

**IMPLEMENTASI ALGORITMA APRIORI PADA PENYEBAB
KEMATIAN BAYI**



TUGAS AKHIR

*Disusun dalam rangka memenuhi salah satu persyaratan
Untuk menyelesaikan program Strata-1 Departemen Teknik Informatika
Fakultas Teknik Universitas Hasanuddin
Makassar*

Disusun Oleh :

RYA DITA PURNAMA

D42116002

**DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
MAKASSAR
2021**

SKRIPSI
IMPLEMENTASI ALGORITMA APRIORI PADA PENYEBAB
KEMATIAN BAYI

Disusun dan diajukan oleh

RYA DITA PURNAMA

D421 16 002



DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
MAKASSAR
2021

LEMBAR PENGESAHAN SKRIPSI
IMPLEMENTASI ALGORITMA APRIORI PADA PENYEBAB
KEMATIAN BAYI

Disusun dan diajukan oleh

RYA DITA PURNAMA

D42116002

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin pada tanggal 22 November 2021 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,



Dr. Ir. Ingrid Nurtanio, MT
Nip. 196108131988112001

Elly Warni, ST., MT.
Nip. 198202162008122001

Ketua Program Studi,



Dr. Anis Ahmad Iham, S.T., M.IT.
Nip. 197310101998021002

PERNYATAAN KEASLIAN

Yang bertanda tangan di bahwa ini :

Name : Rya Dita Purnama
NIM : D42116002
Program Studi : Teknik Informatika
Jenjang : S1

Menyatakan dengan ini bahwa hasil karya tulisan saya berjudul :

Implementasi Algoritma Apriori pada Penyebab Kematian Bayi

Adalah karya tulisan saya sendiri. Bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi ini yang saya tulis ini benar benar merupakan hasil karya saya sendiri.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain. Maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 22 November 2021

Yang menyatakan,



RYA DITA PURNAMA

KATA PENGANTAR

Assalamu Alaikum Wr. Wb.

Puji dan syukur penulis panjatkan atas kehadiran Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini yang berjudul **“Implementasi Algoritma Apriori Pada Penyebab Kematian Bayi”** sebagai salah satu syarat dalam menyelesaikan jenjang Strata Satu (S1) pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa dalam penyusunan dan penulisan skripsi ini tidak lepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan Tugas Akhir. Oleh karena itu, dengan segala kerendahan hati, penulis mengucapkan rasa terima kasih sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa atas semua berkat, karunia serta pertolongan-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Orang tua penulis, Bapak M. Tahir dan Ibu Rostina S. Pd yang selalu memberikan motivasi, dukungan, doa, semangat, dan kasih sayang serta selalu sabar dalam mendidik penulis sejak kecil.
3. Ibu Dr. Ir. Ingrid Nurtanio, M.T selaku dosen pembimbing I dan Ibu Elly Warni., S.T., M.T selaku dosen pembimbing II yang selalu menyediakan waktu, tenaga dan pikirannya yang luar biasa untuk mengarahkan penulis dalam penyusunan Tugas Akhir ini.
4. Bapak Dr. Amil Ahmad Ilham, S.T., M.I.T., selaku Ketua Departemen Teknik Informatika Fakultas Teknik Univeristas Hasanuddin atas bimbingannya selama masa perkuliahan penulis.
5. Segenap Dosen dan Staff Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah banyak membantu penulis selama masa perkuliahan.
6. Sahabat-sahabat TRICHE, Fabyola Larasati Masyita, Afifah Ilham, Ismayanti, Nishrina Nurul Amirah, dan Patricia Viola Palada yang selalu menjadi tempat berbagi cerita dan memberikan dukungan dan bantuan kepada penulis.
7. Teman-teman IGNITER16 atas dukungan dan semangat yang telah diberikan

8. Orang-orang berpengaruh lainnya yang tidak sempat penulis sebutkan satu persatu yang telah memberikan dukungan, semangat dan doanya selama penyusunan Tugas Akhir ini.

Akhir kata, penulis menyadari masih terdapat kekurangan dalam penyusunan laporan ini baik isi maupun cara penyajian. Oleh karena itu penyusun mengharapkan adanya saran dan kritik yang bersifat membangun demi kesempurnaan Tugas Akhir ini. Penulis berharap semoga tugas akhir ini dapat memberikan hal yang bermanfaat dan menambah wawasan bagi pembaca dan khususnya bagi penulis juga.

Makassar, November 2021

Penulis

(Rya Dita Purnama)

ABSTRAK

Angka Kematian Bayi (AKB) merupakan salah satu indikator kesejahteraan suatu bangsa yang mencerminkan tingkat masalah kesehatan masyarakat dan dapat menggambarkan status kesehatan penduduk secara umum. Angka Kematian Bayi (AKB) adalah kematian anak usia kurang dari satu tahun.

Berdasarkan data dari RSIA Sitti Khadijah 1 Muhammadiyah Cabang Makassar, RSIA Sitti Fatimah Makassar, RSIA Bahagia, dan RSUD Labuang Baji tahun 2014 s.d 2021, terdapat beberapa indikator penyebab yang menyebabkan kematian pada bayi. Dengan beberapa indikator penyebab kematian tersebut, maka akan dilakukan asosiasi untuk mengetahui keterkaitan antar faktor penyebab pada kematian bayi. Salah satu teknologi yang dapat digunakan adalah *Data Mining* dengan metode *Association Rule*.

Pada penelitian ini algoritma dari metode *Association Rule* yang diterapkan adalah algoritma apriori. Faktor penyebab yang menjadi fitur dalam penelitian ini yaitu sianosis, kelainan kongenital, sepsis, hygroma colli, asfiksia, BBLR (Berat Badan Lahir Rendah), prematur, placenta pravia, RDS (*Respiratory Distress Syndrom*), dan pendarahan.

Hasil dari penelitian ini menunjukkan bahwa penggunaan algoritma apriori yang diujicobakan pada *dataset* penyebab kematian pada bayi dari RSIA Sitti Khadijah 1 Muhammadiyah Cabang Makassar, RSIA Sitti Fatimah Makassar, RSIA Bahagia, dan RSUD Labuang Baji tahun 2014 s.d 2021, dengan jumlah data yang diolah sebanyak 969 kejadian, dengan menggunakan nilai *minimum support* = 0.75 diperoleh panjang kombinasi item yang terbentuk sebanyak *2-itemset*. Dari *large itemset* yang diperoleh proses penemuan aturan asosiasi dilakukan dengan menerapkan *minimum confidence* = 0.95 maka didapatkan 1 aturan asosiasi. Aturan asosiasi yang didapatkan yaitu jika terjadi kematian bayi akibat Sianosis maka berhubungan dengan penyebab kematian Kelainan Kongenital dengan keyakinan sebesar 95%. Nilai *lift ratio* pada hasil penelitian ini lebih dari 1, yaitu 1.17 yang menunjukkan bahwa hubungan dari aturan asosiasi bisa dikatakan kuat.

Kata Kunci : *data mining*, algoritma *apriori*, pola keterkaitan, penyebab kematian bayi

DAFTAR ISI

KATA PENGANTAR.....	i
ABSTRAK	iii
DAFTAR ISI	iv
DAFTAR GAMBAR.....	vii
DAFTAR TABEL	ix
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Tujuan Penelitian	2
1.4. Manfaat Penelitian	2
1.5. Batasan Masalah	3
1.6. Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA.....	5
2.1. <i>Data Mining</i>	5
2.2. Assosiasi	7
2.3. Algoritma Apriori	8
2.4. <i>Lift Ratio</i>	10
2.5. Penelitian Terkait.....	11
BAB III METODOLOGI PENELITIAN	13
3.1. Studi Literatur	13
3.2. Pengumpulan Data	14
3.3. Perancangan Sistem	14
3.3.1. <i>Data Collection</i>	14
3.3.2. <i>Data Cleaning</i>	16

3.3.3. <i>Data Integration</i>	16
3.3.4. <i>Data Selection & Transformation</i>	17
3.3.5. <i>Data Mining</i>	18
3.3.6. <i>Pattern Evaluation</i>	18
3.3.7. <i>Knowledge Presentation</i>	18
3.4. Implementasi Sistem.....	19
3.5. Evaluasi Sistem.....	21
BAB IV HASIL DAN PEMBAHASAN	22
4.1. Penerapan Algoritma Apriori dengan Menggunakan Program Python	22
4.1.1. <i>Dataset</i> Penyebab Kematian pada Bayi dari Dua RSIA di Makassar.....	22
4.1.2. Pembentukan Kandidat <i>1-Item</i> (C1).....	24
4.1.3. Menentukan <i>Large 1-Itemset</i> (L1)	28
4.1.4. Menentukan Kandidat <i>2-Itemset</i> (C2).....	29
4.1.5. Menentukan <i>Large 2-Itemset</i> (L2)	31
4.1.6. Aturan Asosiasi	34
4.1.7. Lift Ratio	36
4.2. Penerapan Algoritma Apriori dengan Menggunakan <i>Library</i> Python.....	36
4.2.1. <i>Library</i> Python	36
4.2.2. Aturan Asosiasi	36
4.2.3. Lift Ratio	37
4.3. Pembahasan	37
4.4. Pengembangan Aplikasi berbasis Website	40
BAB V PENUTUP	42
5.1. Kesimpulan	42
5.2. Saran	42

DAFTAR PUSTAKA.....	43
LAMPIRAN	45

DAFTAR GAMBAR

Gambar 2. 1. Proses <i>data mining</i> untuk menemukan pengetahuan.....	7
Gambar 3.1. Tahapan penelitian.....	13
Gambar 3. 2. Contoh Data RSIA Sitti Khadijah 1 Muhammadiyah Cabang Makassar.....	15
Gambar 3. 3. Contoh Data RSIA Sitti Fatimah Makassar.....	15
Gambar 3. 4. Contoh Data RSIA Bahagia.....	15
Gambar 3. 5. Contoh Data RSUD Labuang Baji.....	16
Gambar 3. 6. Penggabungan Data	17
Gambar 4. 1. Contoh <i>Dataset</i> Penyebab Kematian pada Bayi.....	23
Gambar 4. 2. <i>Kode Program input dataset</i>	23
Gambar 4. 3. <i>Dataset</i> setelah dilakukan <i>cleaning data</i>	24
Gambar 4. 4. <i>Kode Program cleaning data</i>	24
Gambar 4. 5. Hasil Kandidat <i>1-Item (C1)</i>	25
Gambar 4. 6. Kode Python untuk Pembentukan Kandidat <i>1-Item (C1)</i>	25
Gambar 4. 7. Frekuensi <i>Item</i>	26
Gambar 4. 8. Kode Python untuk menghitung Frekuensi <i>Item</i>	26
Gambar 4. 9. Nilai <i>Support Itemset</i>	27
Gambar 4. 10. Kode python untuk menghitung nilai <i>support item</i>	28
Gambar 4. 11. <i>Large 1-Itemset (L1)</i>	28
Gambar 4. 12. <i>Kode python</i> untuk mendapatkan <i>Large 1-Itemset (L1)</i>	29
Gambar 4. 13. Kandidat <i>2-Itemset (C2)</i>	29
Gambar 4. 14. Frekuensi Kandidat <i>2-Itemset (C2)</i>	30
Gambar 4. 15. Nilai <i>support</i> Kandidat <i>2-Itemset (C2)</i>	30
Gambar 4. 16. Kode python untuk menghasilkan Kandidat <i>k-itemset (Ck)</i>	30
Gambar 4. 17. Kode python untuk menghitung Frekuensi <i>Itemset</i>	31
Gambar 4. 18. Kode python untuk menghitung Nilai <i>suppport itemset</i>	31
Gambar 4. 19. Hasil <i>Large 1-itemset (L2)</i>	32
Gambar 4. 20. Kode python untuk menentukan <i>Large Itemset (Lk)</i>	32
Gambar 4. 21. Kode python untuk menghitung frekuensi <i>Large Itemset (Lk)</i>	33
Gambar 4. 22. Kode python untuk menghitung nilai <i>support Large Itemset (Lk)</i>	33

Gambar 4. 23. Kode python untuk menampilkan Kandidat <i>k-itemset</i> (Ck), frekuensi Ck, nilai <i>support</i> Ck, <i>Large k-itemset</i> (Lk), frekuensi Lk, dan nilai <i>support</i> Lk.....	34
Gambar 4. 24. Aturan Asosiasi yang dihasilkan	35
Gambar 4. 25. Kode program python untuk menghasilkan aturan asosiasi	35
Gambar 4. 26. Nilai Lift Ratio.....	36
Gambar 4. 27. Kode program python menggunakan <i>library efficient_apriori</i>	36
Gambar 4. 28. <i>Large itemset</i> dari <i>library</i> apriori	37
Gambar 4. 29. Aturan Asosiasi dari <i>library</i> apriori.....	37
Gambar 4. 30. <i>Lift Ratio</i>	37
Gambar 4. 31. Kode program python untuk menghitung <i>confidence</i> dan <i>lift ratio</i>	37
Gambar 4. 32. Algoritma <i>Fp-Growth</i>	39
Gambar 4. 33. Tampilan Aplikasi Website	41

DAFTAR TABEL

Tabel 4. 1. Aturan Asosiasi dari program <i>non-library apriori</i> dan <i>library apriori</i>	39
Tabel 4. 2. Perbandingan waktu komputasi antara algoritma Apriori dan algoritma <i>FP-Growth</i>	40

BAB I

PENDAHULUAN

1.1. Latar Belakang

Angka Kematian Bayi (AKB) merupakan salah satu indikator kesejahteraan suatu bangsa yang mencerminkan tingkat masalah kesehatan masyarakat dan dapat menggambarkan status kesehatan penduduk secara umum (Budiati, 2016). Bayi juga merupakan investasi Sumber Daya Manusia sebab mereka adalah generasi penerus yang akan meneruskan pembangunan bangsa.

Angka Kematian Bayi (AKB) adalah jumlah kematian anak berusia di bawah dari 1 tahun per 1000 kelahiran. Berdasarkan evaluasi *Millennium Development Goals* (MDGs) pada tahun 2015, kasus kematian bayi di Indonesia adalah 305 per 100.000 kelahiran. Sedangkan target yang dicanangkan Perserikatan Bangsa-Bangsa (PBB) ialah 102 per 100.000 kelahiran. Dari jumlah kasus tersebut, Indonesia ditempatkan sebagai negara dengan angka kematian tertinggi kedua di Asia Tenggara.

Untuk mengurangi jumlah kasus kematian bayi, maka pemerintah ikut andil dalam melakukan upaya untuk meminimalisir jumlah kasus kematian bayi. Berdasarkan Rencana Pembangunan Jangka Menengah Nasional (RPJMN) 2015-2019 menargetkan penurunan angka kematian bayi 32 per 1.000 kelahiran hidup menjadi 24 di tahun 2019. Selain itu, sehubungan dengan salah satu target *Sustainable Development Goals* (SDGs) di bidang kesehatan dan kesejahteraan diantaranya adalah mengakhiri kematian bayi.

Di Sulawesi Selatan sendiri tercatat 664 kasus kematian bayi pada tahun 2018 dari 8.6 juta penduduk berdasarkan data dari Dinas Kesehatan Sulawesi Selatan. Makassar sebagai ibu kota dari Provinsi Sulawesi Selatan ikut menambahkan jumlah kasus kematian bayi. Angka kematian bayi di Kota Makassar pada tahun 2013 sebesar 6,71 per 1000 kelahiran hidup, dengan jumlah kasus 165 kematian dari 24.576 kelahiran hidup (M & Thaha, n.d.).

Secara garis besar, kematian bayi terdiri dari dua macam yaitu Endogen dan Eksogen. Kematian bayi endogen atau biasa disebut dengan kematian neonatal yaitu kematian bayi yang terjadi pada bulan pertama setelah dilahirkan yang disebabkan oleh faktor-faktor yang dibawa anak sejak lahir, yang diperoleh dari orang tuanya pada saat konsepsi atau didapat selama kehamilan. Sedangkan kematian bayi eksogen atau kematian *post* neonatal, adalah kematian

bayi yang terjadi setelah usia satu bulan sampai menjelang usia satu tahun yang disebabkan oleh faktor-faktor yang bertalian dengan pengaruh lingkungan luar (Budiati, 2016).

Berdasarkan data dari RSIA Sitti Khadijah 1 Muhammadiyah Cabang Makassar, RSIA Sitti Fatimah Makassar, RSIA Bahagia, dan RSUD Labuang Baji tahun 2014 s.d 2021, terdapat beberapa penyebab-penyebab kematian yang dapat menyebabkan kematian pada bayi. Dari beberapa penyebab tersebut, dapat dilakukan assosiasi atau pola hubungan untuk mengetahui keterkaitan antara penyebab kematian sehingga dapat dilakukan pencegahan terhadap faktor penyebab kematian itu sendiri. Faktor penyebab kematian antara lain sianosis, kelainan kongenital, sepsis, hygroma colli, asfiksia, BBLR (Berat Badan Lahir Rendah), prematur, placenta pravia, RDS (*Respiratory Distress Syndrom*), pendarahan.

Data mining dilihat dari sisi teknik pengolahan data menyediakan sejumlah algoritma yang dapat digunakan untuk menggali informasi tersembunyi dari kumpulan data yang multidimensi. Dengan menerapkan metode *Association Rule* (aturan asosiasi) dengan Algoritma Apriori yang akan melakukan penelusuran pada data historis untuk mengidentifikasi pola data yang didasarkan pada sifat-sifat yang teridentifikasi sebelumnya (Istiasih, 2017).

Informasi dari hasil penelitian diharapkan dapat membantu dalam pengambilan keputusan untuk meminimalisir kematian bayi berdasarkan pada pola keterkaitan dari faktor penyebab kematian bayi.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah, dapat dirumuskan permasalahan adalah “Bagaimana mengetahui keterkaitan pada penyebab Kematian Bayi menggunakan Algoritma Apriori?”

1.3. Tujuan Penelitian

Adapun tujuan dari penelitian ini ialah Mengetahui keterkaitan pada penyebab Kematian Bayi menggunakan Algoritma Apriori.

1.4. Manfaat Penelitian

Adapun manfaat dari penelitian ini ialah:

- 1.) Bagi pemerintah, penelitian ini dapat digunakan sebagai pendukung pengambilan keputusan untuk meminimalisir kematian bayi.
- 2.) Bagi ilmu pengetahuan, dapat memberikan suatu karya penelitian baru yang dapat mendukung pengembangan sistem informasi.

1.5. Batasan Masalah

Untuk menghindari kesalahpahaman dan meluasnya pembahasan, maka penulis memberikan batasan masalah sebagai berikut:

- 1.) Data yang digunakan adalah data kematian bayi di RSIA Sitti Khadijah 1 Muhammadiyah Cabang Makassar, RSIA Sitti Fatimah Makassar, RSIA Bahagia, dan RSUD Labuang Baji tahun 2014 s.d 2021.
- 2.) Fitur yang digunakan yaitu sianosis, kelainan kongenital, sepsis, hygroma colli, asfiksia, BBLR, prematur, placenta previa. RDS, pendarahan.
- 3.) Menggunakan bahasa pemrograman Python.

1.6. Sistematika Penulisan

Untuk memberikan gambaran singkat mengenai isi tulisan ini, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu :

BAB I PENDAHULUAN

Bab ini menguraikan latar belakang diangkatnya judul penelitian implementasi algoritma apriori pada penyebab kematian bayi, disertai dengan rumusan masalah, tujuan dan manfaat penelitian, batasan masalah, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi teori-teori yang berhubungan dengan penelitian termasuk *data mining*, *association*, dan algoritma yang digunakan.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang perencanaan dan penerapan algoritma sehingga menghasilkan aturan asosiasi.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil penelitian data dan pembahasan terkait pengolahan data yang telah dilakukan.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan serta saran-saran untuk pengembangan lebih lanjut.

BAB II TINJAUAN PUSTAKA

2.1. *Data Mining*

Data mining merupakan aktivitas yang berkaitan dengan pengumpulan data, pemakaian data historis untuk menemukan pengetahuan, informasi, pola atau hubungan dalam data yang berukuran besar. *Output data mining* dapat digunakan sebagai alternatif dalam pengambilan keputusan atau untuk memperbaiki pengambilan keputusan di masa yang akan datang (Eferi Buulolo, 2020).

Data mining sebagai proses menemukan struktur data yang menarik menggunakan satu atau beberapa algoritma untuk mengidentifikasi *trend* atau pola menarik dalam suatu data. *Knowledge* yang diperoleh dari model *data mining* akan digenerelasikan yang tujuannya dapat diterapkan pada situasi yang baru (Roiger, 2017).

Data mining atau kadang disebut juga *Knowledge Discovery in Database (KDD)* menggunakan metode teknik data mining untuk mengekstraksi pengetahuan dan informasi potensial dari sekumpulan data besar. Pengetahuan dan informasi yang dihasilkan dari KDD bersifat baru dan mudah dimengerti serta bermanfaat.

Proses *data mining* merupakan serangkaian proses yang terdiri dari langkah-langkah sebagai berikut (Zamroni, Werdiningsih, & Purbandini, 2017):

1) *Collection*

Proses *collection data* dilakukan dengan mengumpulkan data yang bertujuan untuk menghasilkan data mentah (*raw data*) yang akan diolah menggunakan teknik data mining. Pada tahap ini, informasi yang akan diperoleh seperti jumlah *data* yang digunakan, *raw data*, dan fitur-fitur dari *data*.

2) *Cleaning / Selection*

Pada tahap ini data yang sudah dipilih akan dilakukan pembersihan. Proses *cleaning* bertujuan untuk menghilangkan *noise* dan *data-data* yang tidak konsisten. *Cleaning* juga

dapat dilakukan proses memperkaya data dengan menambah informasi lain yang relevan yang disebut dengan istilah *enrichment*.

Sedangkan, proses *selection* adalah dengan melakukan pemilihan data yang akan digunakan karena tidak semua data yang ada dapat dipergunakan. Aktivitas pemilihan data meliputi pembuatan kumpulan data target, penentuan fitur, pemilihan sampel data dan penyimpanan data pada sebuah berkas.

3) *Integration*

Data integration merupakan proses penggabungan data yang berasal dari beberapa sumber data. Integritas data dilakukan pada atribut-atribut yang mengidentifikasi entitas-entitas yang unik seperti nama, nomor pelanggan, dan lainnya. *Data* yang berbeda memiliki konvensi penamaan fitur yang berbeda, sehingga akan menyebabkan redundansi data. Pembersihan data dapat dilakukan untuk menghilangkan redundansi data.

4) *Transformation*

Pada tahap ini, *data-data* yang telah terpilih akan di-*transformasi* ke dalam bentuk yang sesuai untuk prosedur *data mining* dengan cara melakukan normalisasi atau agregasi data.

5) *Data Mining*

Data mining merupakan tahapan utama dalam *Knowledge Discovery in Database (KDD)*. *Data mining* adalah proses penggalian dan pencarian pengetahuan dan informasi yang bermanfaat dengan menggunakan algoritma/metode/teknik tertentu sesuai dengan pengetahuan atau informasi yang dicari.

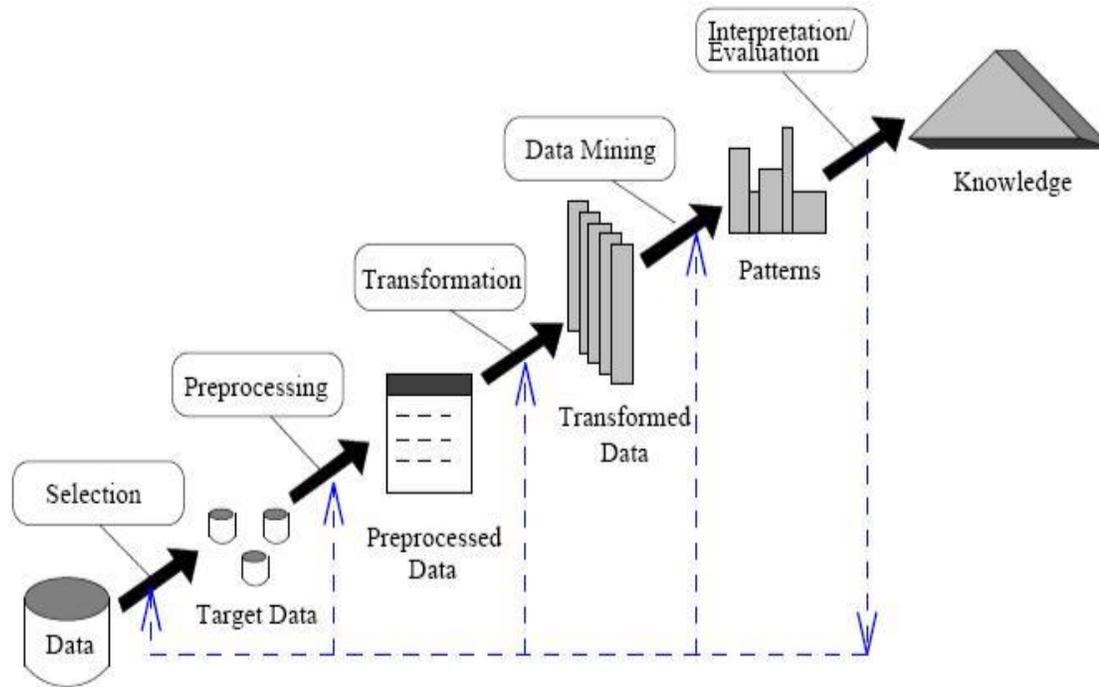
6) *Pattern Evaluation*

Pengetahuan atau informasi yang dihasilkan dari proses *data mining* akan dipresentasikan atau ditampilkan ke dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan seperti informasi ditampilkan dalam bentuk grafik, pohon keputusan, ataupun dalam bentuk *rule*.

7) *Knowledge Presentation*

Tujuan utama proses *Knowledge Discovery in Database (KDD)* adalah memperoleh pengetahuan atau informasi yang bermanfaat dan mudah dimengerti. *Knowledge presentation* merupakan visualisasi dan penyajian pengetahuan mengenai metode yang digunakan untuk memperoleh pengetahuan yang diperoleh pengguna. Tahap terakhir

dari proses *data mining* adalah bagaimana memformulasikan keputusan atau aksi dari hasil analisis yang didapat. Ada kalanya hal ini harus melibatkan orang-orang yang tidak memahami data mining. Karenanya presentasi hasil data mining dalam bentuk pengetahuan yang bisa dipahami semua orang adalah satu tahapan yang diperlukan dalam proses data mining. Dalam presentasi ini, visualisasi juga bisa membantu mengkomunikasikan hasil data mining.



Gambar 2. 1. Proses *data mining* untuk menemukan pengetahuan

2.2. Asosiasi

Asosiasi berguna untuk mengungkap hubungan yang menarik yang tersembunyi dalam dataset besar. Hubungan yang terungkap tersebut dapat dipresentasikan dalam bentuk aturan asosiasi (*association rules*) atau himpunan *item* yang sering muncul (*sets of frequent items*) (Fricles Ariwisanto Santuri, 2019). Aturan asosiasi merupakan salah satu metode yang bertujuan mencari pola yang sering muncul di antara banyak transaksi, dimana setiap transaksi terdiri dari beberapa *item* (Azwar Anas, 2018). Aturan asosiasi sering digunakan dalam menganalisis keranjang belanja. Pengetahuan yang dihasilkan dari aturan asosiasi berbentuk

“*if... then...*” atau “jika...maka...”. Bisa di umpamakan jika seseorang membeli barang A maka akan membeli barang B pula dalam satu transaksi.

Sebagai contoh aturan asosiasi dari analisa pembelian di sebuah supermarket, dapat diketahui berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan membeli susu. Dengan pengetahuan ini, pemilik supermarket dapat mengatur penempatan barang-barangnya dan merancang strategi pemasaran dengan pengetahuan yang diperoleh dari hasil asosiasi tersebut.

Menurut Daniel T. Larose (2005), aturan asosiasi dari *database* besar terdiri dari dua langkah, yaitu (Mandala, 2017):

- 1) Temukan semua *frequent itemset*, yaitu menemukan semua *itemset* dengan frekuensi \geq *minimum support*.
- 2) Dari *frequent itemset*, buat aturan asosiasi yang memenuhi kondisi *minimum support* dan *minimum confidence*.

Pencarian pola kaidah asosiasi menggunakan dua buah parameter nilai yaitu dukungan (*support*) dan keterpercayaan (*confidence*) yang memiliki rentang nilai antara 0%-100% (Putra, Raharjo, Sandi, Ridwan, & Prasetyo, 2019).

2.3. Algoritma Apriori

Algoritma apriori adalah algoritma pengembalian data dengan aturan asosiasi (*Association Rules*) untuk menentukan hubungan asosiasi suatu kombinasi *item*. Algoritma apriori merupakan algoritma yang paling populer dikenal sebagai dengan paradigma *generate and test*, yaitu pembuatan kandidat kombinasi *item* yang mungkin berdasar aturan tertentu lalu diuji apakah kombinasi item tersebut memenuhi syarat *support minimum* (Malik, Azis, Supianto, & Setiawan, 2019).

Menurut Tampubolon (2013), beberapa istilah yang digunakan dalam algoritma apriori antara lain (Tommy, 2017):

- 1) *Support* atau dukungan : probabilitas atribut muncul secara bersamaan dari seluruh transaksi. *Support* untuk aturan “X→Y” adalah probabilitas atribut atau kumpulan atribut X dan Y yang terjadi bersamaan.
- 2) *Confidence* atau tingkat kepercayaan : probabilitas kejadian atribut terjadi bersamaan dimana salah satu atribut sudah pasti terjadi.
- 3) *Minimum Support* : parameter yang digunakan sebagai batasan frekuensi kejadian atau *support count* yang harus dipenuhi suatu kelompok data untuk dapat dijadikan aturan.
- 4) *Minimum Confidence* : parameter yang mendefinisikan minimum level dari *confidence* yang harus dipenuhi oleh aturan yang berkualitas.
- 5) *Itemset* : kelompok atribut
- 6) Kandidat *Itemset* (C_k) : *item-item* yang akan dihitung *support count*-nya.
- 7) *Frequent Itemset* (F_k) : *itemset* yang sering terjadi, atau *itemset- itemset* yang sudah melewati batas *minimum support* yang telah ditentukan.

Algoritma apriori dinyatakan dengan algoritma paling terkenal untuk menemukan pola frekuensi tinggi. Pola frekuensi tinggi adalah pola-pola item di dalam suatu *database* yang memiliki frekuensi atau *support* diatas ambang batas tertentu atau dikenal dengan *Minimum Support* (Fricles Ariwisanto Santuri, 2019). Metode algoritma apriori terdiri dari dua tahap, yaitu :

1) Analisa Pola Frekuensi Tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam *database*. Rumus untuk mendapatkan nilai *support* adalah sebagai berikut :

$$Support(A) = \frac{\text{jumlah kejadian mengandung } A}{\text{total kejadian}} \quad (2.1)$$

Sementara nilai *support* untuk 2 *itemset* diperoleh dengan menggunakan rumus sebagai berikut :

$$Support(A, B) = \frac{\text{jumlah kejadian mengandung } A \text{ dan } B}{\text{total kejadian}} \quad (2.2)$$

Untuk menghitung *support* untuk 3 *itemset*, dilakukan dengan rumus sebagai berikut :

$$Support(A, B, C) = \frac{\text{jumlah kejadian mengandung } A, B, \text{ dan } C}{\text{total kejadian}} \quad (2.3)$$

2) Pembentukan Aturan Asosiasi

Akurasi dari suatu aturan asosiasi disebut *confidence*. *Confidence* disebut dengan nilai kepastian adalah kuatnya hubungan antar *item* dalam aturan asosiasi. Setelah semua pola frekuensi tinggi telah ditemukan, barulah dicari aturan asosiasi yang memenuhi syarat *minimum confidence*, dengan menghitung *confidence* aturan asosiasi $A \rightarrow B$. Untuk memperoleh nilai *confidence*, dilakukan dengan rumus :

$$Confidence(A \rightarrow B) = \frac{\text{jumlah kejadian mengandung } A, \text{ dan } B}{\text{jumlah kejadian mengandung } A} \quad (2.4)$$

Aturan asosiasi $A \rightarrow B$ dikatakan dengan A sebagai *anteseden* dengan B sebagai *konsekuen*. *Anteseden* merupakan sebab yang menjadi *item* *konsekuen*. Sedangkan *konsekuen* adalah sebuah akibat atau juga *item* yang akan terjadi setelah terjadi *anteseden* (Tommy, 2017).

Dalam algoritma apriori, ada 2 proses utama yang dilakukan untuk mendapatkan *frequent itemset*, yaitu (Saputro, 2017) :

1) *Join* (penggabungan)

Dalam proses ini, setiap *item* dikombinasikan dengan *item* lain sampai tidak dapat terbentuk kombinasi lagi.

2) *Pruning* (pemangkasan)

Pada proses ini, hasil kombinasi *item* akan dipangkas berdasarkan *minimum support* yang telah ditentukan.

2.4. Lift Ratio

Lift Ratio menghasilkan ukuran untuk menguji ke-valid-an aturan yang sudah terbentuk dan mengetahui kekuatan pada *association rule* yang telah terbentuk (Riszky & Sadikin, 2019). Nilai *Lift* menunjukkan adanya kekuatan dari aturan atas kejadian acak dari *antecedent* dan *consequent* berdasarkan pada *support*-nya masing-masing. Hal ini akan memberikan informasi

tentang probabilitas dari *consequent* berdasarkan *antecedent* (Despitaria, Sujaini, & Tursina, 2016). Untuk menghitung nilai *lift ratio* digunakan rumus sebagai berikut :

$$Lift\ Ratio = \frac{Confidence\ Antecedent}{Support\ Consequent} \quad (2.5)$$

Aturan dikatakan kuat apabila nilai *lift ratio* > 1. Lebih tinggi nilai *lift ratio* menunjukkan bahwa lebih besar kekuatan dari asosiasi.

2.5. Penelitian Terkait

Dalam penyusunan skripsi ini, terdapat penelitian-penelitian terdahulu yang berkaitan dengan judul penelitian ini antara lain :

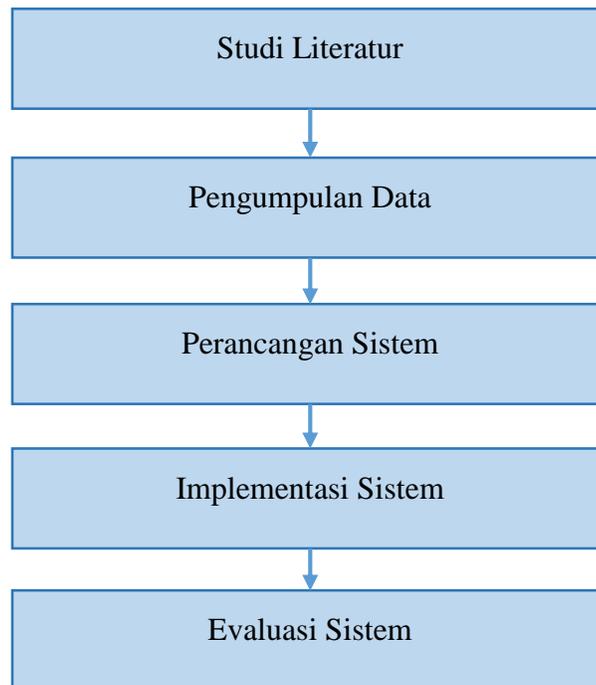
Citra Verawati Purba dan Efori Buulo (2020). Judul penelitian “Implementasi Algoritma Apriori Untuk Menentukan Pola Data Penyakit Pada Anak Usia Dini (Studi Kasus : RS. Estomihi)”. Penelitian tersebut terdapat 7 jenis penyakit yang digunakan sebagai atribut penelitian, diantaranya yaitu : (1) Thypoid, (2) Demam Dengue, (3) Bacterial Infection, (4) Prolenged Fever, (5) Infeksi Saluran Pernapasan Atas, (6) Demam Rematik, (7) Viral Infection. Algoritma apriori digunakan untuk mengetahui pola data penyakit yang sering diderita anak usia dini pada Rumah Sakit Estomihi. Terdapat 3 kesimpulan dari penelitian tersebut, yaitu : (1) Pola penyakit pasien anak usia dini dapat diketahui dengan menggunakan data rekam medis dari rumah sakit dan mengambil data-data yang mempengaruhi penyakit pasien tersebut, (2) Penerapan Algoritma Apriori pada teknik *data mining* sangat efisien dan dapat mempercepat proses pembentukan pola kombinasi *itemset* pada anak usia dini pada RS Estomihi, yaitu dengan *support* dan *confidence* yang tertinggi, (3) Perhitungan *data mining* dengan weka sangat mudah dilakukan dengan menginput data ke dalam sebuah *file* yang berformat .csv kemudian menyimpan *file* tersebut dengan format .arff yang terdapat di dalam weka untuk mendapatkan hasil prediksi dari aplikasi tersebut (Purba & Buulolo, 2020).

Ismul Zamroni, Indah Werdiningsih, dan Purbandini (2017). Judul penelitian “Identikasi Pola Penyakit Anak Di Bawah 5 Tahun (Balita) Dengan Menggunakan Algoritma Apriori”. Atribut yang digunakan dalam penelitian tersebut meliputi keluhan penyakit (1) Panas, (2) Batuk, (3) Pucat, (4) Pilek, (5) Muntah, (6) Pusing, (7) Diare, (8) Mencret, (9) Mual. Dengan

menggunakan Algoritma apriori, penulis mengharapkan untuk dapat membantu menemukan dan mengetahui pola penyakit pada balita. Kesimpulan dari penelitian tersebut adalah berdasarkan hasil analisa yang dilakukan dengan beberapa kombinasi minimum support dan *minimum confidence* diperoleh berupa hasil gejala penyakit yang paling kuat yaitu panas, batuk, pilek, dan pucat (Zamroni et al., 2017).

BAB III METODOLOGI PENELITIAN

Penelitian ini dilakukan dengan mengikuti tahapan meliputi studi literatur, pengumpulan data, perancangan sistem, implementasi sistem, dan evaluasi sistem. Tahapan penelitian diperlihatkan pada Gambar 3.1.



Gambar 3.1. Tahapan penelitian

3.1. Studi Literatur

Studi literatur yang dilakukan adalah memahami latar belakang masalah sehingga dapat menentukan ruang lingkup masalah, tujuan, dan solusi dari masalah yang akan diteliti. Mencari dan mempelajari penelitian terkait yang telah diteliti sebelumnya untuk memberikan referensi dalam menyelesaikan permasalahan yang akan diteliti.

Studi literatur membantu dalam menemukan teori-teori yang berkaitan dengan *data mining*, *association rule* terkhususnya algoritma apriori serta bahasa pemrograman dan ketersediaan *library* yang akan digunakan.

3.2. Pengumpulan Data

Data dari penelitian ini diperoleh dari RSIA Sitti Khadijah 1 Muhammadiyah Cabang Makassar, RSIA Sitti Fatimah Makassar, RSIA Bahagia, dan RSUD Labuang Baji. RSIA Sitti Khadijah 1 Muhammadiyah Cabang Makassar, RSIA Fatimah Makassar, dan RSIA Bahagia merupakan rumah sakit ibu dan anak serta RSUD Labuang Baji merupakan rumah sakit umum daerah Kota Makassar yang memiliki beberapa kasus kematian yang terjadi pada bayi. Data yang digunakan adalah data kematian bayi di RSIA Sitti Khadijah 1 Muhammadiyah Cabang Makassar dari tahun 2014-2021, RSIA Sitti Fatimah Makassar dari tahun 2018-2021, RSIA Bahagia dari tahun 2017-2021, dan RSUD Labuang Baji dari tahun 2014-2021. Jumlah data yang digunakan dalam penelitian ini ialah sebanyak 969 data.

3.3. Perancangan Sistem

Tahapan awal dalam perancangan sistem adalah melakukan tahapan-tahapan data mining. Adapun tahapan-tahapan data mining adalah sebagai berikut:

3.3.1. Data Collection

Proses *data collection* dilakukan dengan mengumpulkan data penyebab kematian bayi sebagai *dataset* yang akan digunakan dalam penelitian. *Dataset* yang digunakan adalah data dari RSIA Sitti Khadijah 1 Muhammadiyah Cabang Makassar, RSIA Sitti Fatimah Makassar, RSIA Bahagia, dan RSUD Labuang Baji dari tahun 2014 - 2021. Jumlah data yang digunakan adalah sebanyak 969 data kematian bayi. Data tersebut berisi nama orang tua bayi, jenis kelamin, penyebab kematian yaitu sianosis, kelainan kongenital, sepsis, *hygroma colli*, asfiksia, BBLR (Berat Badan Lahir Rendah), prematur, placenta pravia, RDS (*Respiratory Distress Syndrom*), dan pendarahan. Adapun fitur yang digunakan sesuai dengan tujuan penelitian ini yaitu data penyebab-penyebab kematian bayi seperti sianosis, kelainan kongenital, sepsis, *hygroma colli*, asfiksia, BBLR (Berat Badan Lahir Rendah), prematur, placenta pravia, RDS (*Respiratory Distress Syndrom*), dan pendarahan. Contoh data dari setiap rumah sakit dapat dilihat pada Gambar 3.2, Gambar 3.3, Gambar 3.4, dan Gambar 3.5.

Nomor	Nama Orang Tua	Jenis Kelamin	Penyebab Kematian					
1	Ammu Mina	L	Sianosis	Kelainan Kongenital	Sepsis	Hygroma Colli		
2	Sri Wahyuni	P	Asfksia	BBLR	Kelainan Kongenital			
3	Mudrika	P	Prematur	Sianosis	Kelainan Kongenital			
4	Nurmi'mah	L	Sianosis	Kelainan Kongenital				
5	Ismiyanti	P	Sianosis	Kelainan Kongenital	Prematur			
6	Nuraena	L	Sianosis	Kelainan Kongenital	BBLR			
7	Musyawwir	P	Prematur	Sianosis	Kelainan Kongenital			
8	Fitriani	P	Asfksia	Sianosis	Kelainan Kongenital			
9	Zulmuwawwinah	P	BBLR	Kelainan Kongenital	Sianosis			
10	Rosalina	P	BBLR	Kelainan Kongenital	Sianosis			

Gambar 3. 2. Contoh Data RSIA Sitti Khadijah 1 Muhammadiyah Cabang Makassar

Nomor	Nama Orang Tua	Jenis Kelamin	Penyebab Kematian					
1	Megawati	L	RDS	Asfksia				
2	Wahyuni /	L	RDS	Kelainan Kongenital	Sianosis	BBLR		
3	Hamdani	L	Kelainan Kongenital	Asfksia	Sianosis			
4	Satriani	P	Asfksia	Sianosis	Kelainan Kongenital			
5	Saenab	L	RDS	BBLR	Sianosis			
6	Juli	P	RDS	Asfksia				
7	Kartia	P	BBLR	Sianosis	Kelainan Kongenital			
8	Suleha	P	BBLR	Prematur				
9	Muliana	L	Pendarahan	Hygroma Colli	Kelainan Kongenital			
10	Ifitah	P	Hygroma Colli	Asfksia	BBLR	Sianosis		

Gambar 3. 3. Contoh Data RSIA Sitti Fatimah Makassar

Nomor	Nama Orang Tua	Jenis Kelamin	Penyebab Kematian					
1	Hasrawati	L	Asfksia	BBLR	RDS	Kelainan Kongenital	Sianosis	
2	Hamdana	P	RDS	BBLR				
3	Nitrawati	L	Sepsis	Sianosis	Kelainan Kongenital			
4	Nurhalisa	L	RDS	Sepsis				
5	Irawati	P	RDS	Kelainan Kongenital	Sianosis			
6	Klara Inggit	P	RDS	Sianosis	Kelainan Kongenital			

Gambar 3. 4. Contoh Data RSIA Bahagia

Nomor	Nama Orang Tua	Jenis Kelamin	Penyebab Kematian				
1	Armita Arief	P	BBLR	Sianosis	Kelainan Kongenital		
2	Sawariah	P	Kelainan Kongenital	RDS	Sianosis		
3	Rabiatul	L	BBLR	RDS	Sianosis	Kelainan Kongenital	
4	Susi Isnawati	L	Kelainan Kongenital	Asfiksia			
5	Nurhidayah	P	RDS	BBLR	Sianosis	Kelainan Kongenital	
6	Arwinda	P	RDS	Kelainan Kongenital	Sianosis		
7	Syamsiah	P	Sianosis	RDS	Sianosis	Kelainan Kongenital	
8	Nurhayati	L	BBLR	RDS	Kelainan Kongenital	Sianosis	

Gambar 3. 5. Contoh Data RSUD Labuang Baji

3.3.2. Data Cleaning

Proses *data cleaning* ialah dengan melakukan pembersihan data terhadap data yang memiliki *noise* dan data yang tidak konsisten. Dalam penelitian ini, dataset yang digunakan terdapat *missing value* akibat dari tidak konsistennya jumlah kolom pada data. Proses pembersihan data akan menghilangkan *missing value* atau biasa muncul sebagai “NaN” yang ada pada *dataset*. *Missing value* atau “NaN” yang terbaca oleh sistem akan diganti menjadi *empty*. *Missing value* pada *dataset* dapat menyebabkan data menjadi kacau dan terjadi kekeliruan dalam proses pengolahan nantinya. Proses *data cleaning* dapat dilihat pada Bab IV bagian 4.1.1.

3.3.3. Data Integration

Proses *data integration* (penggabungan data) dilakukan dengan menggabungkan data dari RSIA Sitti Khadijah 1 Muhammadiyah Cabang Makassar, RSIA Sitti Fatimah Makassar, RSIA Bahagia, dan RSUD Labuang Baji dari tahun 2014 -2021 menjadi sebuah *dataset* baru dengan jumlah *record* yang terjadi yaitu sebanyak 969 baris. Dari data tersebut akan dilakukan proses seleksi data untuk melakukan pemilihan fitur yang akan digunakan. Gambar 3.6 menunjukkan data yang telah di integrasikan.

No.	Nama Orang Tua	Jenis Kelamin	Penyebab Kematian					
			Sianosis	Kelainan Kongenital	Sepsis	Hygroma Colli		
1	Amnu Mina	L	Sianosis	Kelainan Kongenital	Sepsis	Hygroma Colli		
2	Sri Wahyuni	P	Asfiksia	BBLR	Kelainan Kongenital			
3	Mudrika	P	Prematur	Sianosis	Kelainan Kongenital			
4	Nurni'mah	L	Sianosis	Kelainan Kongenital				
5	Ismiyanti	P	Sianosis	Kelainan Kongenital	Prematur			
6	Nuraena	L	Sianosis	Kelainan Kongenital	BBLR			
7	Musyawwir	P	Prematur	Sianosis	Kelainan Kongenital			
8	Fitriani	P	Asfiksia	Sianosis	Kelainan Kongenital			
9	Zulmuwawwinah	P	BBLR	Kelainan Kongenital	Sianosis			
10	Rosalma	P	BBLR	Kelainan Kongenital	Sianosis			
11	M. Dafa Azlia	P	Asfiksia	Kelainan Kongenital	Sianosis			
12	Ade Irma	P	Sepsis	Sianosis	Kelainan Kongenital			
13	Erialma	P	Prematur	Placenta Pravia				
14	Wahidah	P	BBLR	Sianosis	Kelainan Kongenital			
15	Syamsinar	P	Sianosis	Sepsis	Kelainan Kongenital			
16	Jumriani	P	Placenta Pravia	Kelainan Kongenital	Sepsis	Asfiksia	RDS	Sianosis
17	Asni	P	Placenta Pravia	Sianosis	Kelainan Kongenital	BBLR		
18	Nur Inayah	P	Sianosis	BBLR	Kelainan Kongenital			
19	Surianti Harum	L	Sianosis	BBLR	Kelainan Kongenital	Asfiksia		
20	Nirmayanti	P	Sianosis	Kelainan Kongenital	Asfiksia	BBLR		
21	Syahriani	P	Sianosis	Asfiksia	Kelainan Kongenital			
22	Hostati	L	Sianosis	BBLR	Kelainan Kongenital			
23	Fitria	P	Sianosis	BBLR	Kelainan Kongenital			

Gambar 3. 6. Penggabungan Data

3.3.4. Data Selection & Transformation

Tidak semua data akan digunakan untuk proses analisis, dengan melakukan *data selection* maka akan dipilih fitur – fitur yang akan tepat untuk digunakan dalam proses analisis. Proses data selection dilakukan dengan memilih fitur – fitur yang relevan dan mengandung informasi berguna untuk proses analisis sesuai dengan tujuan penelitian. Dari proses tersebut, ditetapkan 10 fitur dari 12 fitur yang ada, fitur tersebut yaitu data yang berupa faktor penyebab kematian bayi meliputi sianosis, kelainan kongenital, sepsis, *hygroma colli*, asfiksia, BBLR (Berat Badan Lahir Rendah), prematur, placenta pravia, RDS (*Respiratory Distress Syndrom*), dan pendarahan. Fitur tersebut dipilih berdasarkan hipotesis yaitu berhubungan dengan penyakit dan penyebab kematian pada bayi. Selain itu, semua fitur tersebut muncul pada setiap data dari ke-empat rumah sakit yang berbeda, sehingga dapat diketahui bahwa semua kejadian penyebab kematian bayi yang menjadi fitur dalam penelitian ini terjadi di setiap rumah sakit tanpa melewatkan salah satu fitur tersebut. Kemudian, data yang tidak relevan atau tidak sesuai dengan tujuan penelitian tidak akan dipilih untuk proses analisis, seperti data nama orang tua bayi dan jenis kelamin bayi

yang tidak relevan dengan penyebab kematian bayi. Selain itu nama orang tua bayi merupakan data yang bersifat rahasia.

3.3.5. Data Mining

Metode *data mining* yang digunakan ialah metode asosiasi dengan menggunakan algoritma apriori. Algoritma apriori dipilih karena pada algoritma apriori melakukan *scan database* setiap kali iterasi, dibandingkan dengan algoritma asosiasi lainnya seperti algoritma *FP_Growth* yang hanya melakukan *scan database* sekali di awal. Setelah melakukan studi literatur, didapatkan beberapa penelitian terkait dengan penggunaan algoritma apriori untuk mengetahui keterkaitan antara fitur penyakit pada anak, sehingga algoritma apriori yang dipilih untuk melakukan asosiasi keterkaitan pada penyebab kematian pada bayi, selain itu algoritma apriori sudah *familiar* digunakan sebelumnya.

3.3.6. Pattern Evaluation

Pattern Evaluation (evaluasi pola) ialah mengidentifikasi pola informasi menarik yang dihasilkan pada proses *data mining*. Pola – pola yang didapatkan akan dievaluasi untuk mengetahui apakah hipotesa telah tercapai. Informasi dari pola menarik yang ditemukan dapat dijadikan umpan balik untuk memperbaiki proses data mining atau metode yang digunakan.

3.3.7. Knowledge Presentation

Knowledge presentation merupakan visualisasi dan penyajian pengetahuan mengenai metode yang digunakan untuk memberikan pengetahuan pada pengguna atau oranglain. Bentuk *knowledge presentation* yang diberikan adalah berupa hasil analisis keterkaitan antara fitur-fitur pada penyebab kematian bayi berdasarkan *dataset* dari RSIA Sitti Khadijah 1 Muhammadiyah Cabang Makassar, RSIA Sitti Fatimah Makassar, RSIA Bahagia, dan RSUD Labuang Baji 2014 - 2021. Serta, peng-aplikasi-an program menggunakan website.

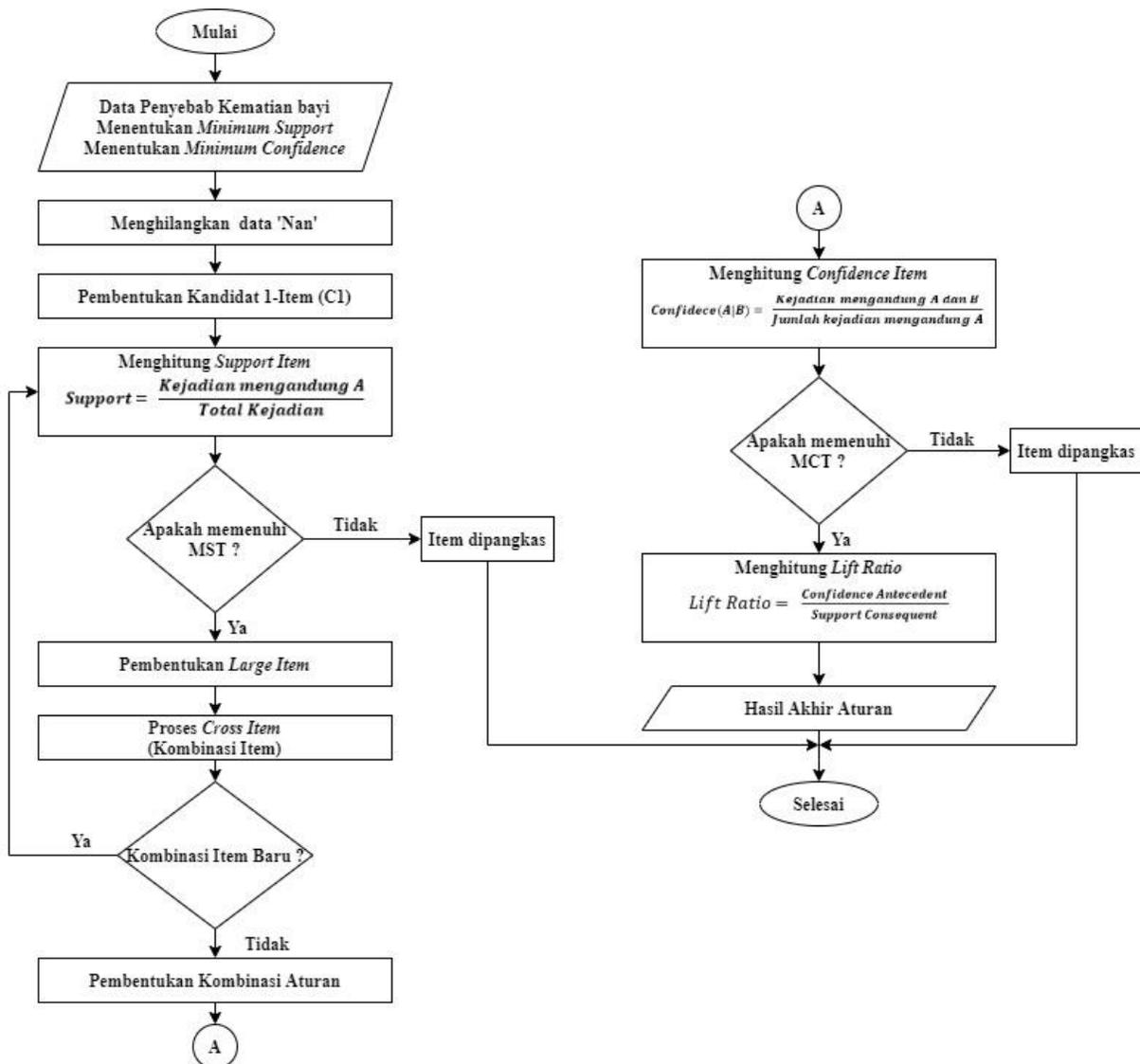
Tahap selanjutnya dalam perancangan sistem penelitian ini adalah dengan mengimplementasikan Algoritma Apriori untuk menentukan penyebab terbesar kematian pada bayi. Tahap awal perancangan yang dilakukan adalah menentukan nilai *minimum support* dan *minimum confidence* yang akan digunakan sebagai nilai batasan. Kemudian menentukan pola *frequent* tinggi, dengan menghitung jumlah frekuensi *item* yang berisi *k-itemset* (*k*). Hasil dari *itemset* yang telah dihitung akan menjadi kandidat *1-itemset* (C1). Himpunan *itemset* yang merupakan kandidat *1-itemset* (C1) akan dihitung nilai *support*-nya. Kandidat *1-itemset* (C1) yang memenuhi *minimum support* akan membentuk *Large 1-itemset* (L1). *Itemset* yang tidak termasuk *Large 1-Itemset* (L1) tidak diikutsertakan dalam iterasi selanjutnya. *Itemset* pada *Large 1-itemset* (L1) akan dilakukan proses *cross* yang akan membentuk kandidat *2-itemset* (C2) dengan melakukan proses penggabungan 2 *item*. Nilai *support* dari kandidat *2-itemset* (C2) yang memenuhi *minimum support* akan termasuk dalam *Large 2-Itemset* (L2). Untuk iterasi selanjutnya, proses gabungan dari *Large 2-itemset* (L2) akan membentuk kandidat *itemset* yang baru (Ck). Dari Kandidat *itemset* (Ck) akan membentuk *Large itemset* yang baru (Lk). Iterasi akan berhenti apabila Lk sudah tidak ada yang memenuhi *minimum support*.

Tahap selanjutnya, pembentukan aturan asosiasi yang menyatakan kuatnya hubungan antar kombinasi *itemset* pada setiap transaksi. Setelah pola-pola frekuensi tinggi ditemukan, kemudian dibentuk aturan asosiasi berdasarkan *item* yang memenuhi syarat *minimum confidence*. *Itemset* yang nilai *confidence* tidak memenuhi *minimum confidence* bukan merupakan aturan asosiasi yang dipakai. Kemudian dilakukan perhitungan *Lift Ratio* untuk melihat kekuatan dari aturan yang telah diperoleh.

3.4. Implementasi Sistem

Implementasi sistem dengan Algoritma Apriori dilakukan dengan dua cara yaitu *scratch* program dengan bahasa pemrograman Python (tanpa *library*) dan program menggunakan *library* Python. Cara pertama adalah dengan membuat program Python tanpa menggunakan *Apriori library*, program akan menampilkan proses dari tiap tahapan-tahapan dari Algoritma Apriori. Cara kedua adalah menggunakan *library* dengan meng-*import* dari *library* Python yang

ada. *Library Python* yang digunakan dalam membangun sistem untuk memudahkan dalam melakukan pemrosesan data adalah *Pandas Library*. *Pandas Library* digunakan untuk membaca inputan data dalam berbagai format seperti csv, xlsx, dan tsv, lalu mengubahnya menjadi bentuk data *frame* yang memiliki baris dan kolom. Sedangkan, untuk proses dari tahapan Algoritma Apriori yaitu menggunakan *efficient-apriori* dari *Apriori Library*. Dengan menggunakan *Apriori Library* sistem akan langsung menampilkan hasil dari proses Algoritma Apriori dengan meng-input-kan *minimum support* dan *minimum confidence*. Gambaran Umum Algoritma Apriori dapat dilihat pada Gambar 3.2.



Gambar 3.2. Algoritma Apriori

3.5. Evaluasi Sistem

Pengujian merupakan salah satu bentuk dalam melakukan evaluasi sistem. Pengujian dilakukan dengan membandingkan hasil asosiasi dari program Python yang menggunakan *library* Apriori dan program yang tidak menggunakan *library* Apriori. Perbandingan dilakukan dengan memperhatikan kesamaan *output* tiap tahapan Algoritma Apriori dan hasil akhir aturan asosiasi yang didapatkan dari kedua program.

Hasil akhir dan tahapan yang dihasilkan program Python tanpa *library* menampilkan aturan asosiasi yang sama dengan program Python dengan *library*. Program Python yang menggunakan *library* Apriori dijadikan sebagai pembanding untuk menunjukkan keberhasilan dari program Python yang tidak menggunakan *library* Apriori. Dengan menganalisis aturan-aturan yang diperoleh dari program dapat diketahui keterkaitan dari penyebab kematian bayi pada tiga Rumah Sakit Ibu dan Anak yang ada di Makassar, serta satu Rumah Sakit Umum Daerah Kota Makassar.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Penerapan Algoritma Apriori dengan Menggunakan Program Python

Penerapan Algoritma Apriori dilakukan dengan beberapa tahapan yaitu peng-*input-an* *dataset*, menentukan *Minimum Support* dan *Minimum Confidence*, menentukan (*CI*), menentukan *Large Itemset (LI)* dengan melihat apakah *Item* memenuhi *Minimum Support*, menentukan *Ck* dan *Lk*, kemudian melihat apakah *Item* memenuhi *Minimum Confidence*. *Item* yang memenuhi *Minimum Confidence* akan menghasilkan aturan asosiasi. Perhitungan *Lift ratio* akan menunjukkan kekuatan dari aturan yang didapatkan.

4.1.1. *Dataset* Penyebab Kematian pada Bayi dari Dua RSIA di Makassar

Dataset yang digunakan adalah data dari tiga Rumah Sakit Ibu dan Anak serta satu Rumah Sakit Umum Daerah yang ada di Makassar, yaitu RSIA Sitti Khadijah 1 Muhammadiyah Cabang Makassar, RSIA Fatimah Makassar, RSIA Bahagia, dan RSUD Labuang Baji. Data tersebut berisi kasus penyebab-penyebab kematian yang terjadi di ketiga Rumah Sakit Ibu dan Anak serta satu Rumah Sakit Umum Daerah, terdiri dari 969 data penyebab kematian yang terjadi dalam kurun waktu 2014-2021.

Dataset tersebut diberi nama “Dataskripsi.xlsx”, yang berisi data penyebab kematian yaitu sianosis, kelainan kongenital, sepsis, *hygroma colli*, asfiksia, BBLR (Berat Badan Lahir Rendah), prematur, placenta pravia, RDS (*Respiratory Distress Syndrom*), dan pendarahan. Pada Gambar 4.1. menampilkan 10 contoh data penyebab kematian pada Bayi. Untuk seluruh data penyebab kematian pada bayi dapat dilihat pada link github yang ada pada Lampiran 2.

Out[4]:

	0	1	2	3	4	5	6
0	Sianosis	Kelainan Kongenital	Sepsis	Hygroma Colli	NaN	NaN	NaN
1	Asfiksia	BBLR	Kelainan Kongenital	NaN	NaN	NaN	NaN
2	Prematur	Sianosis	Kelainan Kongenital	NaN	NaN	NaN	NaN
3	Sianosis	Kelainan Kongenital	NaN	NaN	NaN	NaN	NaN
4	Sianosis	Kelainan Kongenital	Prematur	NaN	NaN	NaN	NaN
5	Sianosis	Kelainan Kongenital	BBLR	NaN	NaN	NaN	NaN
6	Prematur	Kelainan Kongenital	Sianosis	NaN	NaN	NaN	NaN
7	Asfiksia	Kelainan Kongenital	Sianosis	NaN	NaN	NaN	NaN
8	BBLR	NaN	NaN	NaN	NaN	NaN	NaN
9	BBLR	Kelainan Kongenital	Sianosis	NaN	NaN	NaN	NaN
10	Asfiksia	NaN	NaN	NaN	NaN	NaN	NaN

Gambar 4. 1. Contoh *Dataset* Penyebab Kematian pada Bayi

```
import pandas as pd
import numpy as np

min_support = 0.75 # nilai default dari minimum support
min_confidence = 0.90 # nilai default dari minimum confidence

causes_data_file = pd.read_excel(r'E:\UNHAS\Skripsweet\Data\Dataskripsi.xlsx', header = None)
print('Dataset Penyebab Kematian bayi :\n-----')
causes_data_file
```

Gambar 4. 2. Kode Program input dataset

Berdasarkan Gambar 4.2, *dataset* yang di-*input*-kan akan disimpan pada *causes_data_file*. Pada bagian ini, juga dilakukan peng-*input*-an *minimum support* dan *minimum confidence* sebagai nilai batasan yang digunakan, dimana *min_support* = 0.75 dan *min_confidence* = 0.95.

Gambar 4.1. menunjukkan bahwa *dataset* yang di-*input*-kan belum melalui proses *cleaning data*. Untuk menghindari kekeliruan dalam proses analisis, maka dilakukan *cleaning data* untuk menghilangkan nilai “nan” yang dianggap data pada sistem. Gambar 4.3. menampilkan *dataset* yang sudah dilakukan proses pembersihan menggunakan *code* program pada Gambar 4.4.

```
[['Sianosis', 'Kelainan Kongenital', 'Sepsis', 'Hygroma Colli'],
 ['Asfiksia', 'BBLR', 'Kelainan Kongenital'],
 ['Prematur', 'Sianosis', 'Kelainan Kongenital'],
 ['Sianosis', 'Kelainan Kongenital'],
 ['Sianosis', 'Kelainan Kongenital', 'Prematur'],
 ['Sianosis', 'Kelainan Kongenital', 'BBLR'],
 ['Prematur', 'Kelainan Kongenital', 'Sianosis'],
 ['Asfiksia', 'Kelainan Kongenital', 'Sianosis'],
 ['BBLR'],
 ['BBLR', 'Kelainan Kongenital', 'Sianosis'],
 ['Asfiksia'],
 ['Sepsis'],
 ['Prematur', 'Placenta Pravia'],
 ['BBLR'],
 ['Sianosis', 'Sepsis'],
```

Gambar 4. 3. *Dataset* setelah dilakukan *cleaning data*

```
1 dataset = []
2
3 for i in range(0, len(causes_data_file)):
4     dataset.append([cause for cause in causes_data_file.values[i,:] if str(cause)!='nan'])
5
6 print('Dataset (num={0}) :\n-----'.format(len(dataset)))
7 dataset
```

Gambar 4. 4. *Kode Program cleaning data*

Pada Gambar 4.3 akan melakukan *cleaning data* dengan menghilangkan ‘nan’ pada *causes_data_file* yang kemudian disimpan pada *dataset* yang akan digunakan untuk proses selanjutnya.

4.1.2. Pembentukan Kandidat 1-Item (C1)

Item yang akan menjadi kandidat 1-Item (C1) dapat dilihat pada Gambar 4.5 dengan menggunakan kode Python yang diperlihatkan pada Gambar 4.6.

```
[{'Sianosis'},
 {'Kelainan Kongenital'},
 {'Sepsis'},
 {'Hygroma Colli'},
 {'Asfiksia'},
 {'BBLR'},
 {'Prematur'},
 {'Placenta Pravias'},
 {'RDS'},
 {'Pendarahan'}]
```

Gambar 4. 5. Hasil Kandidat 1-Item (C1)

Gambar 4.5 memperlihatkan 10 fitur yang menjadi Kandidat 1-Item (C1), kemudian dilakukan perhitungan jumlah frekuensi kemunculan *item*.

```
# Pembentukan Kandidat 1-Item (C1)
C1 = []

for data in dataset:
    for item in data:
        if not [item] in C1:
            C1.append([item])

print('C1 Items (num={0}): \n-----'.format(len(C1)))
C1 = list(map(set,C1))
C1
```

Gambar 4. 6. Kode Python untuk Pembentukan Kandidat 1-Item (C1)

4.1.2.1. Menghitung Frekuensi *Item*

Perhitungan frekuensi dilakukan untuk mengetahui frekuensi kemunculan dari setiap *item*, serta digunakan untuk menghitung nilai *support item* dari setiap *item* untuk mendapatkan kandidat 1-Item (C1). yang dapat dilihat pada Gambar 4.7 dengan menggunakan kode Python pada Gambar 4.8.

```
{('Sianosis',): 768,
 ('Kelainan Kongenital',): 788,
 ('Sepsis',): 111,
 ('Hygroma Colli',): 70,
 ('Asfiksia',): 342,
 ('BBLR',): 391,
 ('Prematur',): 66,
 ('Placenta Pravia',): 22,
 ('RDS',): 316,
 ('Pendarahan',): 18}
```

Gambar 4. 7. Frekuensi *Item*

Gambar 4.7. memperlihatkan jumlah frekuensi masing-masing *item* dari kandidat 1-*Item* (C1), dimana frekuensi *item* tertinggi yaitu Kelainan Kongenital dengan jumlah kemunculan yaitu sebanyak 788 data.

```
# Menghitung Frekuensi Item (C1)
C1_freq = {}

for transaction in transactions:
    for item in C1:
        if frozenset(item).issubset(transaction):
            if not tuple(item) in C1_freq:
                C1_freq[tuple(item)] = 1
            else:
                C1_freq[tuple(item)] += 1

print('C1 Frequent (num={0}) :\n-----'.format(len(C1_freq)))
C1_freq
```

Gambar 4. 8. Kode Python untuk menghitung Frekuensi *Item*

4.1.2.2. Menghitung Nilai *Support*

Untuk melakukan Perhitungan *Support Item* maka diterapkan dengan melakukan persamaan (2.1). Nilai dari hasil perhitungan sebagai berikut :

$$Support (Sianosis) = \frac{768}{969} = 0.79$$

$$Support (Kelainan Kongenital) = \frac{788}{969} = 0.81$$

$$Support (Sepsis) = \frac{111}{969} = 0.11$$

$$\text{Support (Hygroma Colli)} = \frac{70}{969} = 0.07$$

$$\text{Support (Asfiksia)} = \frac{342}{969} = 0.35$$

$$\text{Support (RDS)} = \frac{316}{969} = 0.33$$

$$\text{Support (Prematur)} = \frac{66}{969} = 0.07$$

$$\text{Support (Placenta Pravia)} = \frac{22}{969} = 0.02$$

$$\text{Support (BBLR)} = \frac{391}{969} \times 100\% = 0.4$$

$$\text{Support (Pendarahan)} = \frac{18}{969} = 0.02$$

Gambar 4.9. menunjukkan Kandidat 1-Itemset (C1) yang telah dilakukan perhitungan *Support Item* dari setiap fitur, hasil perhitungan yang dilakukan dengan menerapkan persamaan (2.1) memperlihatkan hasil yang sama yang dihasilkan oleh sistem, dimana *item* dengan nilai *Support Item* tertinggi adalah Kelainan Kongenital dengan nilai *support item* yaitu 0.81. Gambar 4.10. menunjukkan kode python yang digunakan untuk menghitung nilai *Support Item*.

```
{('Sianosis',): 0.79,  
( 'Kelainan Kongenital',): 0.81,  
( 'Sepsis',): 0.11,  
( 'Hygroma Colli',): 0.07,  
( 'Asfiksia',): 0.35,  
( 'BBLR',): 0.4,  
( 'Prematur',): 0.07,  
( 'Placenta Pravia',): 0.02,  
( 'RDS',): 0.33,  
( 'Pendarahan',): 0.02}
```

Gambar 4. 9. Nilai *Support Itemset*

```

1 C1_support = {}
2 num_transactions = float(len(transactions))
3
4 for key in C1_freq:
5     C1_support[key] = round(C1_freq[key]/num_transactions,2)
6
7 print('C1 Support Values (num={0}) :\n-----'.format(len(C1_support)))
8 C1_support

```

Gambar 4. 10. Kode python untuk menghitung nilai *support item*

Gambar 4.10. memperlihatkan bahwa *num_transactions* akan menyimpan nilai dari jumlah total data yang tersimpan pada *transactions*. Kemudian, *C1_support* merupakan hasil dari perhitungan *C1_freq* dibagi dengan *num_transactions*, sehingga diperoleh nilai *support item* dengan panjang nilai dua angka setelah koma.

4.1.3. Menentukan *Large 1-Itemset (L1)*

Item yang termasuk dalam *Large 1-Itemset* ditentukan dengan melihat *Support Item* dari setiap *Item*. *Support Item* dari setiap *Item* yang memenuhi *minimum support* maka akan termasuk dalam *Large 1-Itemset*. Adapun *Minimum support* yang digunakan adalah 0.75. *Item* yang termasuk dalam *Large 1-Itemset* dapat dilihat pada Gambar 4.11 dengan menggunakan kode python pada Gambar 4.12

```
{('Sianosis',): 0.79, ('Kelainan Kongenital',): 0.81}
```

Gambar 4. 11. *Large 1-Itemset (L1)*

Berdasarkan Gambar 4.11. hanya ada 2 *item* yang memenuhi *minimum support* untuk menjadi *Kandidat 2-Itemset*, yaitu *Sianosis* dan *Kelainan Kongenital*.

```

# pembentukan untuk Large L-itemset (L1)
L1 = []

for key in C1_support:
    if C1_support[key] >= min_support:
        L1.append(list(key))

print('L1 itemset (num={0}): \n-----'.format(len(L1)))
L1 = list(map(set, L1))
L1

```

Gambar 4. 12. Kode python untuk mendapatkan *Large 1-Itemset (L1)*

Gambar 4.12. menunjukkan bahwa *item* yang menjadi *Large 1-Itemset* harus memenuhi syarat dimana nilai *support item* yang ada pada *C1_support* lebih besar atau sama dengan nilai *min_support* yang telah ditentukan sebelumnya. Hasil dari *item* yang memenuhi syarat akan disimpan pada *L1*.

4.1.4. Menentukan Kandidat *2-Itemset (C2)*

Untuk menentukan Kandidat *2-Itemset (C2)* maka dilakukan dengan kombinasi *item* dari *Large 1-Itemset (L1)* yang telah didapatkan. Setelah menemukan pasangan dari kombinasi tersebut, maka akan dihitung kembali frekuensi dari *item* serta menghitung *support item* dari setiap pasangan *item* tersebut. Kandidat *2-Itemset (C2)* dapat dilihat pada Gambar 4.13.

```
[{'Kelainan Kongenital', 'Sianosis'}]
```

Gambar 4. 13. Kandidat *2-Itemset (C2)*

Setelah mendapatkan kandidat *2-itemset (C2)* pada Gambar 4.13, kemudian menghitung jumlah frekuensi dari kombinasi *2 itemset*. Frekuensi dari kandidat *2-itemset* dapat dilihat pada Gambar 4.14.

```
{('Kelainan Kongenital', 'Sianosis'): 730}
```

Gambar 4. 14. Frekuensi Kandidat *2-Itemset* (C2)

Gambar 4.14 menampilkan jumlah frekuensi dari kombinasi kandidat *2-itemset*, setelah mendapatkan frekuensi dari *itemset*, kemudian dilakukan perhitungan untuk mendapatkan nilai *support* dari kandidat *2-itemset* dengan menerapkan persamaan (2.2). Berdasarkan Gambar 4.15 nilai *support item* dari pasangan item Kelainan Kongenital dan Sianosis adalah 0.75.

```
{('Kelainan Kongenital', 'Sianosis'): 0.75}
```

Gambar 4. 15. Nilai *support* Kandidat *2-Itemset* (C2)

Gambar 4.16 merupakan kode python yang digunakan untuk menentukan Kandidat *k-Itemset* (Ck) termasuk Kandidat *2-Itemset* (C2), Kandidat *3-Itemset* (C3), dan seterusnya selama proses masih memenuhi kriteria untuk melakukan *looping*, yaitu perulangan dari kombinasi *Large k-Itemset* (Lk).

```
# mengambil daftar frequent itemsets, Lk, dan ukuran itemsets, k, untuk menghasilkan Ck
def create_Ck(Lk, k):

    Ck = []
    len_Lk = len(Lk)

    for i in range(len_Lk):
        for j in range(i+1, len_Lk):
            L1 = list(Lk[i][:k-2])
            L2 = list(Lk[j][:k-2])

            L1.sort()
            L2.sort()

            if L1==L2: |
                Ck.append(Lk[i] | Lk[j])

    if len(Ck) == 0:
        return None
    else:
        return Ck
```

Gambar 4. 16. Kode python untuk menghasilkan Kandidat *k-itemset* (Ck)

Frekuensi *item* dihasilkan dengan menggunakan kode Python yang dapat dilihat pada Gambar 4.17.

```
def Ck_frequent_calc(transactions, Ck):
    Ck_freq = {}

    for transaction in transactions:
        for item in Ck:
            if frozenset(item).issubset(transaction):
                if not tuple(item) in Ck_freq:
                    Ck_freq[tuple(item)] = 1
                else:
                    Ck_freq[tuple(item)] += 1

    print('Ck Frequent (num={0}): \n-----'.format(len(Ck_freq)))
    print(Ck_freq, '\n')

    return Ck_freq
```

Gambar 4. 17. Kode python untuk menghitung Frekuensi *Itemset*

Nilai *support* dihasilkan dengan menggunakan kode Python yang diperlihatkan pada Gambar 4.18.

```
def Ck_support_calc(transactions, Ck_freq):
    Ck_support = {}
    num_transactions = float(len(transactions))

    for key in Ck_freq:
        Ck_support[key] = round(Ck_freq[key]/num_transactions,2)

    print('Ck Support Value (num={0}): \n-----'.format(len(Ck_support)))
    print(Ck_support, '\n')

    return Ck_support
```

Gambar 4. 18. Kode python untuk menghitung Nilai *support itemset*

Gambar 4.18. menunjukkan bahwa *Ck_support* berisi hasil dari nilai *Ck_freq* dibagi *num_transactions*.

4.1.5. Menentukan *Large 2-Itemset (L2)*

Untuk menentukan *Large 2-Itemset (L2)* dilakukan seperti cara sebelumnya untuk mendapatkan *Large 1-Itemset (L1)*. *Item* yang termasuk dalam *Large 2-Itemset* ditentukan

dengan melihat nilai *Support Item* dari setiap *Item*. Nilai *support Item* dari setiap *Item* yang memenuhi *minimum support* maka akan termasuk dalam *Large 2-Itemset*. *Large 2-Itemset* dapat dilihat pada Gambar 4.19 dengan menggunakan kode Python yang diperlihatkan pada Gambar 4.20 untuk menentukan *large itemset*, Gambar 4.21 untuk menghitung frekuensi *large itemset*. Gambar 4.22 untuk menghitung nilai *support* dari *large itemset*.

```
{('Kelainan Kongenital', 'Sianosis'): 0.75}
```

Gambar 4. 19. Hasil *Large 1-itemset* (L2)

Berdasarkan Gambar 4.19. Kandidat *2-Itemset* yang memenuhi syarat *minimum support* untuk menjadi *Large 1-Itemset* hanya ada satu pasangan *itemset* yaitu Kelainan Kongenital, Sianosis dengan nilai *support item* yaitu 0.75

```
# pembentukan untuk Large k-itemset (Lk)
def create_Lk(Ck_support, min_support):
    Lk = []

    for key in Ck_support:

        if Ck_support[key] >= min_support:
            Lk.append(set(list(key)))

    print('Lk itemset (num={0}): \n-----'.format(len(Lk)))
    print(Lk, '\n')

    return Lk
```

Gambar 4. 20. Kode python untuk menentukan *Large Itemset* (Lk)

Gambar 4.20. menunjukkan bahwa untuk menghasilkan *Large Itemset*, nilai dari *Ck_support* harus lebih besar atau sama dengan *min_support*. Kemudian hasil dari syarat tersebut disimpan ke dalam *Lk*. Proses *looping* bertujuan untuk menghasilkan *Large Itemset* selanjutnya.

```

def Lk_frequent_calc(transactions, Lk):
    Lk_freq = {}

    for transaction in transactions:
        for item in Lk:
            if frozenset(item).issubset(transaction):
                if not tuple(item) in Lk_freq:
                    Lk_freq[tuple(item)] = 1
                else:
                    Lk_freq[tuple(item)] += 1

    print('Lk Frequent (num={0}): \n-----'.format(len(Lk_freq)))
    print(Lk_freq, '\n')

    return Lk_freq

```

Gambar 4. 21. Kode python untuk menghitung frekuensi *Large Itemset* (Lk)

```

def Lk_support_calc(transactions, Lk_freq):
    Lk_support = {}
    num_transactions = float(len(transactions))

    for key in Lk_freq:
        Lk_support[key] = Lk_freq[key]/num_transactions

    print('Lk Support Value (num={0}): \n-----'.format(len(Lk_support)))
    print(Lk_support, '\n')

    return Lk_support

```

Gambar 4. 22. Kode python untuk menghitung nilai *support Large Itemset* (Lk)

Gambar 4.22 menunjukkan bahwa hasil dari *Lk_freq* akan dibagi dengan *num_transactions* yang kemudian disimpan pada *Lk_support* untuk memperoleh nilai *support* dari setiap *item* pada *Large Itemset*. Proses *looping* akan dilakukan untuk memperoleh nilai *support* pada *Large itemset* selanjutnya.

```

print('Output Proses:\n=====')
L = [L1]
support = [L1_support]

k = 2
while(1): # untuk mencari L2,L3,...

    Ck = create_Ck(L[k-2],k)
    if(Ck is None):
        break

    print('\n>> FOR k = {0}:\n====='.format(k))
    print('Ck Item (num={0}):\n-----'.format(len(Ck)))
    print(Ck,'\n')

    Ck_freq = Ck_frequent_calc(transactions, Ck)
    Ck_support = Ck_support_calc(transactions, Ck_freq)

    Lk = create_Lk(Ck_support, min_support)
    Lk_freq = Lk_frequent_calc(transactions, Lk)
    Lk_support = Lk_support_calc(transactions, Lk_freq)

    L.append(Lk)
    support.append(Lk_support)

    k += 1

```

Gambar 4. 23. Kode python untuk menampilkan Kandidat *k-itemset* (Ck), frekuensi Ck, nilai *support* Ck, *Large k-itemset* (Lk), frekuensi Lk, dan nilai *support* Lk

Gambar 4.23 merupakan kode python yang akan menampilkan *output* Kandidat *k-itemset*, frekuensi *itemset*, nilai *support itemset*, dan *Large k-itemset (Lk)* yang telah didapatkan.

4.1.6. Aturan Asosiasi

Dari seluruh *large itemset* yang memenuhi *minimum support* akan dibentuk menjadi aturan asosiasi. Untuk menghitung nilai *confidence* dari setiap *large itemset* adalah dengan menggunakan persamaan (2.3). Aturan asosiasi yang digunakan adalah *item* yang nilai *confidence*-nya memenuhi *minimum confidence*. Adapun *minimum confidence* yang digunakan adalah 0.95. Aturan asosiasi yang dihasilkan diperlihatkan pada Gambar 4.28 dengan menggunakan kode program pada Gambar 4.29.

```
{'Sianosis'} --> {'Kelainan Kongenital'} .... Confidence = 0.95, Lift Ratio = 1.17
```

Gambar 4. 24. Aturan Asosiasi yang dihasilkan

Gambar 4.24 menunjukkan bahwa hanya ada 1 aturan asosiasi yang memenuhi *minimum confidence* yaitu **Sianosis** berhubungan dengan **Kelainan Kongenital** dengan nilai *confidence* sebesar 0.95.

```
def generate_rules(L, support, min_confidence):
    supportset = {}
    for i in range(0, len(support)):
        for key in support[i]:
            supportset[key] = support[i][key]

    rules = []
    for i in range(1, len(L)):
        for itemset in L[i]:
            H = [set([item]) for item in itemset]

            if i<=2:
                confidence_calc(itemset, H, supportset, rules, min_confidence, i)
            else:
                rules_con_sequence(itemset, H, supportset, rules, min_confidence)

    return (rules, supportset)

def confidence_calc(itemset, H, supportset, rules, min_confidence, i):
    H_m = []

    for con_sequence in H:
        try:
            confidence = supportset[tuple(itemset)]/supportset[tuple(itemset-con_sequence)] # hitung nilai confidence
        except:
            confidence = 0.0

        try:
            lift_ratio = confidence / supportset[tuple(con_sequence)]
        except:
            lift_ratio = 0.0

        if confidence >= min_confidence:
            print (itemset-con_sequence, '--> ',con_sequence,'.... Confidence = ',confidence,', Lift Ratio = ',lift_ratio)

            if not (itemset-con_sequence, con_sequence, confidence) in rules:
                rules.append((itemset-con_sequence, con_sequence, confidence))

            H_m.append(con_sequence)
    return H_m

def rules_con_sequence(itemset, H, supportset, rules, min_confidence):
    m = len(H[0])

    if (len(itemset) > (m + 1)): # coba penggabungan Lebih Lanjut
        H_m = create_Ck(H, m + 1) # buat H_m + 1 calon baru

        try:
            H_m = confidence_calc(itemset, H_m, supportset, rules, min_confidence)

            if (len(H_m) > 1): # membutuhkan setidaknya dua set untuk dtgabungkan
                rules_con_sequence(itemset, H_m, supportset, rules, min_confidence)
        except:
            exit
```

Gambar 4. 25. Kode program python untuk menghasilkan aturan asosiasi

4.1.7. Lift Ratio

Perhitungan *lift ratio* dilakukan dengan menerapkan persamaan (2.5) dan). Nilai *lift ratio* adalah sebagai berikut :

$$\text{Lift Ratio} = \frac{0.95}{0.81} = 1.17$$

```
{'Sianosis'} --> {'Kelainan Kongenital'} .... Confidence = 0.95, Lift Ratio = 1.17
```

Gambar 4. 26. Nilai Lift Ratio

Berdasarkan Gambar 4.26 nilai *lift ratio* dari aturan asosiasi **Sianosis** → **Kelainan Kongenital** adalah sebesar 1.17, dimana nilai *lift ratio* yang dihasilkan oleh sistem sama dengan perhitungan manual yang dilakukan. Kode program yang digunakan dapat dilihat pada Gambar 4.25.

4.2. Penerapan Algoritma Apriori dengan Menggunakan *Library* Python

4.2.1. *Library* Python

Library Python yang digunakan adalah *Efficient Apriori*, merupakan *library* pembelajaran mesin menggunakan bahasa pemrograman python.

4.2.2. Aturan Asosiasi

Untuk mendapatkan aturan asosiasi maka dilakukan peng-*import*-an *Apriori* dari *library efficient_apriori* seperti pada yang diperlihatkan pada Gambar 4.27.

```
itemsets, rules = apriori(transactions, min_support=0.75, min_confidence=0.95)
result = list(rules)

#hasil dari Lk
itemsets
```

Gambar 4. 27. Kode program python menggunakan *library efficient_apriori*

Nilai $min_support = 0.75$ dan $min_confidence=0.95$ dari Gambar 4.27 menunjukkan bahwa *minimum support* yang digunakan adalah 0.75 dan *minimum confidence* yang digunakan adalah 0.95.

Gambar 4.28. menunjukkan item yang termasuk *large itemset* yang diperoleh dari proses algoritma apriori, dimana panjang kombinasi *item* yang terbentuk adalah 2 kombinasi *item* atau *Large 2-Itemset*.

```
{1: {('Sianosis',): 768, ('Kelainan Kongenital',): 788},  
2: {('Kelainan Kongenital', 'Sianosis'): 730}}
```

Gambar 4. 28. *Large itemset* dari *library apriori*

Adapun Aturan asosiasi dihasilkan dari penggunaan *library efficient_apriori* dapat dilihat pada Gambar 4.29.

```
{Sianosis} -> {Kelainan Kongenital} (conf: 0.951, supp: 0.753, lift: 1.169, conv: 3.775)
```

Gambar 4. 29. Aturan Asosiasi dari *library apriori*

4.2.3. Lift Ratio

Nilai *lift ratio* dari aturan asosiasi yang diperoleh dapat dilihat pada Gambar 4.30 dengan menggunakan kode python pada Gambar 4.31.

```
{Sianosis} -> {Kelainan Kongenital} (conf: 0.951, supp: 0.753, lift: 1.169, conv: 3.775)
```

Gambar 4. 30. *Lift Ratio*

```
rules_rhs = filter(lambda rule: len(rule.lhs) == 2 and len(rule.rhs) == 1 , rules)  
for rule in sorted(rules_rhs, key=lambda rule: rule.lift):  
    print(rule)
```

Gambar 4. 31. Kode program python untuk menghitung *confidence* dan *lift ratio*

4.3. Pembahasan

Berdasarkan dengan nilai batasan yang digunakan yaitu *minimum support = 0.75* atau 75% maka diperoleh beberapa *item* dari kandidat *itemset* yang memenuhi syarat untuk

menjadi *Large itemset* yaitu *item* yang memiliki nilai *support* ≥ 0.75 . Adapun 10 *item* yang menjadi kandidat *itemset* yaitu sianosis, kelainan kongenital, sepsis, *hygroma colli*, asfiksia, BBLR (Berat Badan Lahir Rendah), prematur, placenta pravia, RDS (*Respiratory Distress Syndrom*), dan pendarahan. Dari 10 *item* tersebut, *Large 1-Itemset* berisi 2 *item* yang memenuhi syarat untuk menjadi *Large 1-Itemset (L1)* yaitu sianosis dan kelainan kongenital. Kemudian sistem akan melakukan proses *looping* untuk menghasilkan *Large k-Itemset (Lk)* yang akan digunakan untuk proses penentuan aturan asosiasi. Dari proses *looping* tersebut, proses berakhir pada *Large 2-Itemset (L2)* atau kombinasi dua *itemset* yaitu Sianosis dan Kelainan Kongenital, hal ini menandakan bahwa proses *looping* berhenti karena sudah tidak ada *item* yang dapat dilakukan kombinasi dan tidak ada yang memenuhi *minimum support*.

Dengan menggunakan *minimum confidence* = 0.95 atau 95% sebagai batasan kepercayaan untuk mendapatkan aturan asosiasi dari *Large itemset*, maka diperoleh 1 aturan asosiasi yang memenuhi *minimum confidence*, yaitu **Sianosis** berhubungan dengan **Kelainan Kongenital**. Berdasarkan dengan perhitungan *confidence* yaitu jumlah kejadian dari **Sianosis** dibagi dengan total kejadian **Kelainan Kongenital** sehingga diperoleh nilai *confidence* 0.95. Nilai ini sesuai dengan syarat batasan *minimum confidence* yaitu *item* yang memiliki nilai *minimum confidence* ≥ 0.95 .

Dari aturan asosiasi yang didapatkan, maka dilakukan uji *lift ratio* untuk menentukan kekuatan dari aturan sebagai hasil akhir aturan asosiasi. Didapatkan Nilai *lift ratio* sebesar 1.17 berdasarkan perhitungan dari *lift ratio* yaitu nilai *confidence* **Kelainan Kongenital** dan **Hygroma Colli** dibagi nilai *support* **Sianosis**. Aturan yang diperoleh dikatakan kuat karena $1.17 > 1$.

Hasil aturan dari program *non-library apriori* dan yang menggunakan *library apriori* diperlihatkan pada Tabel 4.1.

Tabel 4. 1. Aturan Asosiasi dari program *non-library apriori* dan *library apriori*

<i>Non-Library</i>			<i>Library</i>		
<i>Association Rule</i>	<i>Confidence</i>	<i>Lift Ratio</i>	<i>Association Rule</i>	<i>Confidence</i>	<i>Lift Ratio</i>
Sianosis → Kelainan Kongenital	0.95	1.17	Sianosis → Kelainan Kongenital	0.95	1.169

Berdasarkan pada Tabel 4.1 memperlihatkan perbandingan *output* dari kedua program, dari kedua program menunjukkan *output* yang sama, baik dari Aturan asosiasi, *confidence*, dan *Lift Ratio* yang dihasilkan. Hal ini membuktikan keberhasilan dari program python yang tidak menggunakan *library python* karena menunjukkan hasil yang sama dengan program python yang menggunakan *library python*.

Selain menggunakan program python dengan *library apriori*, dilakukan perbandingan dengan menggunakan algoritma asosiasi lainnya, yaitu algoritma *FP-Growth*, untuk membuktikan bahwa algoritma apriori dapat digunakan untuk menghasilkan aturan asosiasi. Hasil analisis menggunakan algoritma *FP-Growth* dapat dilihat pada Gambar 4.32.

	support	itemsets
0	0.813209	(Kelainan Kongenital)
1	0.792570	(Sianosis)
2	0.753354	(Sianosis, Kelainan Kongenital)

Frekuensi Itemset

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Sianosis)	(Kelainan Kongenital)	0.79257	0.813209	0.753354	0.950521	1.168851	0.108829	3.775134

Hasil akhir aturan

Gambar 4. 32. Algoritma *Fp-Growth*

Adapun perbandingan waktu komputasi antara algoritma Apriori dan algoritma *FP-Growth* dapat dilihat pada Tabel 4.2.

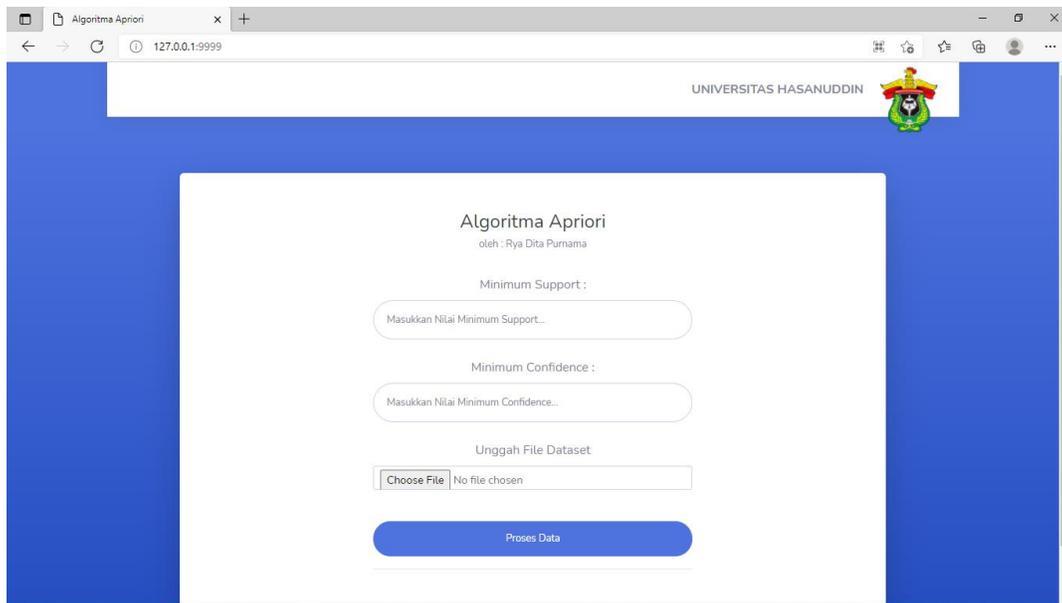
Tabel 4. 2. Perbandingan waktu komputasi antara algoritma Apriori dan algoritma *FP-Growth*

	Algoritma Apriori	Algoritma <i>FP-Growth</i>
<i>Min. Support</i>	0.75	0.75
<i>Min. Confidence</i>	0.95	0.95
Waktu Proses	1.3310908	1.519485

Berdasarkan Tabel 4.2. menunjukkan waktu yang dibutuhkan antara dua algoritma yang cukup dekat dan tidak terlalu jauh perbedaan waktu yang dibutuhkan antara keduanya. Dari hasil yang diperoleh dari Algoritma *Fp-Growth* dan perbandingan waktu komputasi membuktikan bahwa algoritma *Fp-Growth* tidak lebih baik daripada algoritma Apriori karena menunjukkan hasil *Frequent Itemset* dan hasil aturan asosiasi yang sama.

4.4. Pengembangan Aplikasi berbasis Website

Untuk pengaplikasian program maka dapat diakses menggunakan website. Website dibangun menggunakan aplikasi *PyCharm* dengan menggunakan *Flask Apriori*. Aplikasi dapat memproses *file* yang di-*input*-kan dengan menentukan nilai *minimum support* dan *minimum confidence*. Tampilan utama dari website dapat dilihat pada Gambar 4.33



Gambar 4. 33. Tampilan Aplikasi Website

Aplikasi akan menampilkan *output* berdasarkan *step-step* proses algoritma apriori, tampilan *output* dari website dapat dilihat pada lampiran 1.

BAB V

PENUTUP

5.1. Kesimpulan

Penggunaan algoritma apriori diujicobakan pada *dataset* penyebab kematian pada bayi dari RSIA Sitti Khadijah 1 Muhammadiyah Cabang Makassar, RSIA Fatimah Makassar, RSIA Bahagia, dan RSUD Labuang Baji dari tahun 2014 s.d 2021 dengan jumlah data yang diolah sebanyak 969 kejadian. Dari data tersebut, proses analisis dilakukan dengan menggunakan 10 fitur yang menjadi penyebab kematian pada bayi yaitu sianosis, kelainan kongenital, sepsis, *hygroma colli*, asfiksia, BBLR (Berat Badan Lahir Rendah), prematur, placenta pravia, RDS (*Respiratory Distress Syndrom*), dan pendarahan. Dari hasil proses pemangkasan data, diketahui bahwa penyebab kematian Sianosis terjadi bersamaan dengan penyebab kematian Kelainan Kongenital berdasarkan 75% dari total keseluruhan data.

Adapun hasil analisis asosiasi untuk mengetahui hubungan antara penyebab kematian bayi menggunakan algoritma apriori yaitu diketahui bahwa jika terjadi kematian bayi yang diakibatkan oleh Sianosis maka kemungkinan juga terjadi penyebab kematian bayi akibat Kelainan Kongenital berdasarkan tingkat kepercayaan sebesar 95%. Aturan ini dapat dijadikan acuan dalam pengambilan keputusan untuk menangani penyebab-penyebab kematian bayi sehingga dapat meminimalisir kejadian kematian bayi.

5.2. Saran

Bebberapa hal yan penulis sarankan untuk pengembangan selanjutnya, yaitu :

1. Menggunakan *dataset* yang jauh lebih besar sehingga dapat menggunakan *minimum support* yang lebih besar.
2. Pengembangan selanjutnya dapat menggunakan algoritma lainnya yang bisa dijadikan perbandingan hasil antara algoritma sehingga dapat menambah pengetahuan dan memperoleh algoritma yang lebih baik untuk diterapkan.

DAFTAR PUSTAKA

- Azwar Anas, H. (2018). Analisis Algoritma Assosiasi untuk Mendapatkan Pola Pemilihan Judul Skripsi Mahasiswa STIE-GK Muara Bulian. *STIE – Graha Karya Muara Bulian/Program Studi Manajemen*, 4.
- Budiati, I. (2016). FAKTOR - FAKTOR YANG BERHUBUNGAN DENGAN KEMATIAN NEONATAL DINI USIA 0 SAMPAI 7 HARI (Studi Kasus di 4 Puskesmas Kabupaten Banjarnegara), 1–70.
- Despitaria, Sujaini, H., & Tursina. (2016). Analisis Asosiasi pada Transaksi Obat Menggunakan Data Mining dengan Algoritma A Priori. *Justin*, 4(2), 6.
- Fricles Ariwisanto Santuri, P. M. (2019). *Data Mining : Teori dan Aplikasi WEKA*. Medan: CV. Rudang Mayang.
- Istiasih, H. (2017). ANALISIS POLA ASOSIASI DATA PENYAKIT HIPERTENSI PADA IBU HAMIL MENGGUNAKAN ALGORITMA APRIORI, 01(04).
- M, K. A., & Thaha, R. M. (n.d.). KOTA MAKASSAR DETERMINANT RISK ANALYSIS OF HEALTH SOCIAL AND BEHAVIOUR AGAINST INFANT MORTALITY INCIDENT IN THE DISTRICT OF UJUNG TANAH MAKASSAR Bagian Promosi Kesehatan , Fakultas Kesehatan Masyarakat , Universitas Hasanuddin , Alamat Korespondensi : Kik.
- Malik, M., Azis, A., Supianto, A. A., & Setiawan, N. Y. (2019). Aplikasi Data Mining Untuk Klasifikasi Kesiapan Skripsi Menggunakan Algoritma Apriori, 3(6), 5556–5564.
- Mandala, E. P. (2017). Pola Frekuensi Judul Skripsi Mahasiswa Teknik Informatika Dengan Algoritma Apriori. *Fakultas Ilmu Komputer, Universitas Putra Indonesia “TPTK” Padang*.
- Purba, C. V., & Bualolo, E. (2020). Implementasi Algoritma Apriori Untuk Menentukan Pola Data Penyakit Pada Anak Usia Dini (Studi Kasus: RS. Estomihi). *JURIKOM (Jurnal Riset Komputer)*, 7(2), 308. <https://doi.org/10.30865/jurikom.v7i2.2113>
- Putra, J. L., Raharjo, M., Sandi, T. A. A., Ridwan, R., & Prasetyo, R. (2019). Implementasi

- Algoritma Apriori Terhadap Data Penjualan Pada Perusahaan Retail. *Jurnal Pilar Nusa Mandiri*, 15(1), 85–90. <https://doi.org/10.33480/pilar.v15i1.113>
- Riszky, A. R., & Sadikin, M. (2019). Data Mining Menggunakan Algoritma Apriori untuk Rekomendasi Produk bagi Pelanggan. *Jurnal Teknologi Dan Sistem Komputer*, 7(3), 103–108. <https://doi.org/10.14710/jtsiskom.7.3.2019.103-108>
- Roiger, R. J. (2017). *Data Mining a Tutorial-Based Primer Second Edition*. New York: CRC Press, Taylor & Francis Group.
- Saputro, G. A. (2017). Perapan Algoritma Apriori Untuk Mencari Pola Penjualan di Cafe.
- Tommy, E. (2017). Penentuan Aturan Asosiasi Pada Transaksi Penjualan Obat Menggunakan Algoritma Apriori (Studi Kasus Pada RSUD Dr. Soetrasno Rembang).
- Zamroni, I., Werdiningsih, I., & Purbandini. (2017). Identifikasi Pola Penyakit Anak Di Bawah Lima Tahun (Balita) dengan Menggunakan Algoritma Apriori. *Seminar Nasional Matematika Dan Aplikasinya*, 362–367.

LAMPIRAN

Lampiran 1 : Tampilan Aplikasi Website

- Tampilan *dataset* yang diinputkan

Hasil Proses Algoritma Apriori

File Dataset : Dataskripsi.xlsx [Kembali ke Halaman Input](#)

Pembacaan File Dataset

	0	1	2	3	4	5	6
1	Sianosis	Kelainan Kongenital	Sepsis	Hygroma Colli			
2	Asfiksia	BBLR	Kelainan Kongenital				
3	Prematur	Sianosis	Kelainan Kongenital				
4	Sianosis	Kelainan Kongenital					
5	Sianosis	Kelainan Kongenital	Prematur				
6	Sianosis	Kelainan Kongenital	BBLR				
7	Prematur	Sianosis	Kelainan Kongenital				
8	Asfiksia	Sianosis	Kelainan Kongenital				
9	BBLR	Kelainan Kongenital	Sianosis				

- Tampilan Calon Kandidat 1-*item* (C1)

Kandidat 1-Item (C1)

	C1 items	C1 Frequent	C1 Support
1	[Sianosis]	768	0.79257
2	[Kelainan Kongenital]	788	0.813209
3	[Sepsis]	111	0.114551
4	[Hygroma Colli]	70	0.072239
5	[Asfiksia]	342	0.352941
6	[BBLR]	391	0.403509
7	[Prematur]	66	0.068111
8	[Placenta Pravia]	22	0.022704
9	[RDS]	316	0.326109
10	[Pendarahan]	18	0.018576

- Tampilan Large 1-*itemset* (L1)

Large l-Itemset (L1)

	L1 items	L1 Frequent	L1 Support
1	{Sianosis}	768	0.79257
2	{Kelainan Kongenital}	788	0.813209

- Tampilan Calon Kandidat 2-item (C2)

	Ck items	Ck Frequent	Ck Support
1	{Kelainan Kongenital, Sianosis}	730	0.753354

- Tampilan Large 2-itemset (L2)

	Lk items	LK Frequent	LK Support
1	{Kelainan Kongenital, Sianosis}	730	0.753354

- Aturan Asosiasi

Aturan Asosiasi				
	Antecedent	Consequence	Confidence	Lift Ratio
1	{Sianosis}	--> {Kelainan Kongenital}	0.950521	1.168851

Lampiran 2 : Data Penyebab Kematian Bayi pada RSIA Sitti Khadijah Cabang Makassar, RSIA Fatimah Makassar, RSIA Bahagia, RSUD Labuang Baji

Data penyebab kematian bayi yang digunakan sebagai data penelitian ini dapat dilihat pada pranala berikut : <https://github.com/ryadita/Dataskripsi>

LEMBAR PERBAIKAN SKRIPSI

“IMPLEMENTASI ALGORITMA APRIORI PADA PENYEBAB KEMATIAN BAYI”

OLEH:

RYA DITA PURNAMA
D42116002

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 12 November 2021.
Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Dr. Ir. Ingrid Nurtanio, M.T	
Sekretaris	Elly Warni, S.T., M.T	
Anggota	Dr. Ir. Amil Ahmad Ilham., ST., M.IT	
	Dr. Indrabayu, S.T., M.T., M.Bus.Sys	

Persetujuan Perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Dr. Ir. Ingrid Nurtanio, M.T	
II	Elly Warni, S.T., M.T	