BAB IV
ANALISIS MODEL

IV.1 Analisis Metode dan Algoritma

Dalam subbab ini akan dibahas mengenai perbandingan beberapa metode dan algoritma data mining yang digunakan dalam mendapatkan model keputusan dalam penentuan matakuliah pilihan jurusan Teknik Informatika UNIKOM.

IV.1.1 Analisis Metode

Data mining adalah proses logis yang digunakan untuk mencari pengetahuan lebih dari sejumlah besar data dengan mengunakan teknik dan algoritma tertentu. Tujuan dari data mining itu sendiri adalah menemukan pola yang sebelumnya tidak diketahui sehingga dapat digunakan dalam proses pendukung pengambilan keputusan. Pola yang ditemukan biasanya akan digunakan untuk memecahkan masalah atau membantu pihak eksekutif dalam mengambil keputusan. Untuk bisa menemukan pengetahuan yang baik dari data yang besar, ada beberapa metode dan algoritma yang dapat dipergunakan. Adapun evaluasi terhadap metode-metode data mining supaya peneliti dapat memilih dan menggunakan teknik yang tepat dalam melakukan penelitian ini. Hasil evaluasinya dapat dilihat pada tabel IV.1.

Tabel IV.1 Evaluasi metode-metode dalam data mining

Metode	Identifikasi
Statistik	Statistik adalah cabang dari ilmu matematika yang berkonsentrasi terhadap koleksi dan deskripsi data.
Clustering	Berkonsentrasi terhadap pengelompokan objek-objek yang memiliki kemiripan dan membandingkannya antar cluster.
Decision Tree	Metode prediksi yang mengklasifikasikan data menggunakan bentuk pohon.
Association Rule	Metode yang mencari keterhubungan antar dua item atau lebih.

Metode	Environment
Statistik	Teknik ini digunakan dalam tempat yang sama dan untuk permasalahan yang sama (prediksi, dan penemuan klasifikasi)
Clustering	Digunakan oleh end user untuk menandai customer dalam database mereka
Decision Tree	Digunakan untuk pengklasifikasian dan estimasi.
Association Rule	Memberikan gambaran relationship dari suatu domain.
Metode	Keuntungan
Statistik	Membantu menjawab beberapa pertanyaan penting tentang data yang akan diolah sebelum memulai untuk proses data mining.
Clustering	Bisa memberikan gambaran jelas tentang apa yang terjadi di dalam database.
Decision Tree	Bisa menangani seluruh jenis variabel termasuk missing value.
Association Rule	Dapat diformulasikan untuk mencari pola yang sekuensial.
Metode	Kerugian
Statistik	Banyak dari data pada masa sekarang yang tidak bisa dijawab dan diolah menggunakan statistic
Clustering	Terkadang proses split dan merge antar cluster mempunyai kesulitan tersendiri.
Decision Tree	Tidak terlalu cocok untuk sejumlah besar tes diagnostik.
Association Rule	Kurang bisa menunjukkan alasan dari pola yang dihasilkan
Metode	Konsekuensi
Statistik	Bisa menghasilkan prediksi yang naive dan bersifat trivial.
Clustering	Hanya memisahkan data secara kemiripan statistical
Decision Tree	Handal dari segi kecepatan pengklasifikasian
Association Rule	Data yang digunakan biasanya berupa data transaksi
Metode	**Implementasi**
Statistik	Histogram dan linear regression
Clustering	Hierarchical Clustering dan partitional Clustering
Decision Tree	CART dan CHAID
Association Rule	*Frequent Item-set*

Dalam penelitian ini, metode yang akan digunakan adalah *Association Rule* dikarenakan dalam penelitian ini diharapkan dapat menghasilkan model keputusan berdasarkan pola keterkaitan antara matakuliah pilihan yang akan ditawarkan dalam satu semester berdasarkan data hasil *polling* (kemauan) dan data histori nilai (kemampuan).

IV.1.2 Analisis Algoritma

Dalam tahapan metode *Association Rule* terbagi menjadi dua tahap yaitu :

1. Pencarian *Frequent Itemset*

   Pencarian *Frequent Itemset* yang diperoleh harus memenuhi minimum support (*Itemset, Support, dan Confidence*).

2. *Rule Generation*

   Frequent Itemset yang telah dihasilkan dari proses sebelumnya digunakan untuk membentuk *Association Rule*. *Association Rule* yang dihasilkan harus memenuhi minimum *support* dan *minimum confidence.*

Pada tahap *Frequent Itemset Candidate Generation* terdapat beberapa kendala yang harus dihadapi untuk memperoleh *Frequent Itemset.* Seperti banyaknya jumlah kandidat yang memenuhi *minimum support*, dan proses perhitungan *minimum support* dari *Frequent Itemset* yang harus melakukan scan database berulang-ulang. Pendekatan algoritma apriori sangat membantu dalam mengurangi jumlah kandidat *Frequent Itemset*. Algoritma apriori dikenal 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*. Kombinasi item yang memenuhi syarat tersebut disebut *frequent itemset*, yang nantinya dipakai untuk membuat aturan-aturan yang memenuhi syarat *confidence minimum.* Algoritma apriori juga memiliki kekurangan yaitu, untuk melakukan pencarian frequent itemset, Algoritma apriori harus melakukan *scanning database* berulang kali untuk setiap kombinasi item. Hal tersebut menyebabkan banyaknya waktu yang dibutuhkan untuk

melakukan *scanning database*. Selain itu, dibutuhkan *generate candidate* yang besar untuk mendapatkan kombinasi item dari *database*.

Algoritma *FP-Growth* merupakan pengembangan dari algoritma *Apriori*. Sehingga kekurangan dari algoritma *Apriori* diperbaiki oleh algoritma *FP-Growth. Frequent Pattern Growth* (*FP-Growth*) adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. Pada algoritma *Apriori* diperlukan *generate candidate* untuk mendapatkan *frequent itemsets*. Akan tetapi, di algoritma *FP-Growth generate candidate* tidak dilakukan karena *FP-Growth* menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemsets*. Hal tersebutlah yang menyebabkan algoritma *FP-Growth* lebih cepat dari algoritma *Apriori*. Karakteristik algoritma *FP-Growth* adalah struktur data yang digunakan adalah *tree* yang disebut dengan *FP-Tree*. Dengan menggunakan *FP-Tree*, algoritma *FP-growth* dapat langsung mengekstrak *frequent Itemset* dari *FP-Tree.*

Berdasarkan analisis algoritma *Association Rule* diatas maka dalam penelitian ini algoritma *FP-Growth* akan digunakan untuk mempercepat proses penentuan *frequent itemset* sebelum men-*generate rule* sebagai rekomendasi keputusan.

IV.2 *Business Understanding*

Pada subbab ini akan dijelaskan mengenai masalah yang terjadi pada suatu jurusan di institusi pendidikan dalam menentukan keputusan terhadap matakuliah pilihan yang akan ditetapkan dalam suatu semester. Dalam kasus ini, program studi Teknik Informatika mempunyai beberapa matakuliah pilihan yang ditawarkan tetapi hanya dua matakuliah pilihan yang akan ditetapkan disemester tujuh atau di semester delapan.

Mengingat jumlah mahasiswa (*student body*) program studi Teknik Informatika yang selalu bertambah setiap tahun akademiknya menyebabkan beberapa masalah dalam penentuan matakuliah pilihan diantaranya keterkaitannya dengan :

a. Jumlah pengajar matakuliah pilihan

b. Jumlah kelas

c. Jumlah ruangan

Sehingga dari beberapa masalah ini akan menyebabkan perubahan kebijakan yang harus diambil oleh Ketua Program Studi Teknik Informatika. Berdasarkan wawancara dan observasi dengan program studi didapat contoh kebijakan sebagai berikut :

a. Pada saat jumlah mahasiswa bertambah akan menyebabkan jumlah kelas dan kebutuhan ruangan juga akan bertambah, maka Ketua Program Studi akan mengambil kebijakan bahwa penentuan matakuliah pilihan akan berdasarkan pada hasil *polling*

b. Sedangkan pada saat jumlah mahasiswa berkurang, maka Ketua Program Studi akan mengambil kebijakan bahwa penentuan matakuliah pilihan hanya berdasarkan pada kebijakan program studi.

Pada sistem berjalan saat ini, penentuan matakuliah berdasarkan sistem *polling* yang disebarkan kepada mahasiswa semester 6 diawal semester 7. Data yang didapatkan dari sistem *polling* akan merekomendasikan pihak jurusan dalam hal ini Ketua Program Studi dalam menentukan matakuliah pilihan. Sistem *polling* ini merupakan cerminan kemauan mahasiswa terhadap matakuliah pilihan. Padahal berdasarkan wawancara dengan beberapa mahasiswa yang mengisi sistem *polling* bahwa pengisian ini pada umumnya hanya mengikuti pengisian yang oleh teman-temannya bukan sepenuhnya berdasarkan kemauan. Disisi lain setiap matakuliah pilihan memiliki matakuliah prasyarat yang harus diambil sebelum

mengikuti matakuliah pilihan ini. Histori nilai ini merupakan cerminan dari kemampuan mahasiswa terhadap matakuliah prasyarat terhadap matakuliah pilihan. Sehingga dalam penelitian ini akan dilakukan rekomendasi pengambilan keputusan berdasarkan irisan dari data *polling* (kemauan) dan histori nilai (kemampuan) terhadap matakuliah pilihan.

Dalam penelitian ini, model keputusan yang dihasilkan akan memberikan rekomendasi bagi pihak jurusan berupa *rule* matakuliah pilihan. Hasil model keputusan ini berdasarkan penjelasan diatas harus dapat menangani seluruh kebijakan yang akan diambil oleh pihak program studi. Model keputusan ini harus dapat menerima data acuan baik dari data *polling* (kemauan), data histori nilai (kemampuan), gabungan antara data *polling* (kemauan) dan data histori nilai (kemampuan), atau apapun data yang akan dijadikan acuan dalam penentuan matakuliah pilihan. Model keputusan ini juga dapat melakukan perbandingan *rule* dari beberapa data acuan untuk mendapatkan irisan sebagai rekomendasi keputusan.

IV.3 Analisis Model Keputusan

Model keputusan adalah model suatu keputusan yang digunakan untuk menguji hasil analisis model keputusan menggunakan *association rule mining* dengan algoritma *FP-Growth* terhadap data *polling* dan histori nilai sehingga menghasilkan rekomendasi penentuan keputusan terhadap matakuliah pilihan.

Secara garis besar penjelasan model keputusan ini adalah :

1. *User* memasukkan data transaksi sebagai data uji ke sistem. Data uji adalah data suatu transaksi yang akan di-*mining* menggunakan teknik *association rule* dengan pendekatan algoritma *FP-growth*. Setiap data traksaksi tersebut akan ditentukan *itemset* yang akan digunakan. *Itemset* inilah yang akan dijadikan dasar proses *association rule mining*. Data uji bisa berasal dari suatu file atau data dari suatu basis data.

2. Sistem melakukan proses *association rule mining* menggunakan pendekatan *FP-Growth* sesuai dengan data uji berikut nilai *minimum support* dan *minimum confidence*. Proses ini akan menghasilkan sebuah struktur data *tree* yang disebut *FP-tree*. Setelah pembangkitan struktur *tree*, maka akan dilakukan pengekstrakan *frequent itemsets* secara langsung dari *FP-tree*. Tingkat kevalidan data digunakan untuk memastikan bahwa hasil *frequent itemsets* dapat mewakili kasus yang dihadapi.
3. *Frequent Itemset* dari hasil pendekatan FP-growth akan di-*generate* untuk menghasilkan *rule* berdasarkan nilai *minimum support* dan *minimum confidence* yang telah ditentukan sebelumnya. *Rule* yang dihasilkan oleh keputusan ke pengguna. *Rule* yang dihasilkan akan digunakan oleh *user* sebagai bahan masukan/rekomendasi dan diproses dengan menggunakan pertimbangan, aturan-aturan atau prosedur yang ada pada proses pengambilan keputusan.
4. *Rule* yang dihasilkan dapat dibandingan dengan *rule* yang lain untuk mendapatkan irisan sebagai rekomendasi keputusan.
5. *Rule* hasil irisan dapat disajikan kepada *user* dalam bentuk file text (*.txt) atau *database* (*.sql).

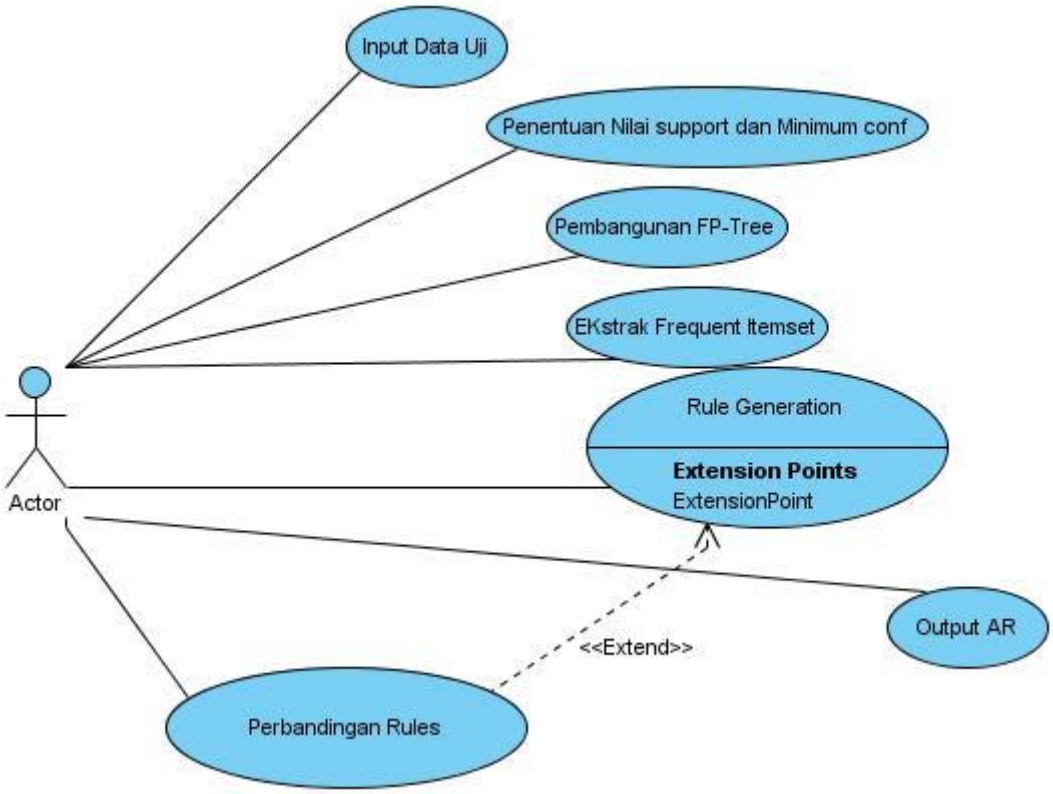
IV.3.1 Batasan-batasan dan Asumsi

Batasan-batasan dan asumsi yang berlaku dalam model keputusan ini diantaranya adalah:

1. Model keputusan ini hanya akan melakukan proses *association rule mining* dengan pendekatan *FP-Growth* saja, sedangkan untuk pengambilan keputusan tetap dilakukan oleh *user* itu sendiri, terutama pada bagian saat penentuan data transaksi yang akan digunakan. Oleh karena itu kemampuan penguasaan *user* terhadap kasus yang dihadapi sangat diperlukan.

2. *User* dalam model keputusan hanya dapat menerima masukan data dari file text atau dari basis data berupa data transaksional. Pembacaan data dari basis data dapat dilakukan dari basis data apapun dengan melalui koneksi ODBC (*Open Database Connectivity*).
3. Model keputusan yang akan dibangun akan menggunakan proses *association rule mining* dengan pendekatan algoritma *FP-Growth*. Algoritma *FP-Growth* dipilih karena algoritma ini hanya akan melakukan pemeriksaan kedalam *database* dua kali saja dalam menghasilkan *frequent itemset* untuk men-*generate rule* yang diinginkan. Dengan algoritma ini kita tidak perlu berulang-ulang memeriksa *database* dalam menghasilkan *frequent itemset*.
4. Model keputusan ini dapat melakukan irisan dari dua *rule* yang dihasilkan dari masing-masing data acuan.

Berdasarkan analisis dan batasan model diatas maka diagram use case model keputusan dapat dilihat pada Gambar 4.1 di bawah ini.



Gambar IV.1 Diagram Use case Model Keputusan Penentuan Matakuliah Pilihan

Ada 7 use case utama yang ada dalam model keputusan ini adalah :

1. Use case Input Data Uji
2. Use case Penentuan Nilai *Minimum Support* dan *Minimum Confidence*
3. Use case Pembangunan *FP-Tree*
4. Use case Ekstrak *Frequent Itemset*
5. Use case *Rule Generation*
6. Use case *Output* AR
7. Use case Perbandingan *Rules*

IV.3.2 Realisasi *Use Case*

Berdasarkan diagram *use case* diatas, penjelasan dari masing-masing *use case* dapat dilihat pada skenario dibawah ini.

IV.3.2.1 Skenario Use Case Input Data Uji

Nama	Input Data Uji	
Deskripsi	Use case ini menggambarkan proses input data uji dalam sistem.	
Aktor	Pengguna	
Skenario Utama		
Kondisi awal	- Sistem menampilkan layar input data uji. - Sistem menampilkan pilihan sumber data uji dari database atau dari file text.	
Aksi Aktor		**Reaksi Sistem**
1. Pengguna memilih sumber data uji		2. Sistem membaca data dan menampilkannya di layar.
		3. Sistem menampilkan pilihan menyimpan data uji atau membatalkan pembacaan datanya.
4. Pengguna memilih menyimpan data uji.		5. Sistem menyimpan data uji ke dalam database, dan layar input data ditutup dan kembali ke pengolahan proyek AR.
Skenario Alternatif : Pemilih memilih pembatalan pembacaan data		
		6. Sistem tidak menyimpan data uji ke dalam database, dan kembali ke pengolahan proyek AR.
Kondisi akhir	Data uji untuk proses pengolahan AR telah tersimpan di dalam database sehingga dapat digunakan untuk proses analisis AR.	

IV.3.2.2 Skenario *Use Case* Penentuan Nilai *Minimum Support* dan *Minimum Confidence*

Nama	Penentuan Nilai *Minimum Support* dan *Minimum Confidence*
Deskripsi	Use case ini menggambarkan proses penentuan nilai *minimum support* dan *minimum confidence* yang akan digunakan dalam analisis AR.
Aktor	Pengguna
Skenario Utama	
Kondisi awal	- Data uji telah dibaca

	- Sistem menampilkan jumlah frekuensi masing-masing item data yang dibaca dari obyek uji.

Aksi Aktor	**Reaksi Sistem**
1. Pengguna memilih data nilai *minimum support* dan *minimum confidence* yang akan ditentukan	2. Sistem menampilkan layar pengisian nilai *minimum support* dan *minimum confidence*
3. Pengguna mengisi *minimum support* dan *minimum confidence*	4. Sistem menyimpan nilai *minimum support* dan *minimum confidence* ke basis data.

Kondisi akhir	Nilai *minimum support* dan *minimum confidence* telah ditentukan dan disimpan dalam basis data dan dapat digunakan dalam proses AR.

IV.3.2.3 Skenario *Use Case* Pembangunan *FP-Tree*

Nama	Pembangunan FP-Tree
Deskripsi	Use case ini menggambarkan proses pembangunan FP-Tree
Aktor	Pengguna
Skenario Utama	
Kondisi awal	- Data uji yang akan dianalisis telah tersedia dan *minimum support* dan *minimum confidence* telah ditentukan

Aksi Aktor	**Reaksi Sistem**
1. Pengguna memilih untuk melakukan pembangunan FP-Tree	2. Sistem akan menghitung jumlah frekuensi item dari data uji berdasarkan *minsup* dan *minconf*
	3. Sistem akan melakukan pembangunan FP-Tree
	4. Sistem menampilkan hasil pembangunan FP-Tree

Kondisi akhir	Pembangunan FP-Tree telah dilakukan dan pengguna dapat melihatnya di layar hasil FP-Tree yang dibangun

IV.3.2.4 Skenario Use Case Ekstrak *Frequent Itemset*

Nama	Ekstrak *Frequent Itemset*	
Deskripsi	Use case ini menggambarkan proses ekstrak *frequent itemset*	
Aktor	Pengguna	
Skenario Utama		
Kondisi awal	- FP-Tree telah dibangun berdasarkan data uji dan *minimum support* serta *minimum confidence*	
Aksi Aktor		**Reaksi Sistem**
1. Pengguna memilih untuk melakukan ekstrak *frequent itemset*		2. Sistem akan mengkombinasikan setiap suffix dari item yang *frequent*
		3. Sistem menyimpan hasil ekstrak *itemset* yang *frequent* kedalam basis data
		4. Sistem menampilkan hasil *itemset* yang *frequent*
Kondisi akhir	Proses ekstrak *frequent itemset* telah dilakukan, data *itemset* yang *frequent* telah disimpan di basis data dan pengguna dapat melihatnya di layar output AR.	

IV.3.2.5 Skenario Use Case *Rule Generation*

Nama	*Rule Generation*	
Deskripsi	Use case ini menggambarkan proses membangkitkan *rule* berdasarkan hasil *frequent itemset*	
Aktor	Pengguna	
Skenario Utama		
Kondisi awal	- Frequent itemset telah dihasilkan	
Aksi Aktor		**Reaksi Sistem**
1. Pengguna memilih *rule generation*		2. Sistem akan mengkombinasikan hasil *frequent itemset*

	1. Sistem akan mengeliminasi hasil kombinasi yang tidak memenuhi *minsup* dan *minconf*
	2. Sistem menyimpan hasil kombinasi yang memenuhi *minsup* dan *minconf*
	3. Sistem menampilkan hasil *rule*
Kondisi akhir	Proses *rule generation* telah dilakukan, hasil *rule generation* telah disimpan di basis data dan pengguna dapat melihatnya di layar output AR.

IV.3.2.6 Skenario *Use Case Output AR*

Nama	Output AR	
Deskripsi	Use case menggambarkan proses output *rule generation* hasil proses AR. Hasil proses AR dapat disimpan/di*export* ke basis data lain atau ke file.	
Aktor	Pengguna	
Skenario Utama		
Kondisi awal	- *Rule* yang berasal dari proses analisis AR telah didapatkan. - Sistem akan menampilkan pilihan format *export* data yang akan dilakukan (file teks atau file sql).	
Aksi Aktor		**Reaksi Sistem**
1. Pengguna memilih format *export* data yang dilakukan dan nama file tujuan.		
Skenario Alternatif : Pengguna memilih *export* ke file SQL		
		2. Sistem akan menyimpan hasil AR ke file sql
Skenario Alternatif : Pengguna memilih *export* ke file Text		
		3. sistem akan menyimpan hasil AR ke file text
Kondisi akhir	Data hasil AR telah di*export* sesuai format yang dipilih.	

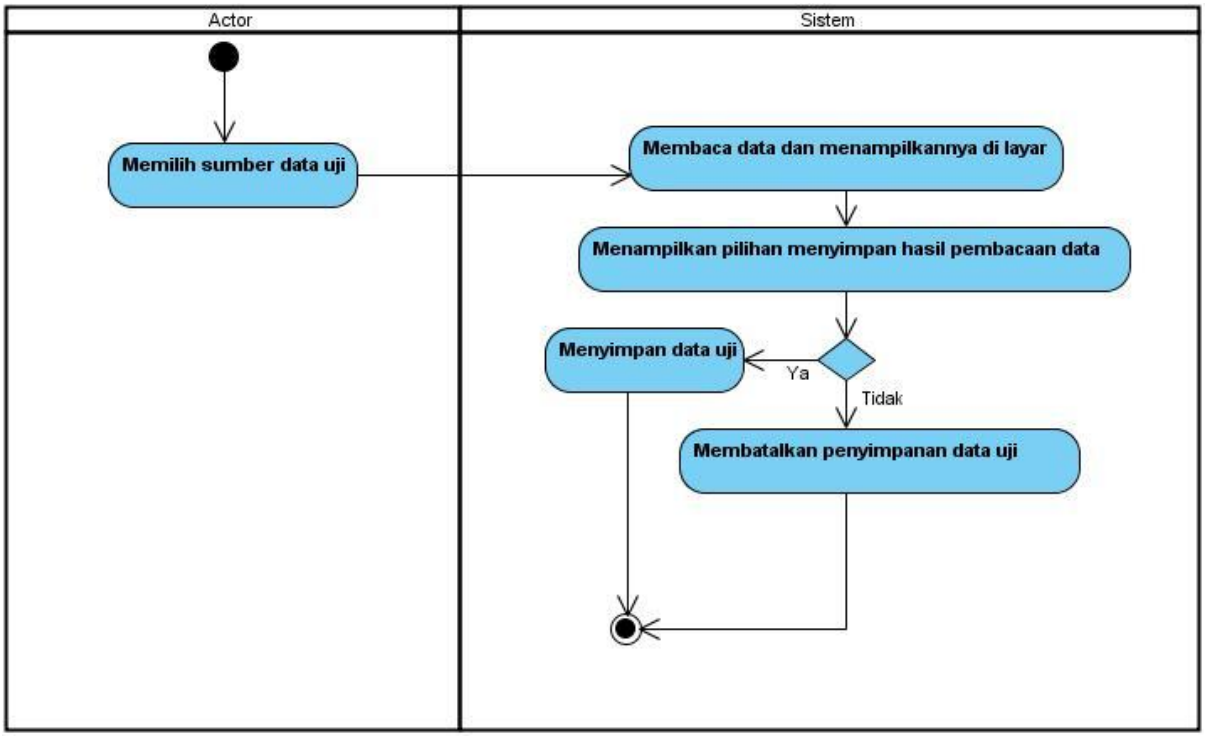
IV.3.2.7 Skenario Use Case Perbandingan *Rules*

Nama	Perbandingan *Rules*	
Deskripsi	Use case ini menggambarkan proses perbandingan antara dua *rule* untuk mendapatkan irisan rule dari beberapa proyek .	
Aktor	Pengguna	
Skenario Utama		
Kondisi awal	Data proyek AR yang pernah dilakukan ada di basis data	
Aksi Aktor		**Reaksi Sistem**
1. Pengguna memilih proyek AR yang akan dibandingkan.		2. Sistem menampilkan menampilkan projek AR yang telah dihasilkan
		3. Sistem mencari irisan rule dari projek AR yang dipilih
		4. Sistem menampilkan hasil irisan *rule* proyek AR
5. Pengguna dapat memilih mengekspor hasil irisan dalam bentuk file text		
Kondisi akhir	Sistem akan menampilkan hasil irisan *rule* atau mengekspor hasil irisan dalam bentuk file text.	

IV.3.3 Realisasi Diagram Aktivitas

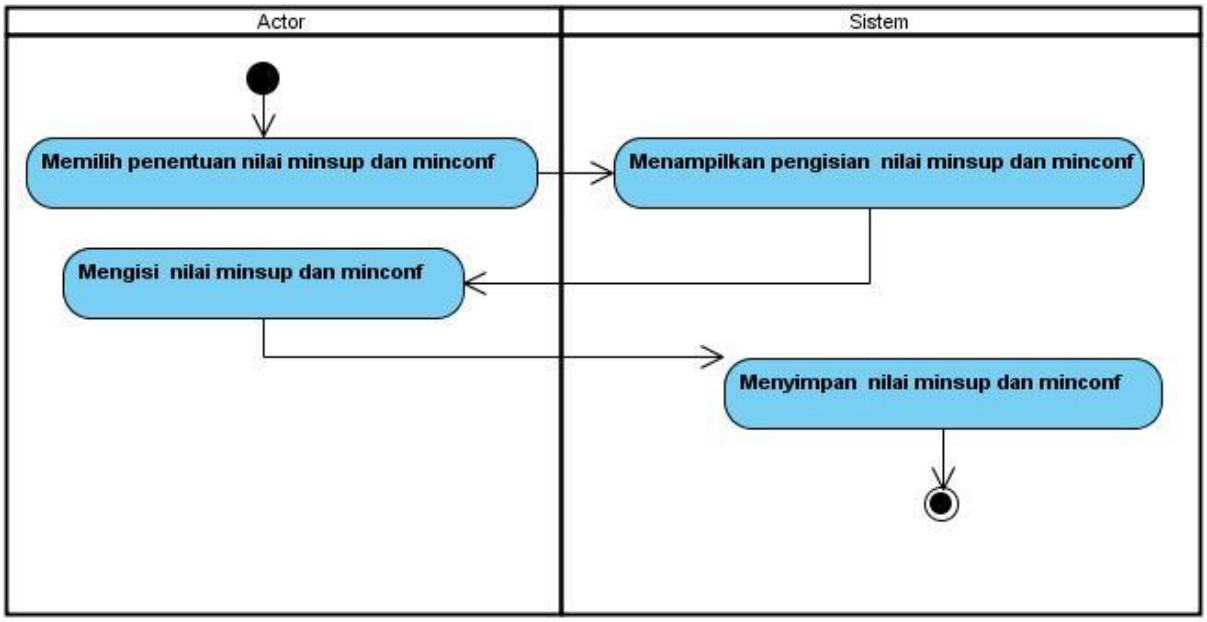
Berdasarkan skenario masing-masing use case diatas, maka penjelasan diagram aktivitasnya dapat dilihat pada gambar dibawah ini.

IV.3.3.1 Diagram Aktivitas *Input* Data Uji



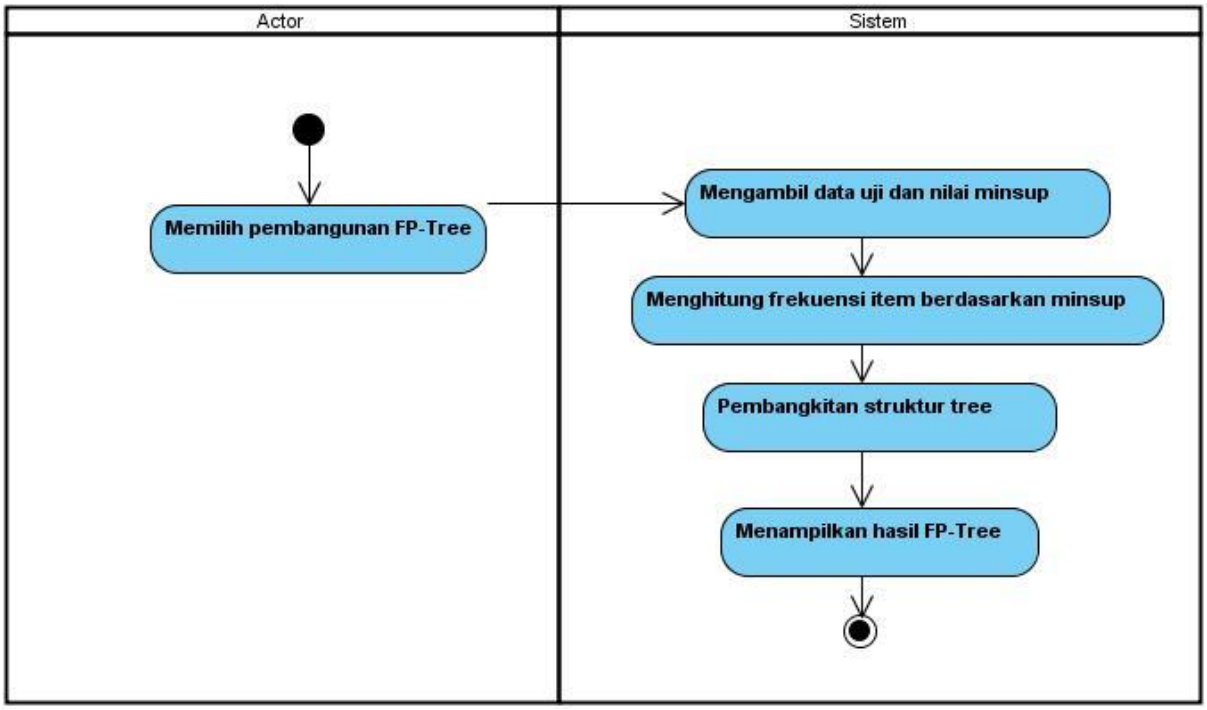
Gambar IV.2 Diagram aktivitas Input Data Uji

IV.3.3.2 Diagram Aktivitas Penentuan Nilai *Minsup* dan *Minconf*



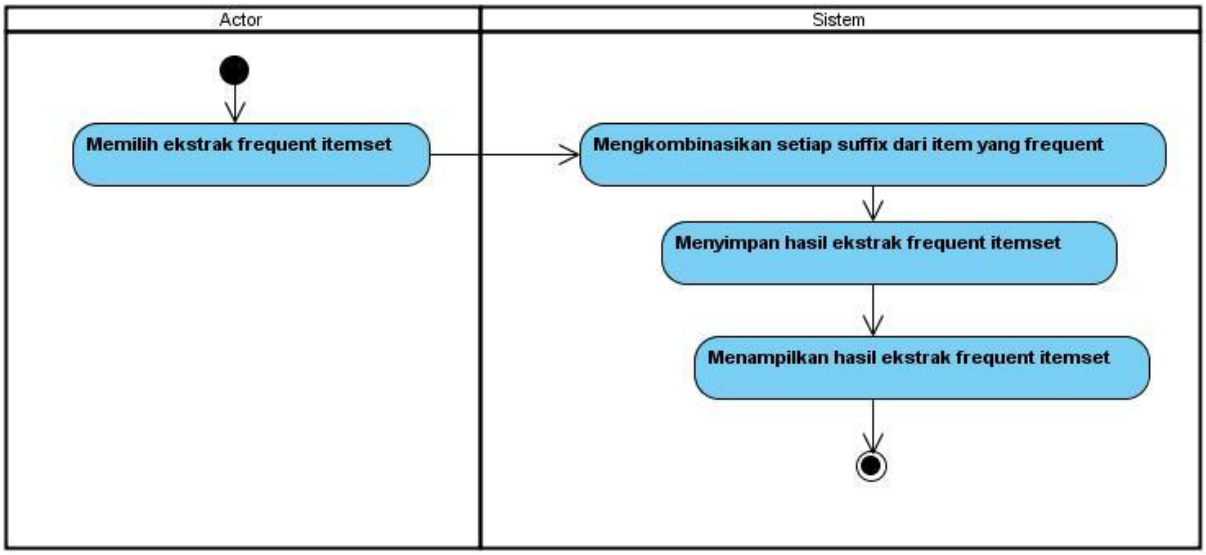
Gambar IV.3 Diagram aktivitas Penentuan Nilai *Minsup* dan *Minconf*

IV.3.3.3 Diagram Aktivitas Pembangunan FP-Tree



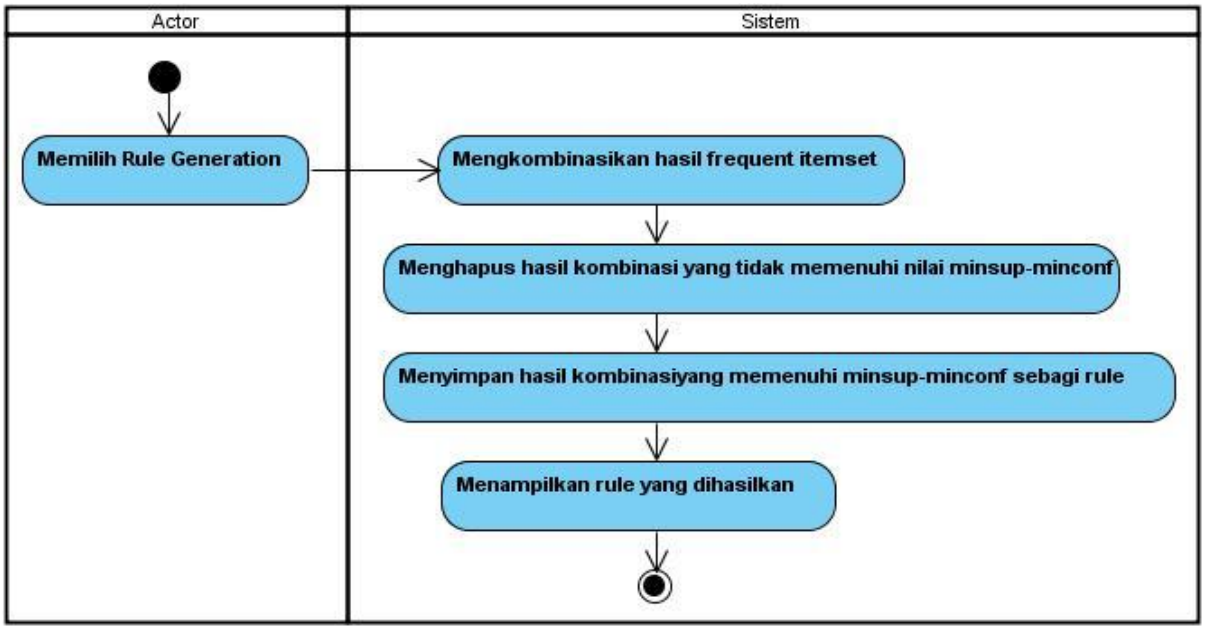
Gambar IV.4 Diagram aktivitas Pembangunan FP-Tree

IV.3.3.4 Diagram Aktivitas Ekstrak *Frequent Itemset*



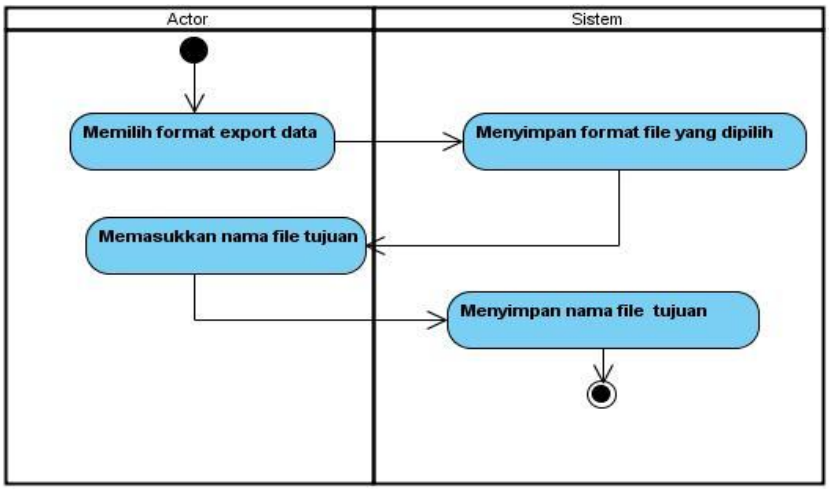
Gambar IV.5 Diagram aktivitas Ekstrak *Frequent Itemset*

IV.3.3.5 Diagram Aktivitas *Rule Generation*



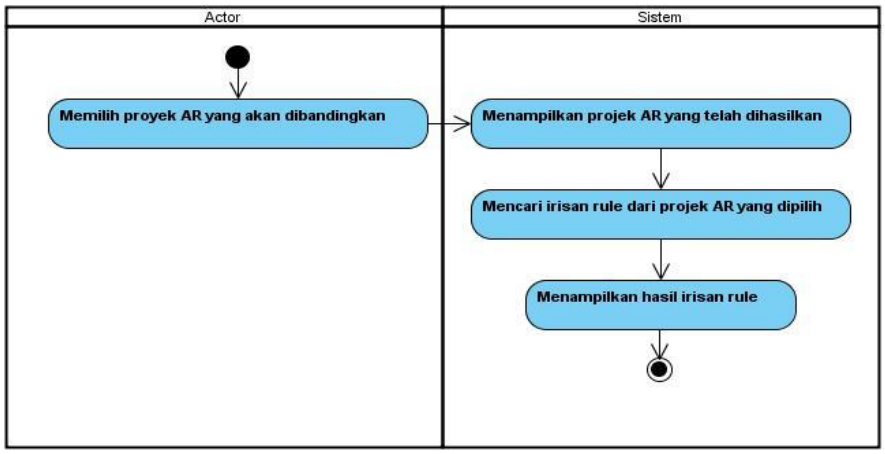
Gambar IV.6 Diagram aktivitas *Rule Generation*

IV.3.3.6 Diagram Aktivitas *Output AR*



Gambar IV.7 Diagram aktivitas *Output AR*

IV.3.3.7 Diagram Aktivitas Perbandingan *Rules*

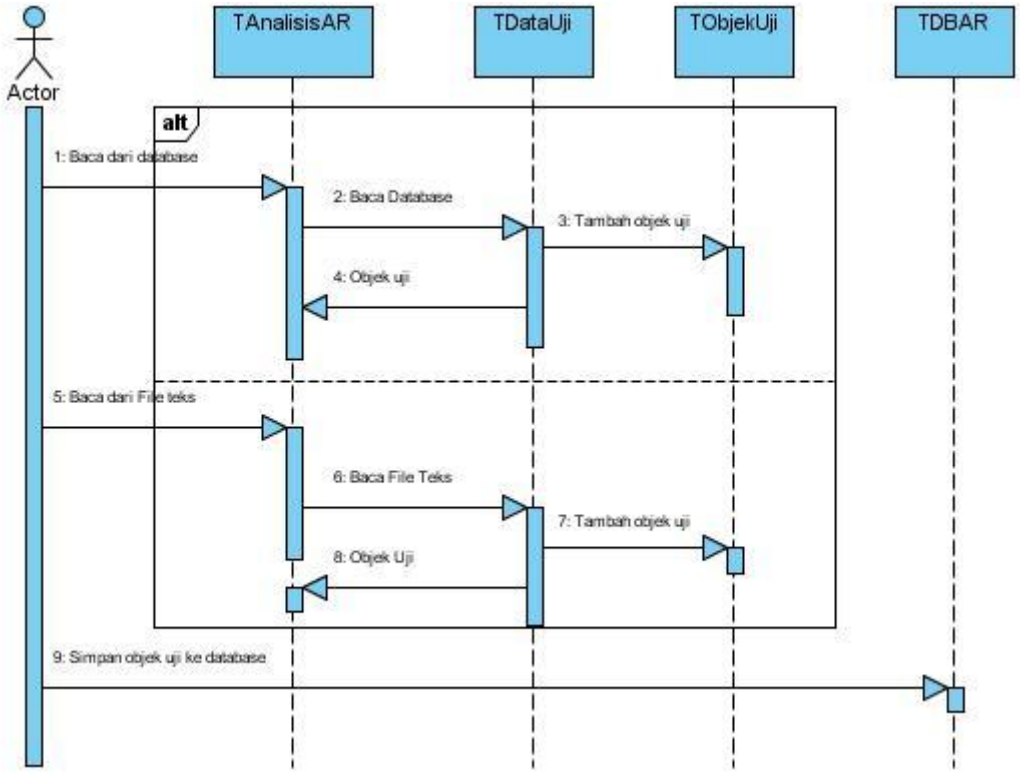


Gambar IV.8 Diagram aktivitas Perbandingan *Rules*

IV.3.4 Realisasi *Sequence*

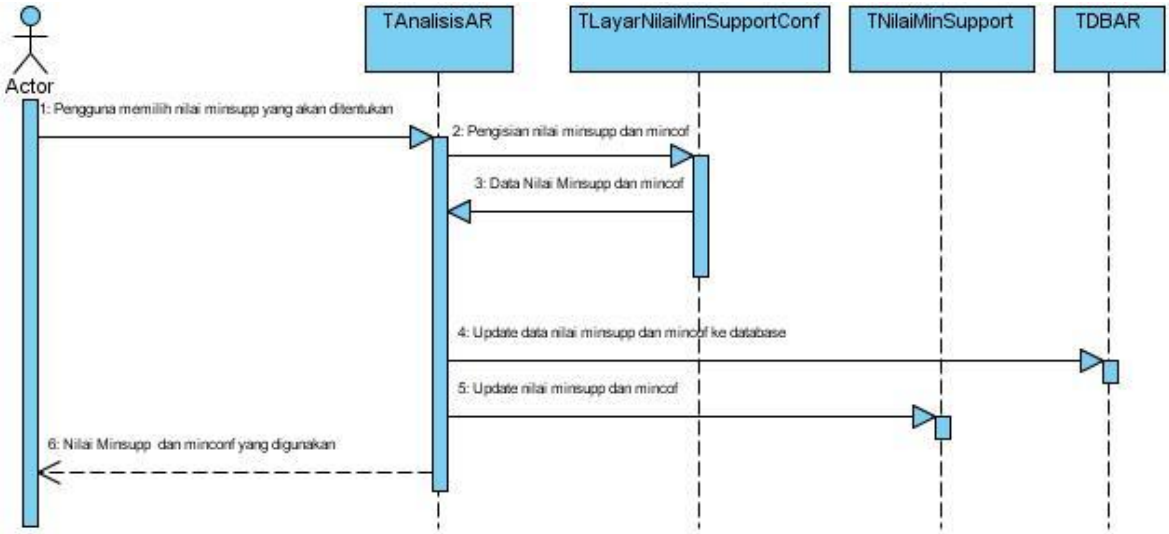
Berdasarkan skenario masing-masing *use case* dan *diagram activity* diatas, maka penjelasan *sequence* dapat dilihat pada gambar dibawah ini.

IV.3.4.1 Sequence *Input* Data Uji



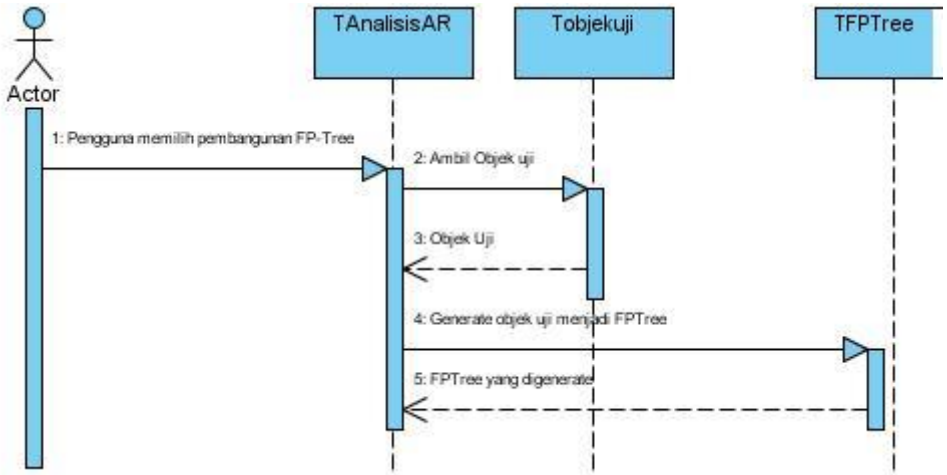
Gambar IV.9 Sequence Input Data Uji

IV.3.4.2 Sequence Penentuan Nilai *Minsup* dan *Minconf*



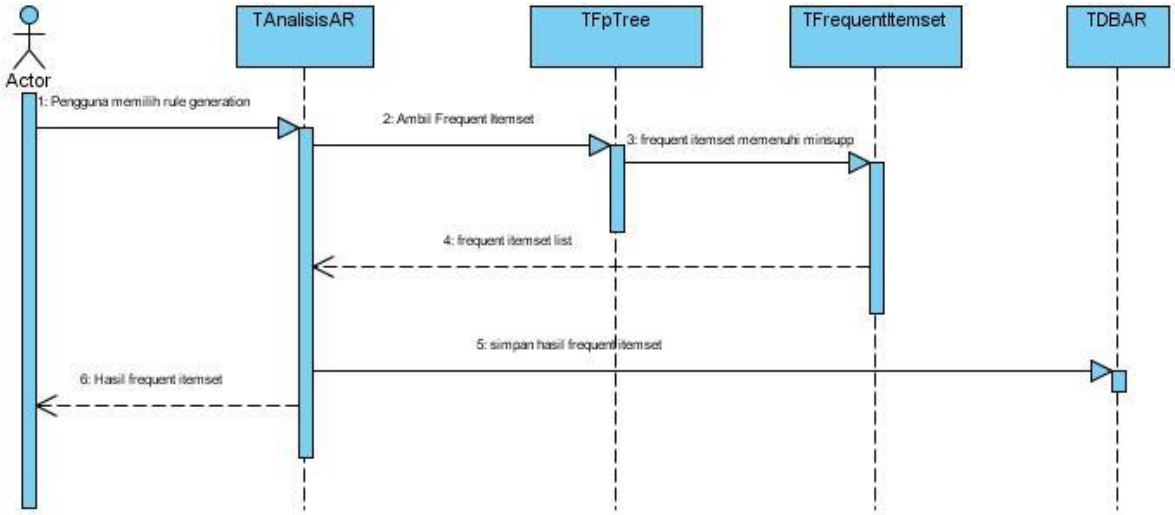
Gambar IV.10 Sequence Penentuan Nilai *Minsup* dan *Minconf*

IV.3.4.3 Sequence Pembangunan FP-Tree



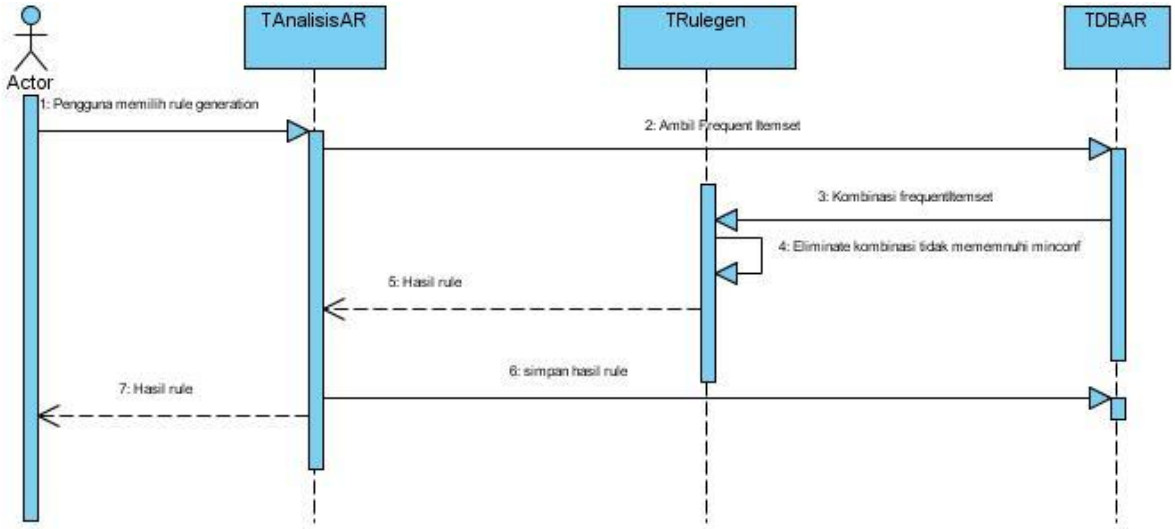
Gambar IV.11 Sequence Pembangunan FP-Tree

IV.3.4.4 Sequence Ekstrak *Frequent Itemset*



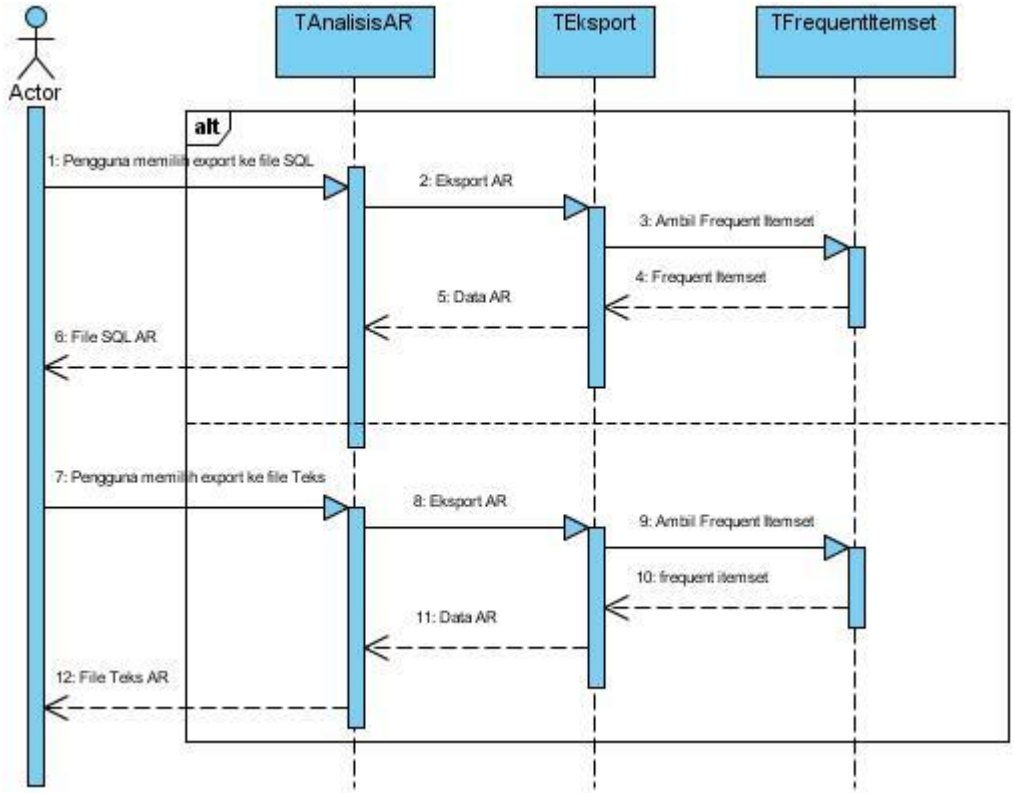
Gambar IV.12 Sequence Ekstrak *Frequent Itemset*

IV.3.4.5 Sequence *Rule Generation*



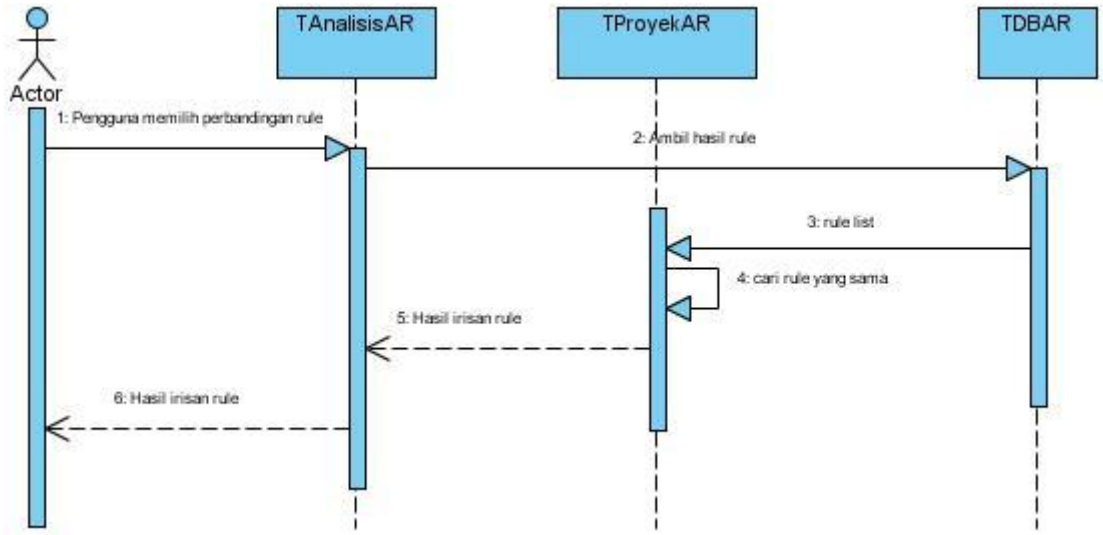
Gambar IV.13 Sequence *Rule Generation*

IV.3.4.6 Sequence *Output AR*



Gambar IV.14 Sequence *Output AR*

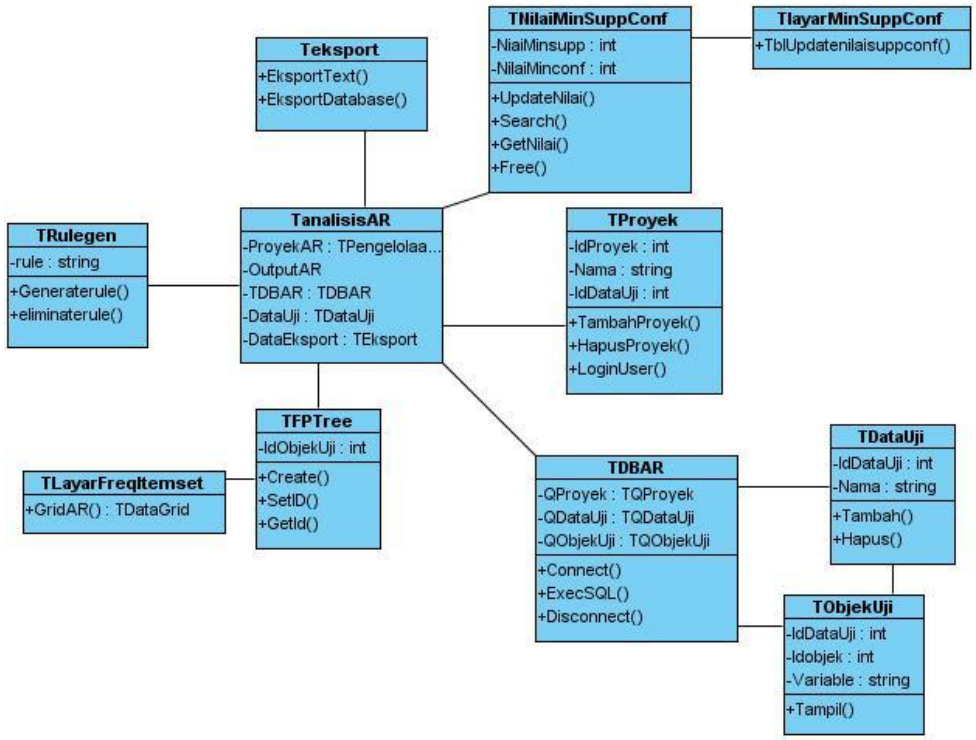
IV.3.4.7 Sequence Perbandingan *Rules*



Gambar IV.15 Sequence Perbandingan Rules

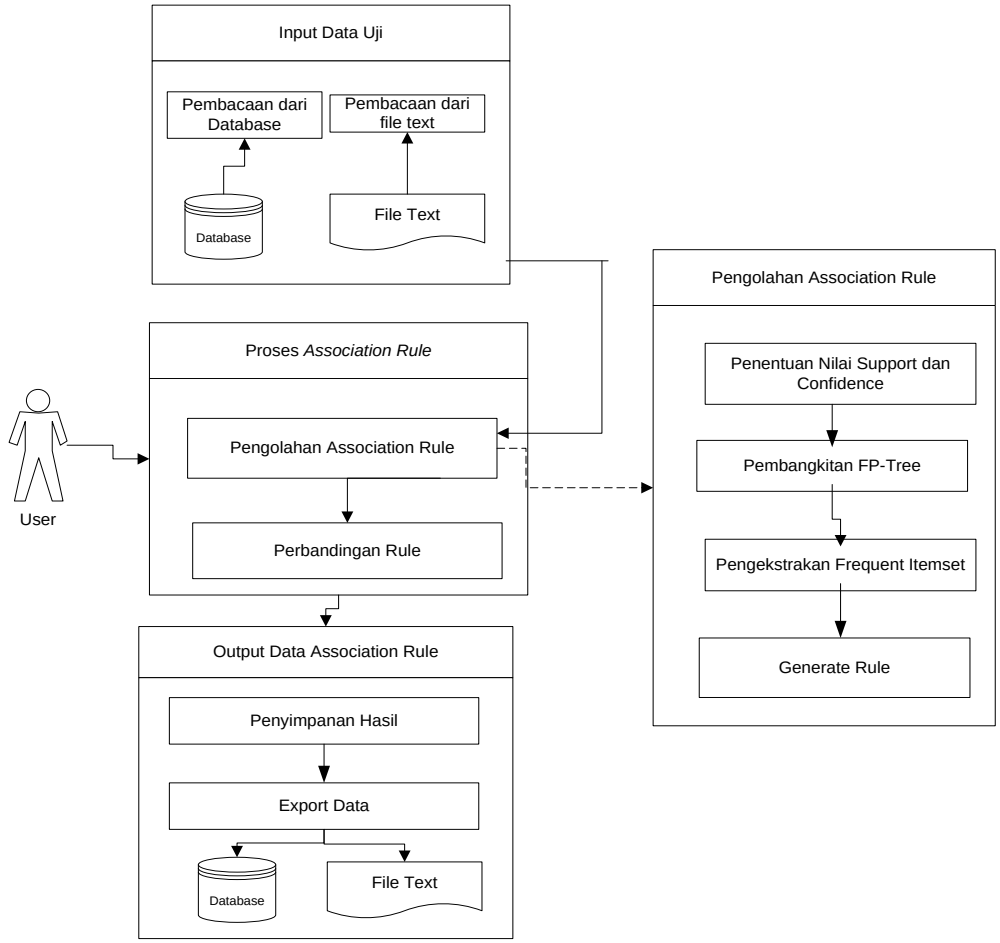
IV.3.5 *Class Diagram*

Adapun *class diagram* yang terbentuk berdasarkan *sequence* diatas dapat dilihat pada gambar IV.16



Gambar IV.16 ***Class Diagram Model Keputusan Penentuan Matakuliah Pilihan***

Secara umum model keputusan yang dibangun dapat dilihat pada gambar arsitektur keputusan dibawah ini.



Gambar IV.17 Arsitektur Model Keputusan *Association Rule*

Secara garis besar arsitektur model keputusan yang akan dibangun terdiri dari 4 modul yaitu :

1. Modul *Input* Data Uji

   Modul ini berguna untuk mengatur *input* data uji. Modul ini terdiri dari beberapa proses, yaitu :

   a. Pembacaan Data dari *Database*

      Proses ini berguna untuk melakukan pembacaan data uji yang bersumber dari suatu *database*. Pembacaan dilakukan dengan melakukan *query* data ke *database* tersebut.

b. Pembacaan Data dari File

Proses ini berguna untuk melakukan pembacaan data dari file. Format file yang bisa dibaca adalah file text.

2. Modul Inisialisasi Proses *Association Rule*

Modul ini berguna untuk mengelola proses *association rule*. Ada beberapa proses yang ada dalam modul ini, yaitu :

a. Pengolahan proses *association rule*

Proses ini berguna untuk mengolah data uji *association rule* yang akan dilakukan. Proses ini mempunyai hubungan ke modul *input* data uji untuk melakukan pembacaan data, modul *association rule*, dan modul *output* data *association rule* untuk melakukan penyimpanan hasil *association rule* ke basis data atau ke target lain seperti ke file text.

b. Perbandingan *rule*

Proses ini berguna untuk melakukan perbandingan antara dua *rule* untuk mendapatkan irisan sebagai rekomendasi keputusan.

3. Modul Analisa *Association rule*

Modul ini berguna untuk melakukan proses *association rule* dengan menggunakan algoritma *FP-Growth*. Modul ini terdiri dari beberapa proses yaitu :

a. Penentuan nilai *minimum support* (*minsupp*) dan *minimum confidence (minconf)*

Proses ini berguna untuk melakukan penentuan terhadap nilai *minimum support* dan *minimum confidence* yang akan digunakan pada proses *association rule*.

b. Proses pembangkitan *FP-Tree*

Proses ini berguna untuk melakukan pembangkitan *FP-tree* untuk menghasilkan struktur data *tree* dari *item* ke *database*.

c. Proses pengekstrakan *Frequent Itemset*

Proses ini berguna untuk mengekstrak *frequent itemset* dari struktur data *tree* yang telah dibangkitkan dalam *FP-tree*. Dengan cara mengekstrak *prefix path sub-trees* dalam setiap *item*.

d. Proses *Rule Generation*

Proses ini berguna untuk mendapatkan *rule* dari hasil ekstrak *frequent itemset* berdasarkan *minsupp* dan *minconf* yang telah ditentukan sebelumnya.

4. Modul perbandingan *Rules*

Modul ini berguna untuk melakukan perbandingan antara *rule* yang telah dihasilkan masing-masing projek (antara 2 projek) untuk mendapatkan irisan sebagai rekomendasi keputusan.

5. Modul output data *Association Rule*

Modul ini berguna untuk mengatur data hasil proses *association rule*. Proses yang ada dalam modul ini adalah :

a. Penyimpanan hasil proses *association rule* ke *database*

Proses ini digunakan untuk melakukan penyimpanan data hasil proses ke basis data pada keputusan ini. Data yang disimpan ke basis data ini dapat diolah kembali atau dihapus.

b. *Export* Data

Export data hasil *association rule* terbagi menjadi 2 yaitu *export* data ke basis data lain yaitu dengan menyusun perintah SQL dan ekspor data ke file misalnya ke file text

Dengan melakukan proses *association rule mining* dengan pendekatan *FP-Growth* terlebih dahulu, maka proses pengambilan keputusan bisa dilakukan dengan lebih cepat karena data uji menjadi lebih sederhana karena telah diekstrak menjadi *frequent itemset* dan menghasilkan *rule* sebagai rekomendasi keputusan.