dengan pasangan usia subur. Hal ini menunjukkan bahwa hampir seluruh data mencakup kategori PUS. Kategori Terlalu Tua (Umur Istri > 35 Tahun) memiliki *support* sebesar 81,58%, yang mengindikasikan bahwa 4079 transaksi terkait dengan keluarga di mana usia istri lebih dari 35 tahun. Sementara itu, kategori Terlalu Banyak Anak (≥ 3 Anak) tercatat dengan *support* sebesar 45,14%, yang menunjukkan adanya 2257 transaksi yang melibatkan keluarga dengan tiga anak atau lebih. Dari data ini, terlihat bahwa tiga kategori tersebut mendominasi dan memenuhi syarat *minimum support* 20%, yang dapat diartikan sebagai faktor atau kondisi yang umum dijumpai dalam data transaksi yang dianalisis. Data yang memenuhi *support* selanjutnya diberikan kode order *address* yang akan digunakan dalam mencari *hash address* di setiap iterasinya seperti pada Tabel 5.

Tabel 5. Data yang memenuhi *support* 20%

No	Kriteria	Support	Address
1	PUS	99.68%	1
2	Terlalu Tua (Umur Istri > 35 Tahun)	81.58%	2
3	Terlalu Banyak Anak (≥ 3 Anak)	45.14%	3

Selanjutnya, dilakukan perhitungan kandidat 2-itemset berdasarkan data yang memenuhi *minimum support*, menggunakan teknik *hash-based* untuk pencarian dan penyimpanan alamat hash dari setiap kombinasi itemset 2. Dalam proses ini, *hash address* setiap pasangan itemset dihitung menggunakan formula yang sudah ditentukan, yaitu Rumus (4). Teknik *hash-based* ini bertujuan untuk memetakan pasangan item ke alamat tertentu dalam tabel hash, sehingga dapat mengelompokkan dan memproses kandidat 2-itemset dengan lebih efisien dibandingkan memproses keseluruhan data secara langsung. Namun, pada proses *hashing* ini, ada kemungkinan terjadi kolisi (*collision*), yaitu kondisi di mana dua atau lebih kombinasi 2-itemset berbeda memiliki alamat hash yang sama. Kolisi dapat menyebabkan informasi yang tidak akurat, karena beberapa kombinasi mungkin tersimpan dalam alamat yang sama, mengaburkan pencatatan frekuensi atau keterhubungan itemset yang seharusnya unik.

Tabel 6. Tabel Hash 2-itemset

No	Kombinasi 2-itemset	Perhitungan	Hash Address
1	PUS & Terlalu tua (umur istri > 35 tahun)	$[(1)*5+(2)] \text{ mod } 5$	2
2	PUS & Terlalu banyak anak (≥ 3 anak)	$[(1)*5+(3)] \text{ mod } 5$	3
3	Terlalu tua (umur istri > 35 tahun) & Terlalu banyak anak (≥ 3 anak)	$[(2)*5+(3)] \text{ mod } 5$	3

Seperti yang terlihat pada Tabel 6, jika ditemukan kolisi dalam tabel hash address (*hash address collision* pada nilai 3), maka proses hashing harus diulang dengan menggunakan teknik atau metode penanganan kolisi dengan menambahkan ctr *hash* serta merubah bilangan modulus. Proses perulangan ini bertujuan agar tidak ada kolisi yang tersisa, sehingga setiap kombinasi item dapat dipetakan ke *hash address* yang berbeda dan unik. Dengan demikian, hasil hashing ini akan memberikan tabel hash address yang bersih dan akurat, yang selanjutnya dapat digunakan untuk seleksi kandidat 2-itemset yang memenuhi *minimum support* tanpa risiko kesalahan data akibat kolisi. Sehingga didapat perhitungan sebagai berikut.

Tabel 7. Iterasi 2 Tabel Hash 2-itemset

No	Kombinasi 2-itemset	Perhitungan	Hash Address
1	PUS & Terlalu tua (umur istri > 35 tahun)	$[(1)*6+(2)] \text{ mod } 7$	1
2	PUS & Terlalu banyak anak (≥ 3 anak)	$[(1)*6+(3)] \text{ mod } 7$	2
3	Terlalu tua (umur istri > 35 tahun) & Terlalu banyak anak (≥ 3 anak)	$[(2)*6+(3)] \text{ mod } 7$	1

Pada Tabel , masih terdapat *collision* pada hasil *address* perhitungan *hashing*. Maka langkah selanjutnya ialah seperti yang dilakukan sebelumnya dengan menambahkan ctr *hash* dan juga merubah bilangan modulusnya. Sehingga didapat perhitungan sebagai berikut.

Tabel 8. Iterasi 3 Tabel Hash 2-Itemset

No	Kombinasi 2-itemset	Perhitungan	Hasil
1	PUS & Terlalu tua (umur istri > 35 tahun)	$[(1)*7+(2)] \text{ mod } 11$	9
2	PUS & Terlalu banyak anak (≥ 3 anak)	$[(1)*7+(3)] \text{ mod } 11$	10
3	Terlalu tua (umur istri > 35 tahun) & Terlalu banyak anak (≥ 3 anak)	$[(2)*7+(3)] \text{ mod } 11$	6

Setelah tidak terdapat lagi kolisi seperti pada Tabel 8, langkah selanjutnya adalah menyusun *hash address* dan menghitung jumlah transaksi (*basket count*) untuk kriteria keluarga berisiko *stunting*. Setelah itu, perhitungan nilai *support* dilakukan dengan kriteria minimal sebesar 20%. Dari Tabel 8, diperoleh 1 kombinasi untuk 3-itemset yang memenuhi nilai *minimum support* (*minsupp*) seperti pada Tabel 9. Teknik *hash-based* yang diterapkan memungkinkan proses penentuan kombinasi item yang mendukung nilai *minimum support* ini menjadi lebih efisien.

Tabel 9. Confidence kombinasi 3-itemset

No	Kombinasi 3-itemset	Frekuensi	Support	Confidence
1	Terlalu tua (umur istri > 35 tahun); terlalu banyak anak (≥ 3 anak); dan PUS	1877	$1877/5000 * 100 = 37,54\%$	$1877/2257 * 100 = 83,16\%$

Hasil penerapan algoritma *Improve Apriori* yang ditingkatkan dengan teknik *hash-based* menghasilkan tiga item kriteria yang menjadi prioritas utama dalam penanganan keluarga berisiko *stunting*. Berdasarkan aturan asosiasi yang ditemukan, keluarga yang memiliki karakteristik berikut berisiko tinggi: (1) **istri berusia lebih dari 35 tahun**, (2) **memiliki tiga anak atau lebih**, dan (3) merupakan **Pasangan Usia Subur (PUS)**. Aturan asosiasi ini memiliki nilai *support* dan *confidence* sebesar 83,16%, yang menunjukkan kekuatan hubungan antar item tersebut dalam konteks risiko *stunting*.

3.5. Pengujian

3.5.1. Pengujian Black Box

Pengujian blackbox digunakan untuk mengevaluasi fungsi aplikasi yang telah dibuat. Metode ini menguji apakah setiap fitur berfungsi sesuai spesifikasi dengan memasukkan input tertentu dan memeriksa keluaran yang dihasilkan. Pengujian ini penting untuk memastikan bahwa aplikasi memenuhi kebutuhan dan spesifikasi yang telah ditentukan. Berikut hasil pengujian *Black Box*.

Tabel 10. Black Box Testing

No	Pengujian	Kegiatan	Hasil Yang diharapkan	Hasil
1	Login	Input nama pengguna dan kata sandi	Dapat masuk ke Halaman Dashboard.	Sesuai
2	Data Master	Pilih menu Data Master	Dapat menampilkan Halaman Data Master	Sesuai
3	Import data keluarga	Masukkan data keluarga berisiko <i>stunting</i> berbentuk excel	Data dapat masuk kedalam database dan ditampilkan di halaman Data Master	Sesuai
4	Detail data master	Pilih tombol detail pada tabel data master	Menampilkan kriteria keluarga	Sesuai
5	Proses Apriori	Pilih menu Proses Apriori	Dapat menampilkan Halaman Proses Apriori	Sesuai
6	Mulai Analisa Keluarga	Input nama penguji, <i>minimum support</i> dan <i>minimum confidence</i>	Dapat menampilkan hasil proses analisa algoritma <i>improve apriori hash-based</i> sesuai dengan inputan <i>minimum support</i> dan <i>confidence</i>	Sesuai
7	Laporan	Pilih menu Laporan	Dapat menampilkan tabel riwayat laporan hasil analisis algoritma apriori sebelumnya	Sesuai
8	Detail laporan	Pilih menu detail pada tabel laporan	Dapat menampilkan riwayat hasil proses analisa algoritma <i>improve apriori</i> sebelumnya sesuai dengan <i>minimum support</i> dan <i>confidence</i>	Sesuai

3.5.2. Pengujian Validitas Aturan

Uji validitas aturan adalah proses untuk mengevaluasi apakah aturan asosiasi yang dihasilkan oleh algoritma seperti Apriori memang relevan, signifikan, dan dapat diandalkan. Uji ini bertujuan untuk memastikan bahwa aturan yang ditemukan mencerminkan hubungan yang bermakna dalam dataset, bukan hanya pola yang muncul secara kebetulan. Metrik yang digunakan dalam uji validitas aturan yaitu *support*, *confidence*, dan *lift*. *Lift* berfungsi untuk mengukur hubungan antara *antecedent* (kondisi awal) dan *consequent* (hasil yang diharapkan) dibandingkan dengan probabilitas kejadian mereka secara independen. Persamaan (5) merupakan rumus untuk mendapatkan nilai *lift*.

$$Lift(A \rightarrow B) = \frac{Confidence \text{ dari } A \rightarrow B}{Support \text{ dari } B} \tag{5}$$

dengan nilai uji lift lebih dari 1 (*lift ratio* > 1) memberikan pernyataan aturan asosiasi valid untuk digunakan sebagai acuan [11].

Tabel 11. Uji Validitas Aturan

No	Rule	Support (A)	Support (B)	Support (A→B)	Confidence	Lift ratio
1	Terlalu tua (umur istri > 35 tahun) → Terlalu banyak anak (>= 3 anak)	81.58	45.14	37.54	37.54/81.58 * 100 = 46.01%	46.01/45.14 = 1.02
2	PUS → Terlalu tua (umur istri > 35 tahun)	99.68	81.58	81.5	81.5/99.68 * 100 = 81.76%	81.76/81.58 = 1.00
3	PUS → Terlalu banyak anak (>= 3 anak)	99.68	45.14	45.14	45.14/99.68 * 100 = 45.28%	45.28/45.14 = 1.00
4	Terlalu tua (umur istri > 35 tahun); Terlalu banyak anak (>= 3 anak) → PUS	37.54	99.68	37.54	83.16%	83.16/99.68 = 0.83

Aturan pertama, Terlalu Tua (Umur Istri > 35 Tahun) → Terlalu Banyak Anak (≥ 3 Anak), jarang terjadi dengan *support* 37,54% dan *confidence* 46,01%, namun memiliki lift tinggi yang menunjukkan hubungan statistik yang kuat. PUS → Terlalu Tua (Umur Istri > 35 Tahun) lebih sering terjadi dengan *support* 81,5% dan *confidence* 81,76%, meskipun nilai lift yang rendah menunjukkan hubungan yang kurang signifikan. Sementara itu, aturan PUS → Terlalu Banyak Anak (≥ 3 Anak) relevan dengan *support* 45,14% dan *confidence* 45,28%, namun juga menunjukkan hubungan yang tidak terlalu signifikan secara statistik. Terakhir, aturan Terlalu Tua (Umur Istri > 35 Tahun); Terlalu Banyak Anak (≥ 3 Anak) → PUS menunjukkan pengaruh antara kedua kondisi ini terhadap kategori PUS. Aturan ini lebih cukup relevan berdasarkan *support* 37.54% serta valid dengan *confidence* 83.16%, namun nilai lift pada aturan ini menunjukkan bahwa hubungan aturan ini relatif kurang signifikan.

4. KESIMPULAN

Penelitian ini berhasil menerapkan algoritma *Improve Apriori* yang dioptimasi dengan teknik *hash-based* untuk mengidentifikasi pola risiko *stunting* pada keluarga. Optimasi ini meningkatkan efisiensi dengan mengurangi kombinasi itemset yang tidak relevan, dan mempercepat analisis. Hasil penelitian menunjukkan tiga kriteria utama yang signifikan terhadap risiko *stunting*: usia ibu di atas 35 tahun, status PUS, dan jumlah anak lebih dari tiga, dengan *support* 37,54% dan *confidence* 83,16%. Pola ini mengindikasikan hubungan kuat antara faktor-faktor tersebut dan risiko *stunting*, yang dapat menjadi prioritas dalam program pencegahan *stunting*. Selain itu, penerapan algoritma ini dalam sistem informasi berbasis web memudahkan analisis risiko secara praktis, mendukung pengambilan kebijakan berbasis data, dan mempercepat program pengentasan *stunting*, terutama di wilayah berisiko tinggi. Penelitian ini memberikan kontribusi yang efisien dan aplikatif untuk analisis risiko kesehatan masyarakat dan intervensi *stunting* yang lebih tepat sasaran. Untuk penelitian selanjutnya, disarankan untuk memperluas cakupan data dengan melibatkan lebih banyak wilayah dan periode waktu yang berbeda guna meningkatkan generalisasi hasil, serta menambah variabel seperti status gizi ibu atau tingkat pendidikan kepala keluarga untuk memberikan wawasan lebih mendalam dalam memahami pola risiko *stunting*.

5. DAFTAR PUSTAKA

[1] Ruswati et al., "Risiko Penyebab Kejadian Stunting pada Anak," *J. Pengabd. Kesehat. Masy. Pengmaskesmas*, vol. 1, no. 2, pp. 34–38, 2021.

[2] SSGI, "Hasil Survei Status Gizi Indonesia (SSGI) 2022," *Kementeri. Kesehat. Republik Indones.*, pp. 77–77, 2023, [Online]. Available: <https://promkes.kemkes.go.id/materi-hasil-survei-status-gizi-indonesia-ssgi-2022>

[3] Novrizaldi, "Menko PMK : Penanganan Stunting Harus Berkelanjutan," 2023. [Online]. Available: <https://www.kemenkopmk.go.id/menko-pmk-penanganan-stunting-harus-berkelanjutan#:~:text=Kemudian%2C menurut SSGI 2022%2C prevalensi,di bawah rata-rata nasional.>

[4] U. R. Amanda and D. P. Utomo, "Penerapan Data Mining Algoritma Hash Based Pada Data Pemesanan Buah Impor Cv. Green Uni Fruit," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 5, no. 1, pp. 86–93, 2021, doi: 10.30865/komik.v5i1.3653.

[5] A. Triayudi and Sumiati, "Penerapan Algoritma Hash Based dalam Penemuan Aturan Asosiasi Penjualan Tanaman Hias," *Build. Informatics, Technol. Sci.*, vol. 4, no. 3, pp. 1293–1300, 2022, doi: 10.47065/bits.v4i3.2626.

[6] I. S. Nindyya, Gusmelia Testiana, and Irfan Dwi Jaya, "Implementasi Algoritma Apriori dan ECLAT (Equivalence Class Transformation) Pada Data Transaksi Penjualan," *J. CoSciTech (Computer Sci. Inf. Technol.)*, vol. 4, no. 2, pp. 525–533, 2023, doi: 10.37859/coscitech.v4i2.5444.

[7] M. Wilson, M. S. Nair, P. P. Nair, and M. Anusree, "A perfect hashing to enhance the performance of Apriori algorithm," *2023 2nd Int. Conf. Electr. Electron. Inf. Commun. Technol. ICEEICT 2023*, no. June, pp. 1–6, 2023, doi: 10.1109/ICEEICT56924.2023.10157902.

[8] J. L. Scahfer and J. W. Graham, "Missing data: our view of state of the art," *Psychol. Methods*, vol. 7, no. 2, p. 147, 2002, doi: <https://doi.org/10.1037/1082-989X.7.2.147>.

[9] S. Saefudin and S. DN, "Penerapan Data Mining Dengan Metode Algoritma Apriori Untuk Menentukan Pola Pembelian Ikan," *JSiI (Jurnal Sist. Informasi)*, vol. 6, no. 2, p. 36, 2019, doi: 10.30656/jsii.v6i2.1587.

[10] N. L. Utami, A. Nazir, E. Budianita, and F. Insani, "Jurnal Computer Science and Information Technology (CoSciTech) algoritma apriori," vol. 5, no. 1, pp. 75–83, 2024.

[11] Faishal Zoelfiandi and Utomo Budiayanto, "Penerapan Data Mining Menggunakan Algoritma Apriori Pada Toko Adelia Frozen Food," *J. Ticom Technol. Inf. Commun.*, vol. 11, no. 1, pp. 13–19, 2022, doi: 10.70309/ticom.v11i1.65.