# BAB III
# ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang analisis dan perancangan dalam membangun aplikasi *data mining* untuk mengetahui tingkat kelulusan mahasiswa di jurusan Sistem Komputer. Analisis meliputi analisis masalah, analisis data, analisis algoritma apriori, analisis fungsional, analisis non fungsiaonal serta perancangan aplikasinya.

## 3.1 Analisis Masalah

UNIKOM merupakan salah satu perguruan tinggi swasta yang berada di kota bandung yang memiliki beberapa jurusan, salah satunya adalah jurusan Sistem Komputer. Di UNIKOM khususnya di Jurusan Sistem Komputer tingkat kelulusan mahasiswa masih terbilang rendah dengan Indeks Prestasi yang juga terbilang kecil, dengan masa studi lebih dari 5 tahun dari yang dijadwalkan yaitu 4 tahun. Oleh karena itu dengan memanfaatkan data akademik dan data kelulusan mahasiswa penulis mengimplementasikan *data mining* untuk menemukan informasi atau pengetahuan baru sehingga dapat digunakan untuk membantu evaluasi sistem pembelajaran di Sistem Komputer. Informasi yang dibutuhkan adalah nilai mahasiswa dan kategori kelulusan mahasiswa dengan atribut mata kuliah di jurusan Sistem Komputer.

## 3.2 Analisis Data

Dalam penulisan tugas akhir ini akan dicari nilai *support* dan *confidence* dari hubungan kategori kelulusan dengan nilai mata kuliah di jurusan Sistem Komputer. Nilai mata kuliah mahasiswa yang akan dicari hubungannya meliputi mata kuliah Algoritma Pemrograman I, Bahasa Rakitan, Sistem Digital, dan Sistem Mikroprosesor. Adapun yang akan diproses *mining* meliputi :

1. Hubungan Kategori Kelulusan dengan Mata Kuliah

   Hasil dari proses *mining* ini dapat membantu untuk mengetahui bagaimana hubungan kategori kelulusan mahasiswa dengan nilai mata kuliah di jurusan sistem komputer dan dapat dimanfaatkan untuk dapat membantu evaluasi sistem pembelajaran di jurusan Sistem Komputer. Sehingga diharapkan dapat meningkatkan tingkat kelulusan mahasiswa.

### 3.2.1 Sumber Data

Data yang digunakan dalam penulisan tugas ahir ini merupakan data akademik mahasiswa angkatan 2001-2009. Hal ini didasarkan pada kebutuhan data, dengan asumsi bahwa mahasiswa angkatan 2001-2009 akan lulus dari rentang waktu tahun 2005-2013. Data tersebut diperoleh dari sekretariat Jurusan Sistem Komputer. Data yang diambil hanya dari data mahasiswa S1 di Jurusan Sistem Komputer. Adapun atribut dari data tersebut adalah

Tabel 3.1 Tabel Data Akademik Mahasiswa

| ATRIBUT | KETERANGAN |
|---|---|
| **NIM (Nomor Induk Mahasiswa)** | Nomor Induk Mahasiswa atau yang disingkat NIM adalah kode yang dimiliki mahasiswa sebagai nomor unik identitas di Perguruan Tinggi. |
| **Tahun Masuk/ Angkatan** | Merupakan tahun masuk mahasiswa yang bersangkutan di jurusan Sistem Komputer. |
| **Algoritma Pemrograman I** | Merupakan salah satu mata kuliah di jurusan Sistem Komputer |
| **Bahasa Rakitan** | Merupakan salah satu mata kuliah di jurusan Sistem Komputer |
| **Sistem Digital** | Merupakan salah satu mata kuliah di jurusan Sistem Komputer |
| **Sistem Mikroprosesor** | Merupakan salah satu mata kuliah di jurusan Sistem Komputer |

| ATRIBUT | KETERANGAN |
|---|---|
| **Lama Studi** | Merupakan lama studi, dihitung dimulai dari terdaftar sebagai mahasiswa sampai dinyatakan lulus. |
| **IPK** | Indeks Prestasi Kumulatif (IPK) adalah ukuran kemampuan mahasiswa sampai pada waktu tertentu yang dapat dihitung berdasarkan jumlah (satuan kredit semester) SKS mata kuliah yang diambil sampai pada periode tertentu dikalikan dengan nilai bobot masing-masing mata kuliah dibagi dengan jumlah seluruh SKS mata kuliah (Anonim, 2009). |

### 3.2.2 Pembersihan Data

Proses pembersihan data pada tugas akhir di sini adalah pembersihan data yang bersifat manual. Dimana proses pembersihan data dilakukan di luar aplikasi. Data dibersihkan dari data yang memilki *missing value.* Maksud dari *missing value* di sini contohnya adalah data yang tidak lengkap. Maksud dari data tidak lengkap contohnya ada *field* yang kosong. Dan apabila ada *field* yang kosong maka data tersebut akan dihilangkan secara manual. Atau contoh lain dari *missing value* adalah data yang isi dari fieldnya jauh dari data yang seharusnya, misalkan nilai mata kuliah kosong, maka data tersebut akan dihilangkan secara manual.

### 3.2.3 Transformasi Data

Transformasi data merupakan proses pengubahan atau penggabungan data ke dalam format yang sesuai untuk diproses dalam *data mining*. Seringkali data yang akan digunakan dalam proses *data mining* mempunyai format yang belum langsung bisa digunakan, oleh karena itu perlu diubah formatnya. Dalam penulisan tugas akhir ini penulis mencari keterkaitan antara tingkat kelulusan dengan data akademik mahasiswa. Kategori kelulusan mahasiswa dapat dilihat dari lama studi dan IPK (Indeks Prestasi Kumulatif). Dari dua parameter tersebut data diubah menjadi tipe data yang memudahkan untuk diproses. Kategori kelulusan diukur dari lama studi dan IPK, lama studi dikategorikan berdasarkan

peraturan akademik yaitu untuk Program sarjana (S1) adalah 4 tahun (8 semester) dengan beban 144 -160 SKS, dan batas waktu studi paling lama adalah 14 semester terhitung sejak terdaftar sebagai mahasiswa semester I di UNIKOM. Sedangkan IPK dikategorikan berdasarkan predikat kelulusan yaitu:

Tabel 3.2 Predikat Kelulusan

| IPK | YUDISIUM | LAMA STUDI |
|---|---|---|
| **3,50 ≤ IPK ≤ 4,00** | **CUM LAUDE** | **≤ 4 TAHUN** |
| **2,75 ≤ IPK ≤ 3,49** | **SANGAT MEMUASKAN** | **-** |
| **2,00 ≤ IPK ≤ 2,74** | **MEMUASKAN** | **-** |

Dari Tabel 3.2 data kelulusan berdasarkan IPK dapat dikategorikan menjadi tiga yaitu :

1. IPK memuaskan dengan IPK 2,00 – 2,74
2. IPK sangat memuaskan dengan IPK 2,75 – 3,49
3. IPK tipe dengan pujian dengan IPK 3,50 – 4,00

Pengkategorian data kelulusan berdasarkan lama studi yaitu :

1. Sesuai jadwal, bila lama studi 4 tahun atau kurang dari 4 tahun
2. Tidak sesuai jadwal, bila lama studi lebih dari 4 tahun

Dari dua pengkategorian tersebut dapat dibuat kategori berdasarkan kombinasi keduanya, seperti yang dapat dilihat pada Tabel 3.3

Tabel 3.3 Transformasi Data

| KATEGORI | KETERANGAN |
|---|---|
| A1 | Lama Studi 4 tahun atau kurang dari 4 tahun dan IPK 3,50 - 4,00 |
| A2 | Lama Studi 4 tahun atau kurang dari 4 tahun dan IPK 2,75 – 3,49 |
| A3 | Lama Studi 4 tahun atau kurang dari 4 tahun dan IPK 2,00 – 2,74 |
| B1 | Lama Studi lebih dari 4 tahun dan IPK 3,50 - 4,00 |

| KATEGORI | KETERANGAN |
|---|---|
| B2 | Lama Studi lebih dari 4 tahun dan IPK 2,75 – 3,49 |
| B3 | Lama Studi lebih dari 4 tahun dan IPK 2,00 – 2,74 |

Dari kombinasi yang terdapat di Tabel 3.3 terdapat enam tingkatan untuk mengukur tingkat kelulusan mahasiswa.

Sedangkan untuk tabel indeks nilai dapat dilihat pada tabel di bawah ini:

Tabel 3.4 Tabel Nilai

| INDEKS | PREDIKAT |
|---|---|
| A | LULUS, Sangat Baik |
| B | LULUS, Baik |
| C | LULUS, Cukup |
| D | LULUS, Kurang |
| E | TIDAK LULUS |

Tabel 3.4 digunakan untuk menunjukkan kategori kelulusan dalam mata kuliah yang diuji.

Tabel 3.5 Tabel Mata Kuliah

| Kategori | MATA KULIAH | KETERANGAN |
|---|---|---|
| AP | Algoritma Pemrograman I | Singkatan untuk mata kuliah Algoritma Pemrograman I |
| BR | Bahasa Rakitan | Singkatan untuk mata kuliah Bahasa Rakitan |
| SD | Sistem digital | Singkatan untuk mata kuliah Sistem Digital |
| SM | Sistem Mikroprosesor | Singkatan untuk mata kuliah Sistem Mikroprosesor |

Tabel 3.5 digunakan untuk menunjukkan singkatan dari mata kuliah yang akan diuji.

## 3.3 Analisis Algoritma Apriori

Algoritma apriori adalah algoritma paling terkenal untuk menemukan pola frekuensi tinggi. Pola frekuensi tinggi adalah pola-pola item di dalam suatu *database* yang memiliki frekuensi atau *support* di atas ambang batas tertentu yang disebut dengan istilah minimum *support* atau *threshold*. *Threshold* adalah batas minimum transaksi. Jika jumlah transaksi kurang dari *threshold* maka item atau kombinasi item tidak akan diikutkan perhitungan selanjutnya. Penggunaan *threshold* dapat mempercepat perhitungan[6].

Algoritma apriori dibagi menjadi beberapa tahap yang disebut iterasi. Tiap iterasi menghasilkan pola frekuensi tinggi dengan panjang yang sama dimulai dari pass pertama yang menghasilkan pola frekuensi tinggi dengan panjang satu. Di iterasi pertama ini, *support* dari setiap item dihitung dengan men-scan *database*. Setelah *support* dari setiap item didapat, item yang memiliki *support* lebih besar dari minimum *support* dipilih sebagai pola frekuensi tinggi dengan panjang 1 atau sering disingkat 1-itemset. Singkatan k-itemset berarti satu set yang terdiri dari k item[6].

Iterasi kedua menghasilkan 2-itemset yang tiap set-nya memiliki dua item. Pertama dibuat kandidat 2-itemset dari kombinasi semua 1-itemset. Lalu untuk tiap kandidat 2-itemset ini dihitung *support*-nya dengan men-scan *database*. *Support* artinya jumlah transaksi dalam *database* yang mengandung kedua item dalam kandidat 2-itemset. Setelah *support* dari semua kandidat 2-itemset didapatkan, kandidat 2-itemset yang memenuhi syarat minimum *support* dapat ditetapkan sebagai 2-itemset yang juga merupakan pola frekuensi tinggi dengan panjang 2[6].

Contoh proses *mining* untuk mengetahui hubungan tingkat kelulusan dengan nilai mata kuliah adalah seperti contoh pada Tabel 3.6 berikut:

Tabel 3.6 Data Awal

| NIM | Kategori Kelulusan | Nilai Algoritma Pemrograman I |
|---|---|---|
| 1020xxx1 | A2 | B(AP) |
| 1020xxx2 | A2 | B(AP) |
| 1020xxx3 | A1 | A(AP) |
| 1020xxx4 | A1 | A(AP) |
| 1020xxx5 | B2 | A(AP) |
| 1020xxx6 | A1 | A(AP) |
| 1020xxx7 | B2 | B(AP) |
| 1020xxx8 | B2 | D(AP) |
| 1020xxx9 | B2 | E(AP) |
| 1020xx10 | B2 | E(AP) |
| 1020xx11 | B2 | E(AP) |
| 1020xx12 | B2 | E(AP) |
| 1020xx13 | B2 | D(AP) |
| 1020xx14 | B2 | A(AP) |
| 1020xx15 | B1 | C(AP) |

Dari data awal tersebut didapat kandidat pertama (C1) seperti pada Tabel 3.7.

Tabel 3.7 Kandidat Pertama (C1)

| Itemset | *Count* |
|---|---|
| A1 | 3 |
| A2 | 2 |
| B1 | 1 |
| B2 | 9 |
| A(AP) | 5 |
| B(AP) | 3 |
| C(AP) | 1 |
| D(AP) | 2 |
| E(AP) | 4 |

Ditetapkan *threshold* = 3, maka kandidat yang nilainya kurang dari 3 akan dihapus. Sehingga hasil seperti Tabel 3.8 (L1).

Tabel 3.8 Hasil Setelah *Threshold* Ditetapkan (L1)

| **Itemset** | ***Count*** |
|---|---|
| A1 | 3 |
| B2 | 9 |
| A(AP) | 5 |
| B(AP) | 3 |
| E(AP) | 4 |

Dari Tabel 3.8 didapat kandidat kedua (C2) seperti pada Tabel 3.9

Tabel 3.9 Kandidat Kedua (C2)

| **Itemset** | ***Count*** |
|---|---|
| A1, A(AP) | 3 |
| A1,B(AP) | 0 |
| A1,E(AP) | 0 |
| B2,A(AP) | 2 |
| B2,B(AP) | 1 |
| B2,E(AP) | 4 |

Setelah ditetapkan *threshold* menghasilkan data seperti Tabel 3.10

Tabel 3.10 Hasil Kedua (L2)

| **Itemset** | ***Count*** |
|---|---|
| A1, A(AP) | 3 |
| B2, E(AP) | 4 |

Dari Tabel 3.10 dapat diambil hasil sebagai berikut:

*Support* A1, A(AP) = *count* (A1,A(AP))/jumlah transaksi
= 3/15

*Support* B2, E (AP) = *count* (B2, E(AP))/jumlah transaksi
= 4/15

*Confidence* A1, A(AP) = *count* (A1, A(AP))/*count* (A1)
= 3/3

*Confidence* B2, E(AP) = *count* (B2, E(AP))/*count* (B2)
= 4/9

Dapat dilihat bahwa proses *mining* hubungan tingkat kelulusan dengan nilai Algoritma Pemrograman mahasiswa dengan *threshold* 3 menghasilkan hubungan A1, A(AP) mempunyai nilai *support* 3/15 dan *confidence* 3/3 dan hubungan B2,E(AP) mempunyai nilai *support* 4/15 dan *confidence* 4/9. Dari contoh tersebut dapat disimpulkan bahwa mahasiswa yang mempunyai nilai A dalam mata kuliah Algoritma Pemrograman I, maka kategori kelulusannya adalah A1 yaitu lama Studi 4 tahun atau kurang dari 4 tahun dan IPK 3,50 - 4,00, dengan nilai keyakinan 100% dari 20 % seluruh proses dalam *database*. Sedangkan mahasiswa yang mendapat nilai E maka kategori kelulusannya adalah B2 yaitu Lama Studi lebih dari 4 tahun dan IPK 2,75 – 3,49 dengan nilai keyakinan 44,44% dari 26,27 % dari seluruh proses *database*.

Contoh kedua dari proses *mining* untuk mengetahui hubungan tingkat kelulusan dengan nilai mata kuliah Bahasa Rakitan, dapat dilihat dalam contoh Tabel 3. 11 berikut:

Tabel 3.11 Data Awal Contoh Kedua

| NIM | Kategori Kelulusan | Nilai Algoritma Pemrograman I | Nilai Bahasa Rakitan |
|---|---|---|---|
| 1020xxx1 | A2 | B(AP) | B(BR) |
| 1020xxx2 | A2 | B(AP) | C(BR) |
| 1020xxx3 | A1 | A(AP) | D(BR) |
| 1020xxx4 | A1 | A(AP) | B(BR) |
| 1020xxx5 | B2 | A(AP) | E(BR) |
| 1020xxx6 | A1 | A(AP) | A(BR) |
| 1020xxx7 | B2 | B(AP) | C(BR) |
| 1020xxx8 | B2 | D(AP) | B(BR) |
| 1020xxx9 | B2 | E(AP) | E(BR) |
| 1020xx10 | B2 | E(AP) | E(BR) |
| 1020xx11 | B2 | E(AP) | E(BR) |
| 1020xx12 | B2 | E(AP) | D(BR) |
| 1020xx13 | B2 | D(AP) | C(BR) |
| 1020xx14 | B2 | A(AP) | C(BR) |
| 1020xx15 | B1 | C(AP) | B(BR) |

Dari data tersebut didapat kandidat pertama (C1) seperti pada Tabel 3.12 berikut

Tabel 3.12 Kandidat Pertama (C1) contoh kedua

| **Itemset** | ***Count*** |
|---|---|
| A1 | 3 |
| A2 | 2 |
| B1 | 1 |
| B2 | 9 |
| A(AP) | 5 |
| B(AP) | 3 |
| C(AP) | 1 |
| D(AP) | 2 |
| E(AP) | 4 |
| A(BR) | 1 |
| B(BR) | 4 |
| C(BR) | 4 |
| D(BR) | 2 |
| E(BR) | 4 |

Ditetapkan *threshold* = 3, maka kandidat yang nilainya kurang dari 3 akan dihapus, maka akan didapatkan hasil seperti pada Tabel 3.13

Tabel 3.13 Hasil Setelah *Threshold* Ditetapkan (L1) sebagai Contoh Kedua

| **Itemset** | ***Count*** |
|---|---|
| A1 | 3 |
| B2 | 9 |
| A(AP) | 5 |
| B(AP) | 3 |
| E(AP) | 4 |
| B(BR) | 4 |
| C(BR) | 4 |
| E(BR) | 4 |

Dari Tabel 3.13 didapat kandidat kedua (C2) seperti pada Tabel 3.14

Tabel 3.14 Kandidat Kedua (C2) contoh kedua

| **Itemset** | ***Count*** |
|---|---|
| A1,A(AP) | 3 |
| A1,B(AP) | 0 |

| Itemset | *Count* |
|---|---|
| A1,E(AP) | 0 |
| A1,B(BR) | 1 |
| A1,C(BR) | 0 |
| A1,E(BR) | 0 |
| B2,A(AP) | 2 |
| B2,B(AP) | 1 |
| B2,E(AP) | 4 |
| B2,B(BR) | 1 |
| B2,C(BR) | 3 |
| B2,E(BR) | 4 |
| A(AP), B(BR) | 1 |
| A(AP), C(BR) | 1 |
| A(AP), E(BR) | 1 |
| B(AP), B(BR) | 1 |
| B(AP), C(BR) | 2 |
| B(AP), E(BR) | 0 |
| E(AP), B(BR) | 0 |
| E(AP), C(BR) | 0 |
| E(AP), E(BR) | 3 |

Setelah diberikan *threshold* = 3 maka akan menghasilkan data seperti Tabel 3.15:

Tabel 3.15 Hasil Kedua (L2) Contoh Kedua

| Itemset | *Count* |
|---|---|
| A1,A(AP) | 3 |
| B2,E(AP) | 4 |
| B2,C(BR) | 3 |
| B2,E(BR) | 4 |
| E(AP), E(BR) | 3 |

Dari Tabel 3.15 didapat kandidat ketiga (C3) seperti pada Tabel 3.16

Tabel 3.16 Kandidat Ketiga (C3)

| Itemset | *Count* |
|---|---|
| B2,E(AP),E(BR) | 3 |

Dari data tersebut dapat diambil hasil sebagai berikut:

*Support* (B2, E(AP), E(BR)) = *count* (B2, E(AP), E(BR))/jumlah transaksi

= 3/15

*Confidence* (B2, E(AP), E(BR)) = *count* (B2, E(AP), E(BR))/*count* (B2)
= 3/9

Dalam proses tersebut ditetapkan minimum *support* atau *threshold* adalah 3. Pada iterasi pertama, item yang *support*-nya atau *count*-nya kurang dari 3 di eliminasi dari 1-itemset L1. Kemudian kandidat 2-itemset C2 dari iterasi kedua dibentuk dari *cross product* item-item yang ada di L1. Setelah kandidat 2-itemset itu dihitung dari *database*, ditetapkan 2-itemset L2. Proses serupa berulang di iterasi ketiga, tetapi selain (B2, E(AP), E(BR)) yang menjadi kandidat 3-itemset C3 sebenarnya ada juga itemset lain seperti (A1, A(AP), E(BR)), (B2, E(AP, C(BR) yang dapat diperoleh dari kombinasi L2, namun itemset tersebut dipangkas karena tidak memenuhi syarat minimum. Proses ini akan berulang sampai tidak ada lagi kandidat baru yang dapat dihasilkan dari minimum *threshold.* Jadi dari proses tabel kandidat ketiga dapat diketahui bahwa mahasiswa yang mendapatkan nilai E dalam Algoritma Pemrograman dan nilai E dalam Bahasa Rakitan akan lulus dalam kategori B2 dengan Lama Studi lebih dari 4 tahun dan IPK 2,75 – 3,49 dengan nilai keyakinan 33,33 % dengan *support* 20 % proses dalam *database*.

Dari Tabel 3.11 bisa dilihat bahwa algoritma apriori dapat mengurangi jumlah kandidat yang harus dihitung *support*-nya dengan pemangkasan. Misalnya kandidat 3-itemset dapat dikurangi dari 3 menjadi 1 saja. Pengurangan jumlah kandidat ini merupakan sebab utama peningkatan performa algoritma *apriori*.

### 3.3.1 *Report* dan Penyajian Hasil Proses

Setelah proses *mining* akan disajikan hasil dari *data mining* berupa tabel hubungan kekuatan dengan nilai *support* dan *confidence* masing-masing atribut serta *threshold* yang digunakan. Semakin tinggi nilai *confidence* dan *support* maka semakin kuat nilai hubungan antara atribut.

## 3.4 Analisis Kebutuhan Non Fungsional

Analisis merupakan penguraian dari suatu masalah atau objek yang akhirnya menghasilkan suatu kesimpulan. Hal ini dimaksud untuk mengidentifikasi dan mengevaluasi masalah atau objek.

Analisis non fungsional adalah sebuah langkah dimana seorang pembangun perangkat lunak menganalisis sumber daya yang akan menggunakan perangkat lunak yang dibangun.

Analisis non fungsional tidak hanya menganalisis siapa saja yang akan menggunakan aplikasi tetapi juga menganalisis perangkat keras dan perangkat lunak yang dimiliki oleh pemesan, sehingga dapat ditentukan kompabilitas aplikasi yang dibangun terhadap sumber daya yang ada. Setelah melakukan analisis non fungsional, maka dilanjutkan ke langkah berikutnya yaitu menentukan kebutuhan non fungsional sistem yang akan dibangun untuk disesuaikan dengan fakta yang ada.

Apabila terjadi ketidakcocokan antara fakta dan kebutuhan maka perlu adanya penyesuaian fakta terhadap kebutuhan yang ada. Apabila kebutuhan tidak dipenuhi maka system yang dibangun tidak akan berjalan baik sesuai yang diharapkan.

Analisis non fungsional dan kebutuhan non fungsional yang dilakukan dibagi dalam tiga tahapan, yaitu:

1. Kebutuhan pengguna (*user*)
2. Kebutuhan perangkat keras (*hardware*)
3. Kebutuhan perangkat lunak (*software*)

## 3.5 Analisis Kebutuhan Pengguna (User)

Aplikasi tingkat kelulusan mahasiswa ini akan digunakan oleh sekretariat jurusan, dimana keterlibatannya dalam menggunakan aplikasi ini harus mengacu pada spesifikasi pengguna sebagai berikut:

1. Terbiasa menggunakan aplikasi yang ada di sistem operasi Windows.
2. Memiliki pengetahuan tentang *database*.
3. Mengetahui atribut yang dianggap kuat untuk dilibatkan dalam proses *Data mining*

## 3.6 Analisis Kebutuhan Perangkat Keras (Hardware)

Kebutuhan *hardware* yang disarankan untuk menjalankan spesifikasi ini minimal memiliki spesifikasi sebagai berikut:

a. Processor 1,5 GHz
b. Harddisk 20 GB
c. RAM 512 MB
d. Monitor yang mendukung kualitas warna 16 bit dengan resolusi 800 x 600 *pixels*.

Berdasarkan spesifikasi yang telah ada, secara keseluruhan kebutuhan perangkat keras untuk aplikasi ini telah terpenuhi.

## 3.7 Analisis Kebutuhan Perangkat Lunak (Software)

Perangkat lunak merupakan sarana pendukung lainnya bagi pembangunan aplikasi tingkat kelulusan mahasiswa di jurusan Sistem Komputer ini. Perangkat lunak yang disarankan untuk menjalankan aplikasi ini adalah sebagai berikut:

a. Sistem Operasi

   Untuk sistem operasi windows disarankan, karena sistem operasi ini banyak dikenal oleh *user* awam dan lebih mudah dipelajari.

*b.* *Database Management System*

   *Database Management System* (DBMS) adalah sistem untuk mengelola basis data yang digunakan. Untuk aplikasi *Data mining* ini digunakan Microsoft Access sebagai DBMS.

## 3.8 Analisis Kebutuhan Fungsional

Analisis merupakan penguraian dari suatu masalah atau objek yang akhirnya menghasilkan suatu kesimpulan. Hal ini dimaksud untuk mengidentifikasi dan mengevaluasi masalah atau objek. Dalam langkah ini akan dilakukan penentuan entitas-entitas baik entitas internal maupun entitas eksternal, data yang mengalir, serta prosedur-prosedur yang bisa dilakukan oleh masing-masing entitas.

### 3.8.1 Spesifikasi Kebutuhan Fungsional

Spesifikasi kebutuhan fungsional pada Aplikasi *Data Mining* ini merujuk pada kebutuhan akan perancangan *data mining*, seperti berikut ini :

- Dapat menggabungkan data yang akan diproses *mining* dari data matakuliah dan data mahasiswa.
- Dapat memilih data serta atribut yang akan diproses.
- Dapat mengubah data menjadi data yang siap diproses.
- Dapat memproses data untuk di*mining.*
- Dapat menampilkan hasil proses *mining* dengan nilai *support* dan *confidence.*

### 3.8.2 Analisis Batasan Sistem

Batasan dari sistem ini adalah :

1. Proses pengolahan data (penambahan, pengurangan dan hapus data) dilakukan oleh administrator.

### 3.8.3 Diagram Konteks

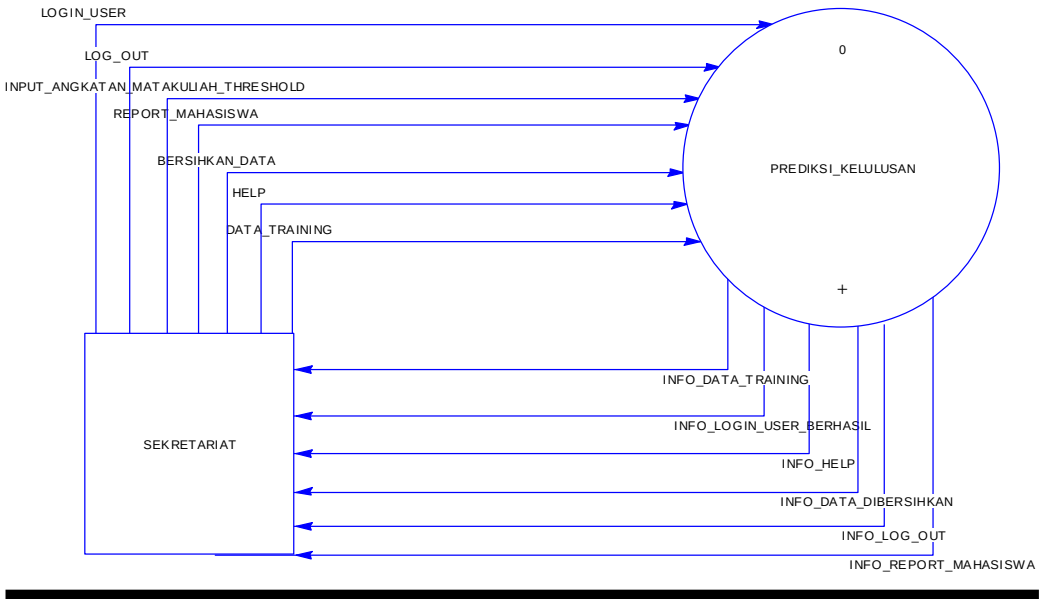
Diagram konteks merupakan salah satu alat bantu dalam melakukan analisis terstruktur. Diagram konteks ini menggambarkan suatu sistem secara garis besarnya atau keseluruhannya saja. Dalam diagram konteks juga

digambarkan entitas eksternal yang merupakan *brainware* yang menghasilkan data yang akan diolah oleh sistem maupun tujuan dari informasi yang dihasilkan oleh sistem.

Gambar di bawah merupakan Diagram konteks pada aplikasi hubungan kategori kelulusan terhadap nilai mata kuliah mahasiswa di jurusan Sistem Komputer menggunakan algoritma apriori.

1. DCD/ DFD Level-0

Gambar 3.1 merupakan Diagram konteks pada Aplikasi *Data mining* tingkat kelulusan mahasiswa menggunakan algoritma apriori.

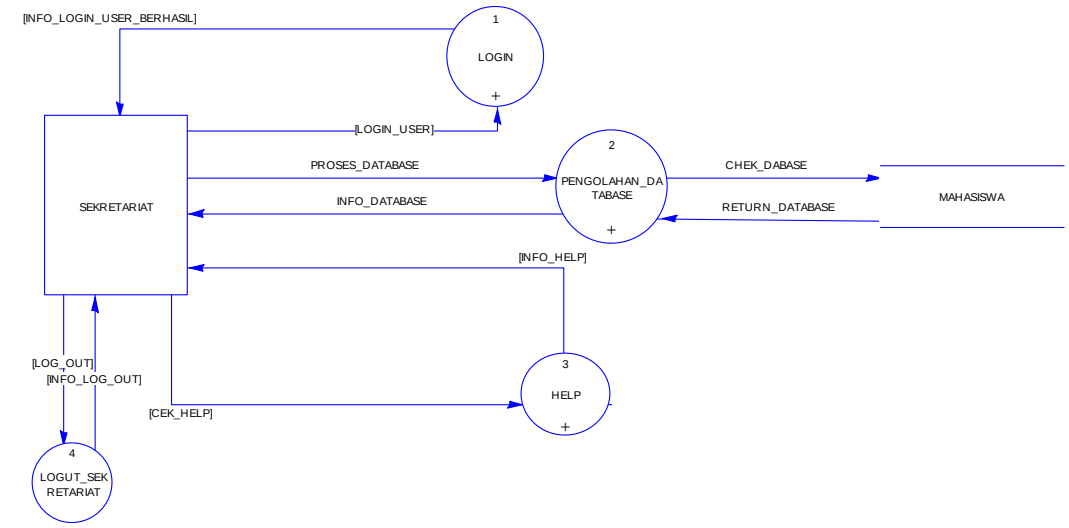



Gambar 3.1 DFD Level-0

2. DFD Level-1

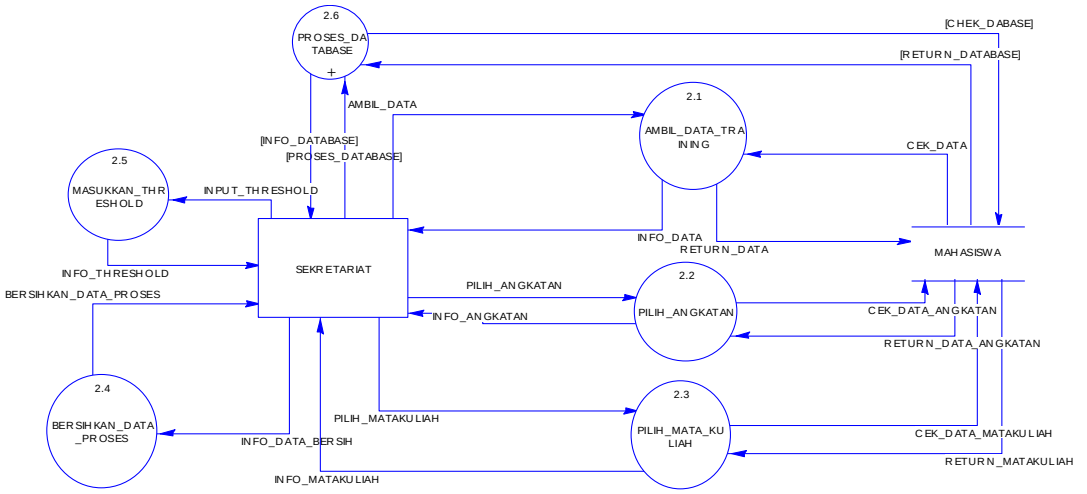
Diagram alir data merupakan sebuah representasi dari suatu sistem yang menggambarkan bagian-bagian dari sistem tersebut beserta keterkaitan antara bagian-bagian yang ada. Dari diagram alir data ini seseorang bisa mengetahui sumber dari informasi di dalam sistem, maupun tujuan dari masukan yang berasal

dari entitas eksternal. Adapun diagram alir data level-1 dari aplikasi *data mining* ditunjukan oleh gambar 3.2 berikut:




Gambar 3.2 DFD Level-1

3. DFD Level-2 untuk Proses 2




Gambar 3.3 DFD Level-2 untuk Proses 2

Gambar 3.3 menjelaskan tentang proses dari DFD level-2 proses 2. Dalam gambar tersebut pengolahan data *training* dipecah menjadi 3 bagian sub proses, dan spesifikasi prosesnya akan dijelaskan di Tabel 3.18.

Tabel 3.17 Spesifikasi *Proses DFD Level-1 Aplikasi Data Mining*

| | Proses | Keterangan |
|---|---|---|
| 1 | No.Proses | 1 |
| | Nama Proses | *Login* |
| | Sumber | Sekretariat |
| | Input | Data *login* user dan password |
| | Output | Data *login* |
| | Deskripsi | Proses untuk dapat mengakses atau menjalankan aplikasi *data mining.* |
| | Logika Proses | Masukkan User dan Password.<br>Login sukses, maka akan menampilkan menu utama dan jika login gagal atau user dan passwordnya salah, maka akan menampilkan pesan. |
| 2 | No.Proses | 2 |
| | Nama Proses | Pengolahan *Database* |
| | Sumber | Sekretariat, Mahasiswa |
| | Input | *Chek database*, Proses *database* |
| | Output | *Info Database, Return database* |
| | Logika Proses | 1. Klik *button* database. |
| 3 | No.Proses | 3 |
| | Nama Proses | *Help* |
| | Sumber | Sekretariat |
| | Input | *Info Help* |
| | Output | *Return* Data H*elp* |
| | Logika Proses | 1. Klik *button Help* |
| 4 | No.Proses | 4 |
| | Nama Proses | *Logout* |
| | Sumber | Sekretariat |
| | Input | Data *logout* |
| | Output | Info *logout* |
| | Logika Proses | Lakukan logout<br>Tampilkan informasi akun sudah logout |

Tabel 3.18 Spesifikasi *Proses DFD Level-2* untuk Proses 2

| | Proses | Keterangan |
|---|---|---|
| 1 | No.Proses | 2.1 |
| | Nama Proses | Ambil Data *Training* |
| | Sumber | Mahasiswa, Sekretariat |
| | Input | Ambil data *training*, return data *training* |
| | Output | Cek data *training* Info data |
| | Logika Proses | 1. Ambil data yang ingin diproses dari tabel mahasiswa<br>2. Tampilkan report data ke sekretariat |
| 2 | No.Proses | 2.2 |
| | Nama Proses | Pilih Angkatan |
| | Sumber | Mahasiswa, Sekretariat |
| | Input | Pilih angkatan, *Return* data |
| | Output | Cek Data Angkatan, info angkatan |
| | Logika Proses | 1. Pilih angkatan yang ingin diproses dari tabel mahasiswa<br>2. Tampilkan report angkatan ke sekretariat |
| 3 | No.Proses | 2.3 |
| | Nama Proses | Pilih Mata Kuliah |
| | Sumber | Sekretariat, Mahasiswa |
| | Input | Pilih mata kuliah, *Return* mata kuliah |
| | Output | Cek data mata kuliah, Info data mata kuliah |
| | Logika Proses | *1.* Ubah data *training*<br>2. Simpan di tabel mahasiswa |
| 4 | No.Proses | 2.4 |
| | Nama Proses | Bersihkan Data Proses |
| | Sumber | Sekretariat |
| | Input | Bersihkan Data |
| | Output | Data berhasil dibersihkan |
| | Logika Proses | 1. Bersihkan data proses yang sudah ditampilkan, untuk memproses data baru |
| 5 | No.Proses | 2.5 |
| | Nama Proses | Masukkan *Threshold* |
| | Sumber | Sekretariat |
| | Input | *Input threshold* |
| | Output | Info *threshold* |
| | Logika Proses | *1.* Masukkan nilai threshold yang akan diproses |

| | Proses | Keterangan |
|---|---|---|
| 6 | No.Proses | 2.6 |
| | Nama Proses | Proses *Database* |
| | Sumber | Sekretariat, Mahasiswa |
| | Input | Proses *database*, *Return database* |
| | Output | *Chek Database*, Info Proses |
| | Logika Proses | *1.* `Tekan button proses`<br>2. `Tampilkan Report data yang sudah diproses` |

## 3.9 Perancangan Sistem

Perancangan merupakan bagian dari metodologi pembangunan suatu perangkat lunak yang harus dilakukan setelah tahapan analisis. Pada bagian ini akan dijelaskan perancangan sistem yang akan dibangun dan akan dijelaskan pada bab IV.

## 3.10 Perancangan Data

Perancangan data atau lebih dikenal dengan perancangan basis data yaitu menciptakan atau merancang data yang terhubung dan disimpan secara bersama-sama. Untuk menggambarkannya digunakanlah struktur file.

### 3.10.1 Struktur Tabel

Pada proses *data mining* tingkat kelulusan mahasiswa ini, dibutuhkan *database* untuk menyimpan data hasil proses. Adapun struktur tabel digunakan untuk menggambar tabel-tabel beserta *field-field* yang digunakan secara lebih detail, dimana tipe data dan ukuran suatu *field* serta kunci-kunci (*primary key* atau *foreign key*) disajikan.

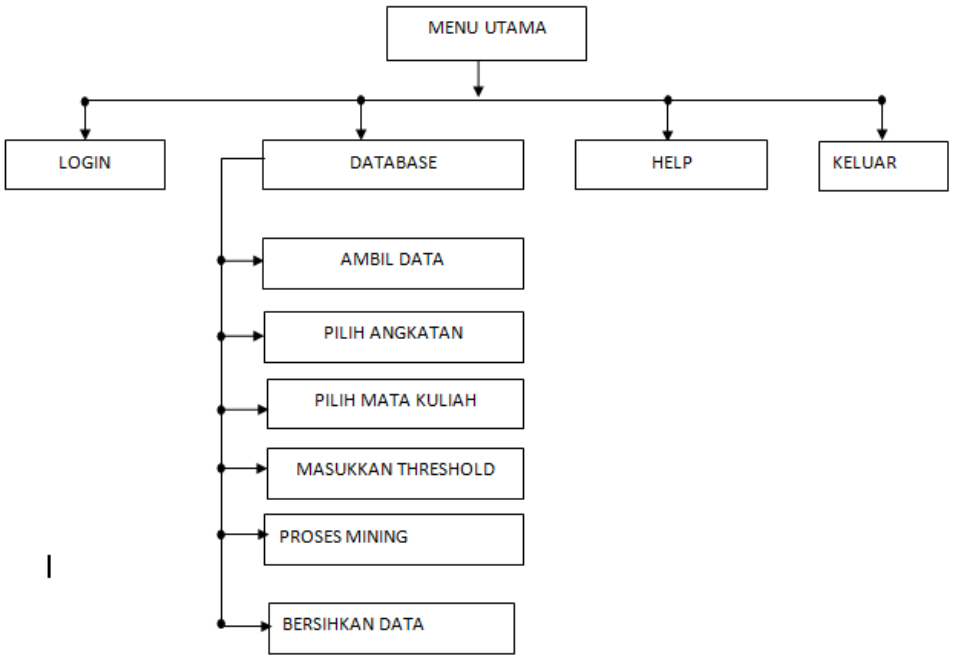
Tabel di bawah ini merupakan tabel beserta *field* dan *data type* yang digunakan dalam *database* dengan menggunakan *Microsoft Access* 2007:

Tabel 3.19 Data_Mahasiswa

| Nama *Field* | Tipe | Panjang |
|---|---|---|
| Nim | *Number* | 8 |
| Angkatan | *Number* | 4 |
| IPK | *Number* | 4 |
| Lama_Studi | *Number* | 2 |
| Algoritma | *Text* | 4 |
| Rakitan | *Text* | 4 |
| Mikro | *Text* | 4 |

### 3.10.2 Arsitektur Antarmuka

Berikut ini adalah perancangan struktur menu yang akan diterapkan pada program dan dirancang dengan *efisien* untuk mengurangi kompleksitas sistem yang akan ditunjukkan oleh Gambar 3.4 berikut.

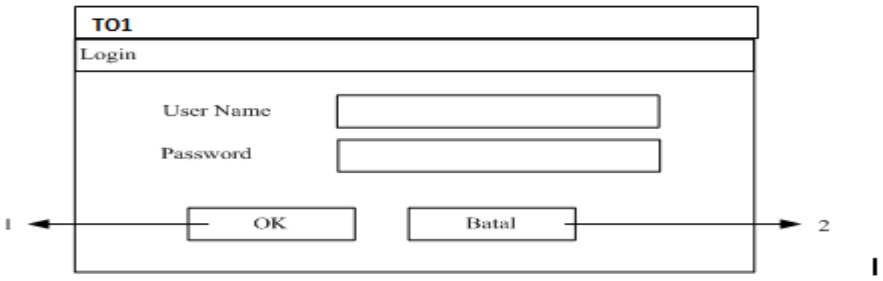



Gambar 3.4 Arsitektur Menu

## 3.11 Perancangan Antarmuka

Perancangan antarmuka merupakan suatu bentuk dari program yang akan dibuat untuk kebutuhan *interface* dengan pengguna. Spesifikasi antarmuka terdiri dari perancangan tampilan menu, tampilan form dan keluaran.

1. *Form* Login

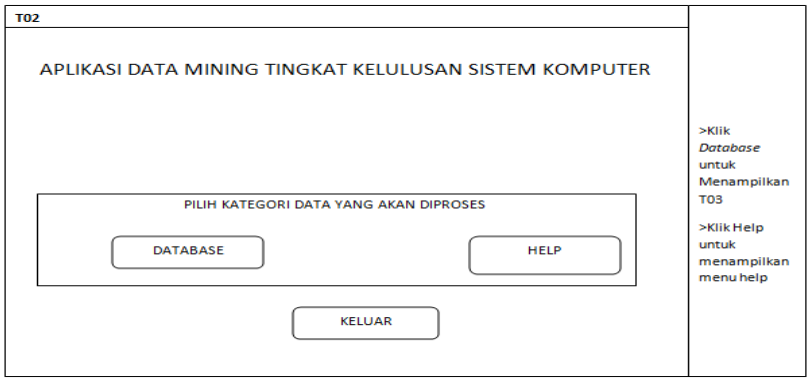



Gambar 3.5 Perancangan Form Login

Tabel 3.20 Penjelasan Tentang Perancangan Form Login

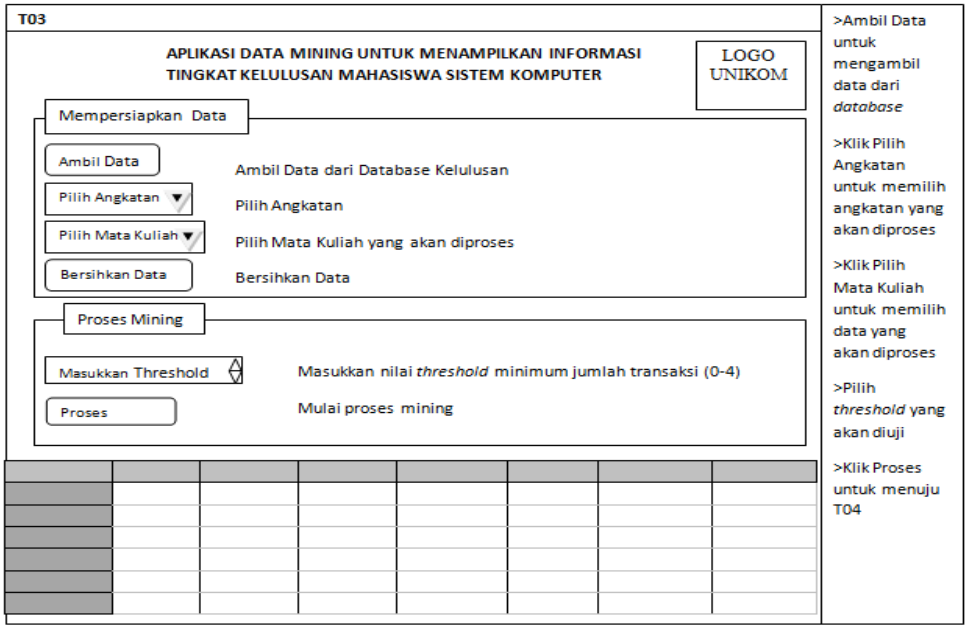
| No | Keterangan |
|---|---|
| **1** | Jika berhasil, maka akan masuk ke Form Utama dan jika gagal maka akan diminta login ulang. |
| **2** | Merupakan *button* untuk membatalkan login. |

2. *Form* Menu



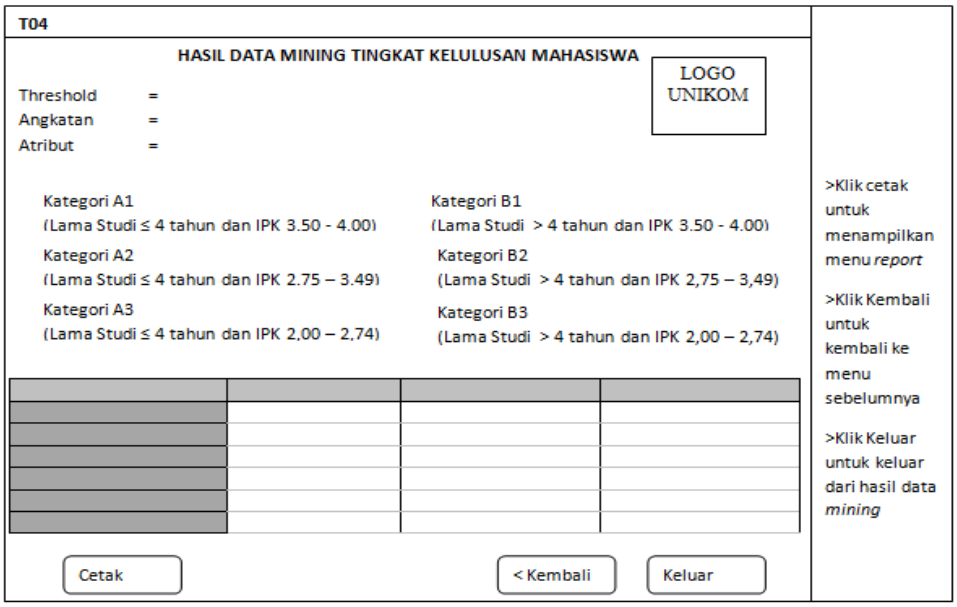

Gambar 3.6 Perancang *Form* Utama

3. *Form* Keseluruhan




Gambar 3.7 Perancangan Form Keseluruhan *Data Mining*

*4. Form Report*




Gambar 3.8 Perancangan *Form Report*

## 3.12 Perancangan Prosedural

Perancangan prosedural akan menjelaskan tentang bagaimana perangkat lunak berjalan. Perancangan prosedural digambarkan dalam bentuk *flowchart* sistem.

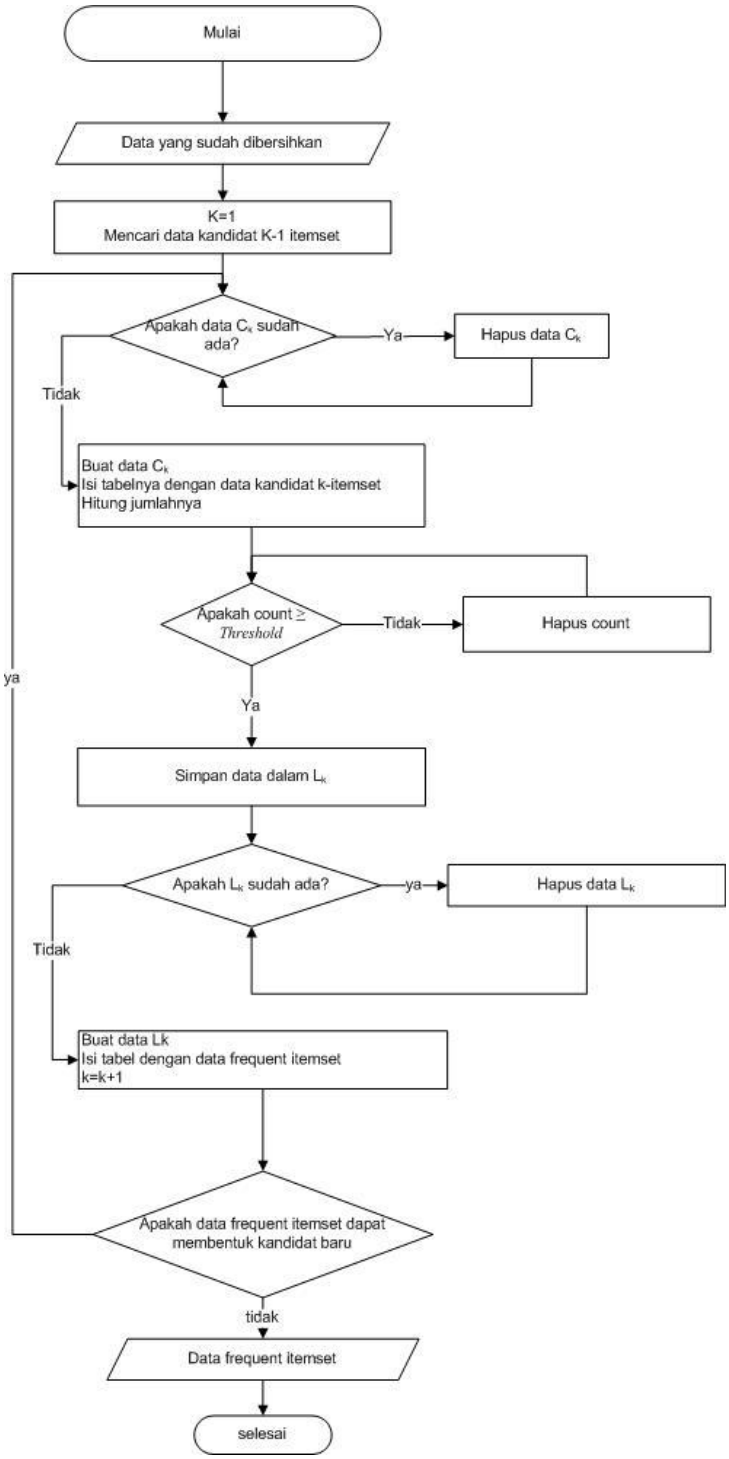
### 3.12.1 *Procedure Data Mining*

Berikut merupakan *flowchart* sistem Aplikasi Tingkat Kelulusan Mahasiswa yang akan dibangun



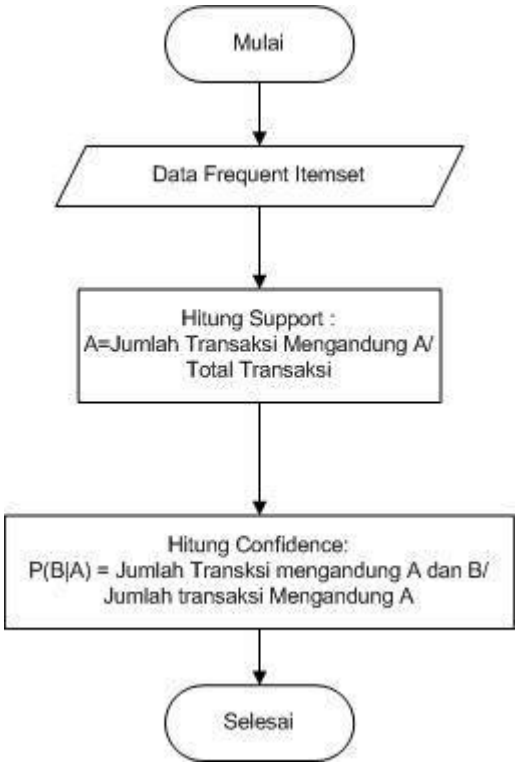

Gambar 3. 9 *Flowchart Procedure Data Mining*

### 3.12.2 *Procedure* Aturan Assosiatif Untuk Menentukan *Frequent Itemset*




Gambar 3.10 *Flowchart Procedure Frequent Itemset*

### 3.12.3 *Procedure* Aturan *Assosiatif* untuk Menentukkan Nilai *Support* dan *Confidence.*




Gambar 3.11 *Flowchart Procedure* Aturan *Assosiatif*