

BAB 2

LANDASAN TEORI

2.1 Data Mining

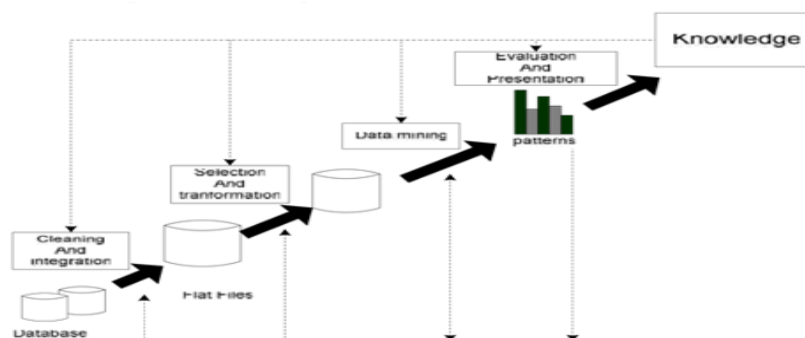
Data mining adalah proses pencarian pola-pola yang menarik dan tersembunyi (*hidden pattern*) dari suatu kumpulan data yang berukuran besar yang tersimpan didalam suatu basis data, data *warehouse*, atau tempat penyimpanan data lainnya. Kunci dari data mining meliputi data, informasi dan keputusan bisnis. Tujuan akhir dari data mining yaitu menggunakan informasi untuk membuat lebih efisien keputusan bisnis dan mengambil keputusan secara tepat (Hardiani, Sulisty, dan Hartanto, 2014). Data mining disebut juga sebagai *Knowledge Discovery In Database* (KDD). KDD adalah kegiatan yang meliputi pengumpulan, pemakaian data, historis untuk menemukan keteraturan, pola atau hubungan dalam *set* data berukuran besar (Santoso, Hariyadi, dan Prayitno, 2016).

Menurut Kusriani dan Luthfi (2009) hal-hal penting yang terkait dengan data mining adalah:

1. Data mining merupakan suatu proses otomatis terhadap data yang sudah ada.
2. Data yang akan diproses berupa data yang sangat besar.
3. Tujuan data mining adalah untuk mendapatkan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat.

2.1.1 Proses Data Mining

Dibawah ini merupakan tahap-tahap *Knowledge discovery in database* (KDD) pada Gambar 2.1.



Gambar 2.1. Tahap-tahap *knowledge didcovery in database* (KDD)

Berikut merupakan proses data mining menurut Han dan Kamber (2006):

1. Pembersihan data (*data cleaning*)

Merupakan proses menghilangkan *noise* dan data yang tidak konsisten atau data tidak relevan.

2. Integrasi data (*data integration*)

Merupakan penggabungan data dari berbagai *database* ke dalam satu *database* baru.

3. Seleksi data (*data selection*)

Data yang ada pada database sering kali tidak semuanya dipakai, oleh karena itu hanya data yang sesuai untuk dianalisis yang akan diambil dari *database*.

4. *Data Transformation*

Data diubah atau digabung ke dalam format yang sesuai untuk diproses dalam data mining.

5. Proses mining

Merupakan suatu proses utama saat metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data.

6. Evaluasi pola (*pattern evaluation*)

Untuk mengidentifikasi pola-pola menarik kedalam *knowledge based* yang ditemukan.

7. Presentasi pengetahuan

Merupakan visualisasi dan penyajian pengetahuan mengenai metode yang digunakan untuk memperoleh pengetahuan yang diperoleh pengguna. Dalam presentasi ini, visualisasi juga bisa membantu mengkomunikasikan hasil data mining.

2.1.2 Pengelompokan Data Mining

Menurut Han dan Kamber (2006) data mining dikelompokkan berdasarkan fungsinya, yaitu:

1. Deskripsi.

Deskripsi merupakan cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data yang dimiliki.

2. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variable target estimasi lebih ke arah numerik daripada ke arah kategori. Model yang dibangun menggunakan *record* lengkap yang menyediakan nilai variable target sebagai nilai prediksi

3. Prediksi

Prediksi menerka sebuah nilai yang belum diketahui dan juga memperkirakan nilai untuk masa mendatang.

4. Klasifikasi

Dalam klasifikasi terdapat target variable kategori, misal penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu tinggi, sedang, dan rendah.

5. *Klustering*

Merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan.

6. Asosiasi

Asosiasi bertugas menemukan atribut yang muncul dalam satu waktu. dalam dunia bisnis lebih umum disebut analisis keranjang belanja.

2.2 *Association Rule Mining (ARM)*

Menurut Kusrini dan Luthfi (2009) analisis asosiasi atau *association rule mining* adalah teknik data mining untuk menemukan aturan asosiasi antara kombinasi *item*. Analisis asosiasi atau *association rule mining* adalah teknik data mining untuk menemukan aturan suatu kombinasi *item*. Penting tidaknya suatu asosiasi dapat diketahui dengan dua tolok ukur, yaitu: *support* dan *confidence*. *Support* (nilai penunjang) adalah persentase kombinasi *item* tersebut dalam *database*, sedangkan *confidence* (nilai kepastian) adalah kuatnya hubungan antara *item* dalam aturan asosiasi.

2.2.1 Analisa Pola Frekuensi Tinggi (*Support*)

Support (nilai penunjang) adalah persentase kombinasi *item* tersebut dalam *database*. Mencari kombinasi *item* yang memenuhi syarat *minimum* dari nilai *support* dalam basis data. Suatu ukuran untuk menentukan apakah suatu *item* atau *item-set* layak untuk dicari *confidence*-nya (misalnya, dari keseluruhan transaksi yang ada, seberapa besar tingkat dominasi yang menunjukkan bahwa *item* A dan B dibeli bersamaan).

Nilai *support* sebuah item diperoleh dengan menggunakan Persamaan 2.1.

$$Support(A) = \frac{JumlahTransaksimengandungA}{TotalTransaksi} \quad (2.1)$$

Sedangkan untuk nilai *support* dari 2 item diperoleh dengan menggunakan Persamaan 2.2.

$$Support(A \cap B) = \frac{JumlahTransaksimengandungAdanB}{TotalTransaksi} \quad (2.2)$$

2.2.2 Pembentukan Aturan Asosiasi (*Confidence*)

Confidence (nilai kepastian) adalah kuatnya hubungan antara-item dalam aturan asosiasi. Suatu ukuran yang menunjukkan hubungan antar dua item secara *conditional* (misalnya, seberapa sering item B dibeli jika orang membeli item A). Perhitungan *confidence* menggunakan Persamaan 2.3.

$$ConfidenceP(A | B) = \frac{JumlahTransaksimengandungAdanB}{JumlahTransaksimengandungA} \quad (2.3)$$

untuk menentukan aturan asosiasi yang akan dipilih maka harus diurutkan berdasarkan *support* kali *confidence*. Aturan diambil sebanyak n aturan yang memiliki hasil terbesar.

2.2.3 Langkah-Langkah Proses Perhitungan *Association Rule*

Proses perhitungan *association rule* terdiri dari beberapa tahap adalah sebagai berikut (Tyas, 2008):

1. Sistem men-scan *database* untuk mendapat kandidat 1-*itemset* (himpunan *item* yang terdiri dari 1 *item*) dan menghitung nilai *support*. Kemudian nilai *support* tersebut dibandingkan dengan *minimum support* yang telah ditentukan, jika nilainya lebih besar atau sama dengan *minimum support* maka *itemset* tersebut termasuk dalam *large itemset*.
2. *Itemset* yang tidak termasuk dalam *large itemset* tidak diikuti dalam *iterasi* selanjutnya.
3. Pada *iterasi* kedua sistem akan menggunakan hasil *large itemset* pada *iterasi* pertama (L1) untuk membentuk kandidat *itemset* kedua (L2). Pada *iterasi* selanjutnya sistem akan menggunakan hasil *large itemset* pada *iterasi* selanjutnya akan menggunakan hasil *large itemset* pada *iterasi* sebelumnya (Lk-1) untuk membentuk kandidat *itemset* berikut (Lk). Sistem akan menggabungkan (*join*) Lk-1 dengan Lk-1 untuk mendapatkan Lk, seperti pada *iterasi* sebelumnya sistem akan menghapus (*prune*) kombinasi *itemset* yang tidak termasuk dalam *large itemset*.
4. Setelah dilakukan operasi *join*, maka pasangan *itemset* baru hasil proses *join* tersebut dihitung *support*-nya.
5. Proses pembentuk kandidat yang terdiri dari proses *join* dan *prune* akan terus dilakukan hingga himpunan kandidat *itemset*nya *null*, atau sudah tidak ada lagi kandidat yang akan dibentuk.
6. Setelah itu, dari hasil *frequent itemset* tersebut dibentuk *association rule* yang memenuhi nilai *support* dan *confidence* yang telah ditentukan.

7. Pada pembentukan *association rule*, nilai yang sama dianggap sebagai satu nilai.
8. *Association rule* yang terbentuk harus memenuhi nilai *minimum* yang telah ditentukan.
9. Untuk setiap *large itemset* L, kita cari himpunan bagian L yang tidak kosong untuk setiap himpunan bagian tersebut, dihasilkan *rule* dengan bentuk $aB(L-a)$ jika *support*-nya (L) dan *support*-nya (a) lebih besar dari *minimum support*.

2.2.4 Algoritma Apriori

Algoritma Apriori adalah salah satu algoritma yang melakukan pencarian *frequent itemset* dengan menggunakan teknik *association rule* (Erwin, 2009 dalam Santoso 2016). Algoritma apriori atau sering disebut juga dengan analisis asosiasi (*association rule mining*) adalah salah satu teknik data mining untuk mencari pola dalam kelompok data yang besar (Yotenka, 2017).

Algoritma Apriori dibagi menjadi beberapa tahap yang disebut narasi atau *pass* (Setiawati, 2012):

1. Pembentukan kandidat *itemset*, kandidat *k-itemset* dibentuk dari kombinasi ($k-1$) *itemset* yang didapat dari iterasi sebelumnya. Satu cara dari algoritma Apriori adalah adanya pemangkasan kandidat *k-itemset* yang subset-nya yang berisi $k-1$ *item* tidak termasuk dalam pola *frekuensi* tinggi dengan panjang $k-1$.
2. Penghitungan *support* dari tiap kandidat *k-itemset*. *Support* dari tiap kandidat *k-itemset* didapat dengan *scan database* untuk menghitung jumlah transaksi yang memuat semua *item* didalam kandidat *k-itemset* tersebut. Ini adalah juga ciri dari algoritma Apriori dimana diperlukan penghitungan dengan cara seluruh *database* sebanyak *k-itemset* terpanjang.
3. Tetapkan pola *frekuensi* tinggi. Pola *frekuensi* tinggi yang memuat k *item* atau *k-itemset* ditetapkan dari kandidat *k-itemset* yang *support*-nya lebih besar dari *minimum support*.
4. Bila tidak didapat pola *frekuensi* tinggi baru maka seluruh proses dihentikan. Bila tidak, maka k ditambah satu dan kembali bagian 1.

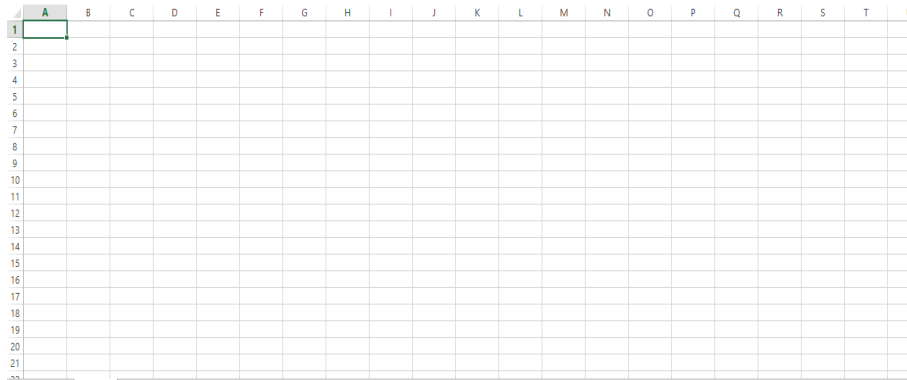
Ada dua proses utama yang dilakukan Algoritma Apriori, yaitu:

1. *Join* (penggabungan): untuk menemukan L_k , C_k dibangkitkan dengan melakukan proses *join* L_{k-1} dengan dirinya sendiri, $C_k=L_{k-1} \times L_{k-1}$, lalu anggota C_k diambil hanya yang terdapat didalam L_{k-1} .
2. *Prune* (pemangkasan): menghilangkan anggota C_k yang memiliki *support count* lebih kecil dari *minimum support* agar tidak dimasukkan ke dalam L_k .

2.2.5 Pembuatan Tabel *Boolean*

Cara membuat tabel *boolean* dapat dilihat sebagai berikut:

1. Buka *Ms.Excel* pada laptop/PC, dapat dilihat pada Gambar 2.2.



Gambar 2.2. Buka *Ms.Excel*

2. Buka data yang akan di ubah menjadi *boolean* yang telah berupa *Ms.Excel*, dapat dilihat pada Gambar 2.3.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Nomor	Transaksi	Item									
2	1	Surat 1 Ayat 1	Tauhid,									
3	2	Surat 1 Ayat 2	Ibadah, Tauhid,									
4	3	Surat 1 Ayat 3	Ibadah									
5	4	Surat 1 Ayat 4	Tidak Ada									
6	5	Surat 1 Ayat 5	Tidak Ada									
7	6	Surat 1 Ayat 6	Tidak Ada									
8	7	Surat 1 Ayat 7	Tidak Ada									
9	8	Surat 2 Ayat 1	Al-Qur'an									
10	9	Surat 2 Ayat 2	Al-Qur'an, Albidah, Agama,									
11	10	Surat 2 Ayat 3	Al-Qur'an, Agama,									
12	11	Surat 2 Ayat 4	Al-Qur'an, Albidah, Agama,									
13	12	Surat 2 Ayat 5	Al-Qur'an, Agama, Metafisika (Ketuhanan),									
14	13	Surat 2 Ayat 6	Agama, Metafisika (Ketuhanan),									
15	14	Surat 2 Ayat 7	Agama, Metafisika (Ketuhanan),									
16	15	Surat 2 Ayat 8	Agama,									
17	16	Surat 2 Ayat 9	Agama,									
18	17	Surat 2 Ayat 10	Agama,									
19	18	Surat 2 Ayat 11	Agama,									
20	19	Surat 2 Ayat 12	Agama,									
21	20	Surat 2 Ayat 13	Agama,									

Gambar 2.3. Buka data yang akan diubah menjadi *boolean*

3. Dari data yang telah berupa *Ms.Excel* tersebut, selanjutnya dibuat tabel *boolean*, dapat dilihat pada Gambar 2.4.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	
1	Transaksi	Al-Qur'an	Akidah	Khilaf dan Pendidikan Moral	Ibadat	Taubid	Ilmu dan Seni	Tabligh	Agama	UU Kewasayarakatan	Sejarah	Muhammad SAW	Bari Israil	Taurat	Na'rani	Metafisika (Ketuhanan)	Syari'at	Kesuksesan	Perdagangan												
2	Surat 1 Ayat 1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0												
3	Surat 1 Ayat 2	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0												
4	Surat 1 Ayat 3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
5	Surat 1 Ayat 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
6	Surat 1 Ayat 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
7	Surat 1 Ayat 6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
8	Surat 1 Ayat 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
9	Surat 2 Ayat 1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
10	Surat 2 Ayat 2	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0												
11	Surat 2 Ayat 3	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0												

Gambar 2.4. Data yang telah di ubah menjadi *boolean*

4. Penjelasan pada Gambar 2.4 diatas, apabila ayat al-qur'an membahas pokok bahasan akidah, maka di label dengan (1), sebaliknya apabila ayat al-qur'an tidak membahas pokok bahasan akidah maka di label dengan (0).

2.2.6 RapidMiner

Menurut (CTI dkk., 2017), *RapidMiner* merupakan *software*/perangkat lunak untuk pengolahan data. Dengan menggunakan prinsip dan algoritma data mining, *RapidMiner* mengekstrak pola-pola dari data *set* yang besar dengan mengkombinasikan metode statistika, kecerdasan buatan dan *database*. *RapidMiner* memudahkan penggunaanya dalam melakukan perhitungan data yang sangat banyak dengan menggunakan operator-operator. Operator ini berfungsi untuk memodifikasi data. Data dihubungkan dengan *node-node* pada operator kemudian kita hanya tinggal menghubungkannya ke node hasil untuk melihat hasilnya. Hasil yang diperlihatkan *RapidMiner* pun dapat ditampilkan secara visual dengan grafik. Menjadikan *RapidMiner* adalah salah satu *software* pilihan untuk melakukan ekstraksi data dengan metode-metode data mining.

Menurut Aprilla (2013), *RapidMiner* memiliki beberapa sifat sebagai berikut:

1. Ditulis dengan bahasa pemrograman *Java* sehingga dapat dijalankan diberbagai sistem operasi;
2. Proses penemuan pengetahuan dimodelkan sebagai *operator trees*;
3. Representasi *XML internal* untuk memastikan format standar pertukaran data;

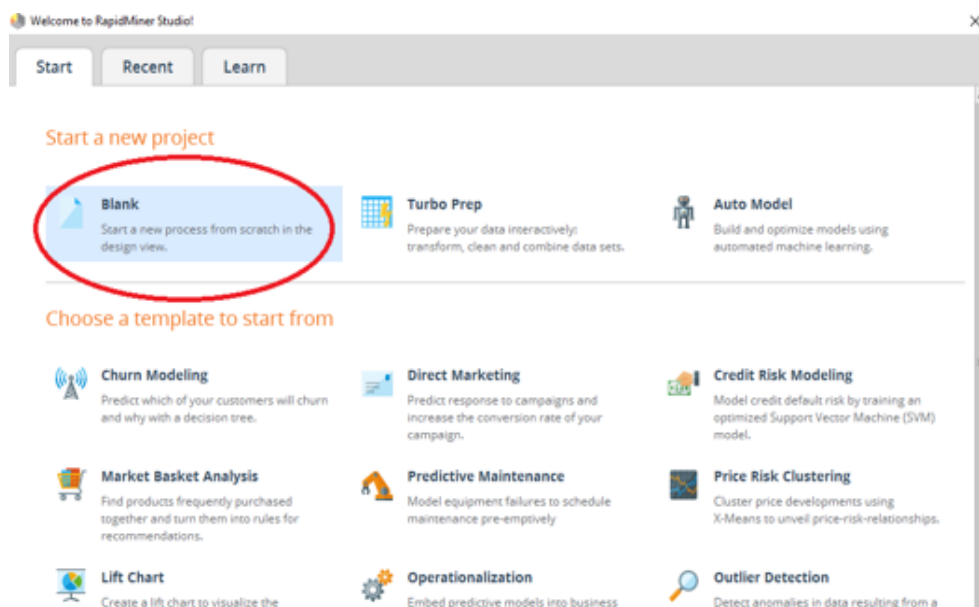
4. Bahasa *scripting* memungkinkan untuk eksperimen skala besar dan otomatisasi eksperimen;
5. Konsep *multi-layer* untuk menjamin tampilan data yang efisien dan menjamin penanganan data;
6. Memiliki *GUI*, *command line mode*, dan *Java API* yang dapat dipanggil dari program lain.

Menurut (Aprilla, 2013), beberapa fitur *RapidMiner* antara lain:

1. Banyaknya algoritma data mining, seperti *decision tree* dan *self-organization map*;
2. Bentuk grafis yang canggih, seperti tumpang tindih diagram *histogram*, *tree chart* dan *3D Scatter plots*;
3. Banyaknya variasi *plugin*, seperti *text plugin* untuk melakukan analisis teks;
4. Menyediakan prosedur data mining dan *machine learning* termasuk: *ETL(extraction, transformation, loading)*, data *preprocessing*, visualisasi, *modelling* dan evaluasi;
5. Proses data mining tersusun atas operator-operator yang *nestable*, dideskripsikan dengan *XML*, dan dibuat dengan *GUI*;
6. Mengintegrasikan proyek data mining *Weka* dan statistika R.

Implementasi Apriori pada *RapipMiner* adalah:

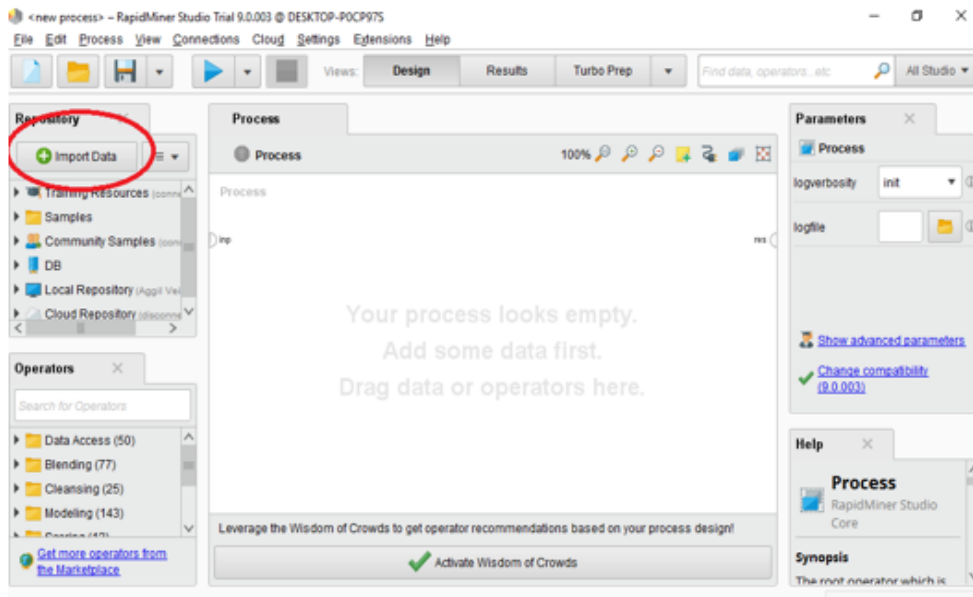
1. Buka tool *RapidMiner*, kemudian pilih *start* dan *blank* seperti pada Gambar 2.5.



Gambar 2.5. Tampilan *star* pada *RapidMiner*

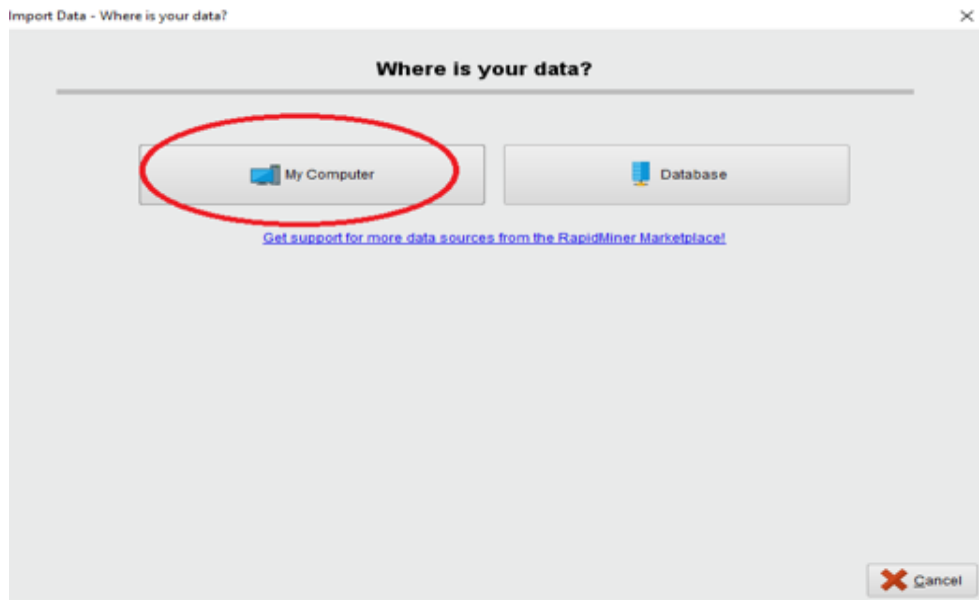
2. Maka tampilan *RapidMiner* akan terbuka dan muncul lembar kerja lalu pilih

menu *import* data agar dapat mengambil data yang akan diolah seperti pada Gambar 2.6.



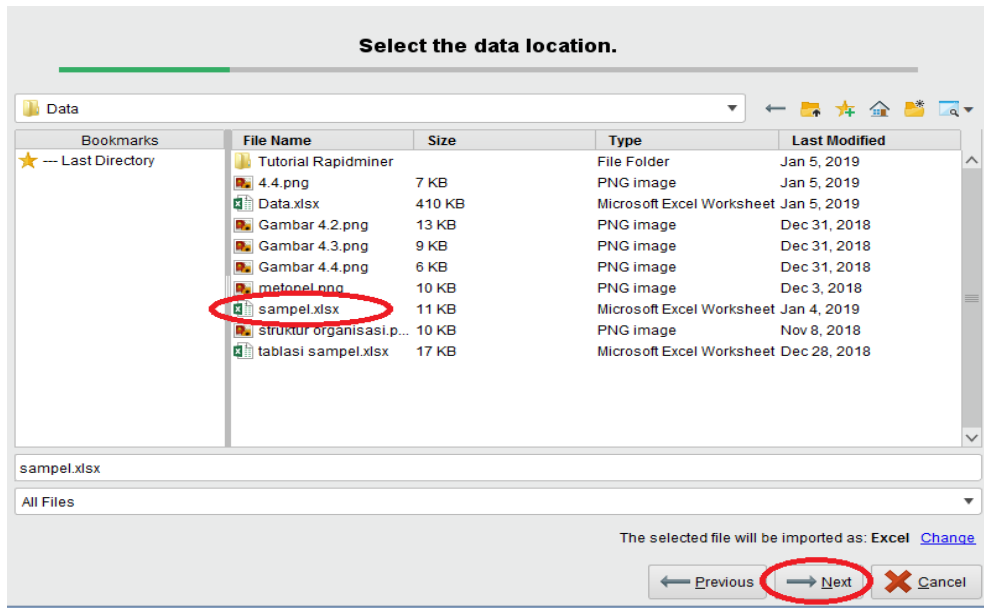
Gambar 2.6. Tampilan lembar kerja *RapidMiner*

3. Lalu setelah itu pilih *My Computer* agar dapat mencari *file* data yang telah tersimpan seperti pada Gambar 2.7.



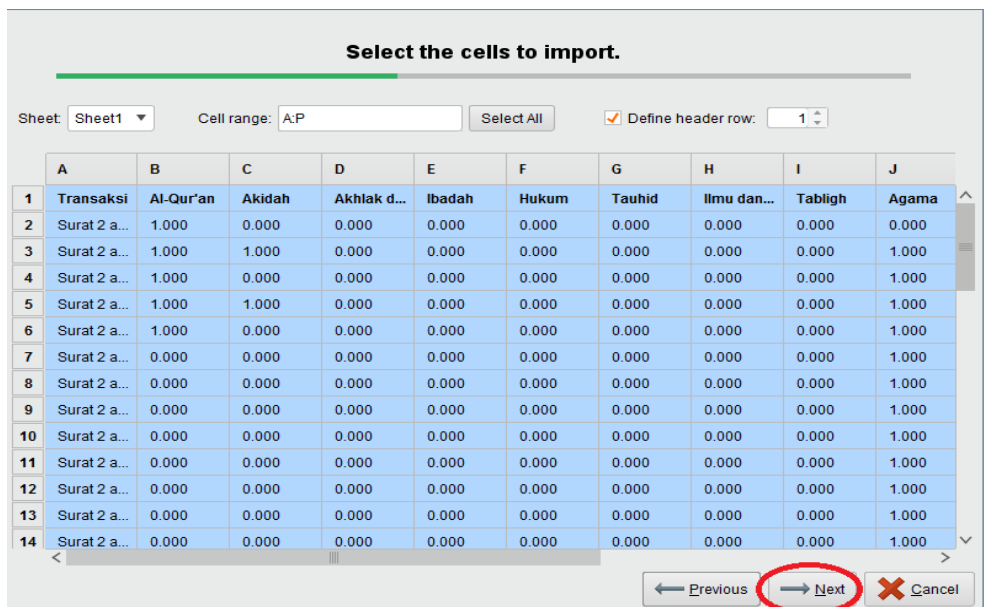
Gambar 2.7. Pilihan pencarian penyimpanan data

4. Setelah itu pilih *file* yang akan diolah lalu klik *next* seperti pada Gambar 2.8.



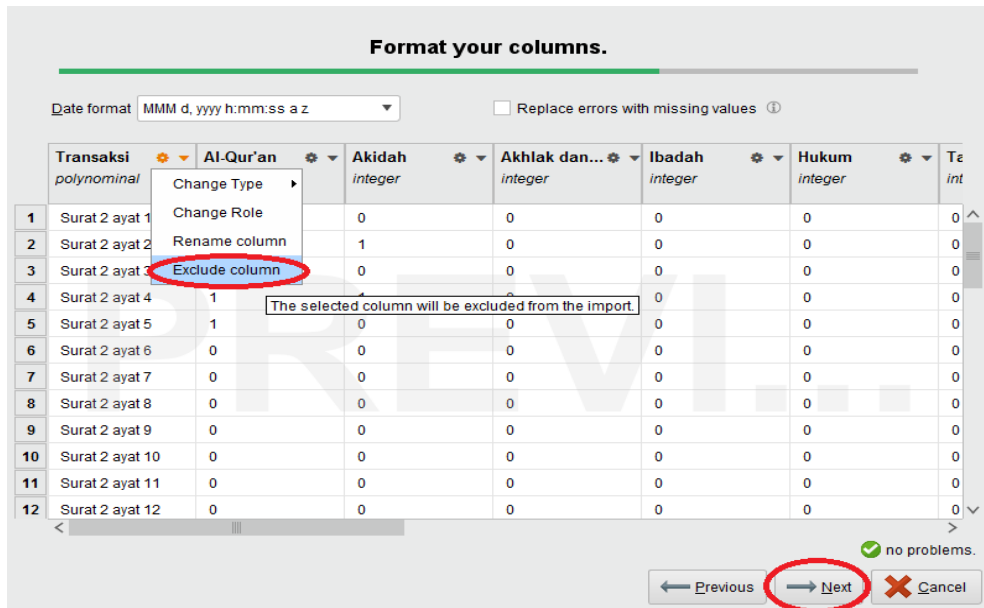
Gambar 2.8. Pemilihan *file* yang akan diolah

5. Lalu akan muncul tampilan data, kemudian klik *next* untuk melanjutkan proses seperti pada Gambar 2.9.



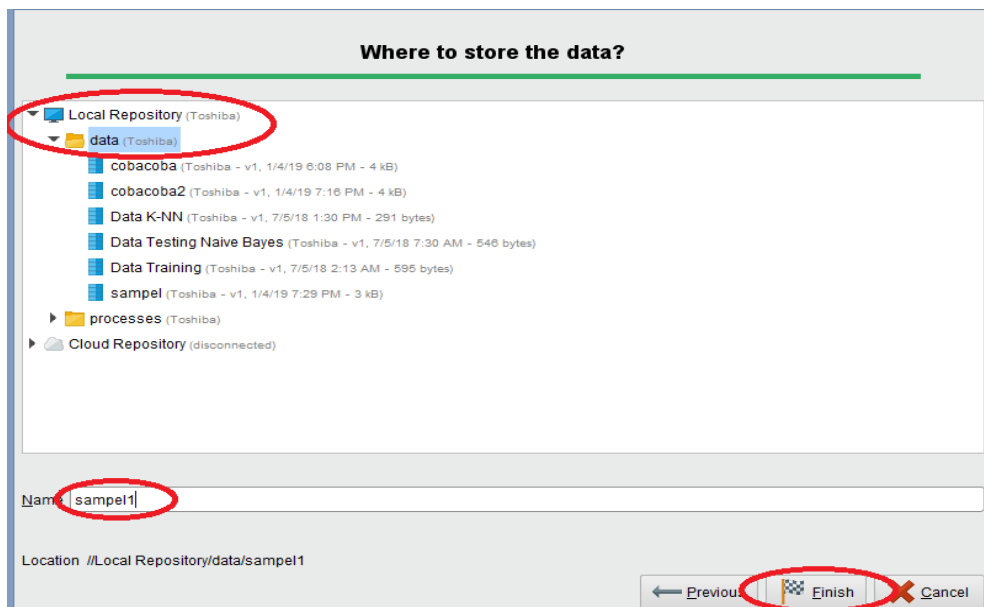
Gambar 2.9. Tampilan *export* data

6. Setelah itu akan muncul tampilan data kemudian pada kolom transaksi ubah tipe menjadi *Exclude* column lalu klik *next* untuk melanjutkan seperti pada Gambar 2.10.



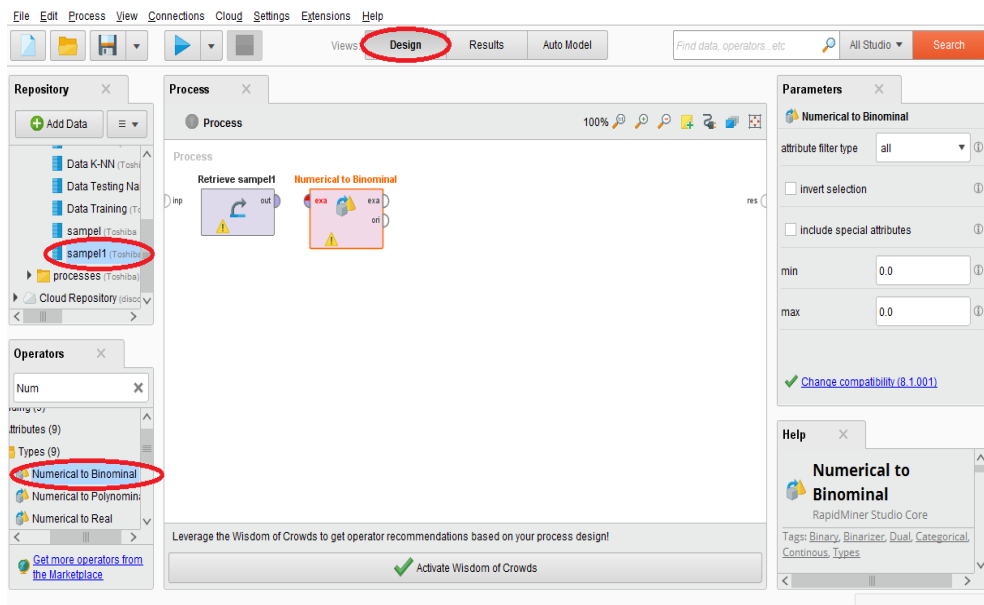
Gambar 2.10. Tampilan ubah tipe data

- Selanjutnya akan muncul tampilan penyimpanan data yang telah diimport, dengan memilih lokasi penyimpanan dan merubah nama data yang akan diolah lalu klik *next* untuk melanjutkan proses selanjutnya seperti pada Gambar 2.11.



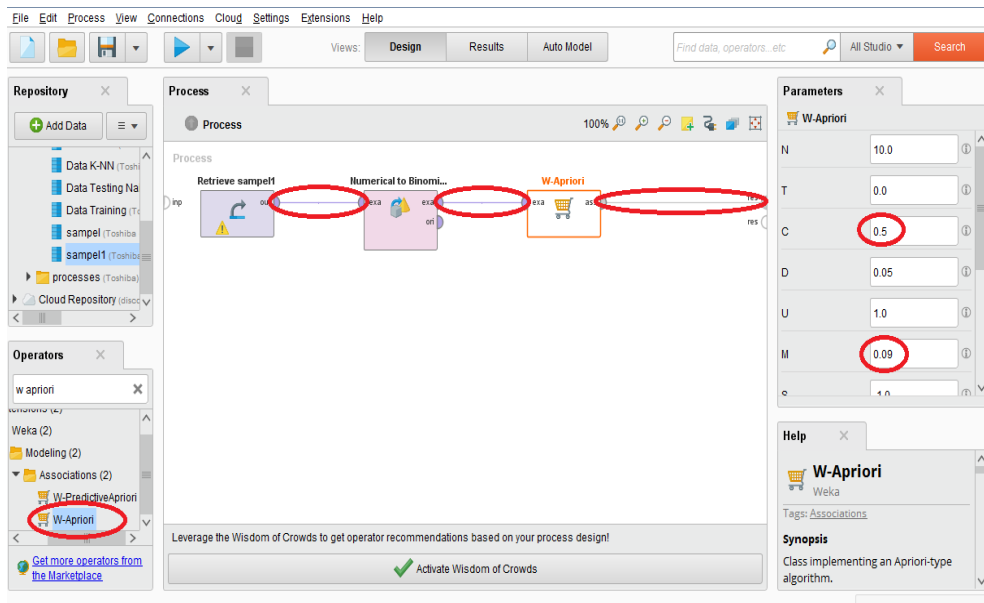
Gambar 2.11. Pemilihan dan perubahan nama data

- Lalu pilih menu *design, drag* data yang akan diolah pada halaman, lalu pilih *numerical to binominal* pada *operator*, lalu *drag* seperti pada Gambar 2.12.



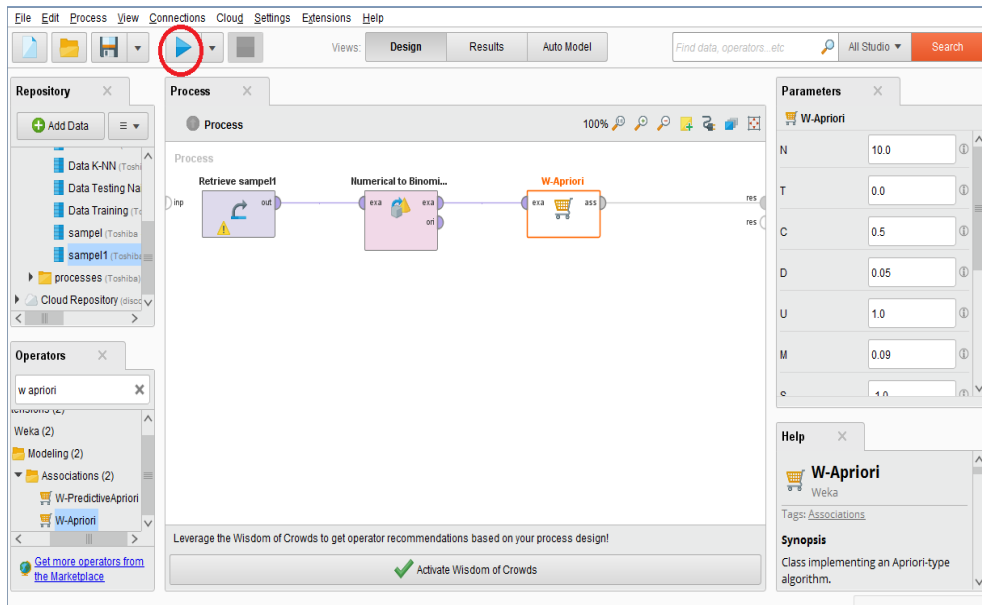
Gambar 2.12. Proses olah data

9. Lalu pilih *W- Apriori* pada *operator* dan hubungkan tiga atribut yang telah di *drag* dan atur *support confidence* seperti pada Gambar 2.13



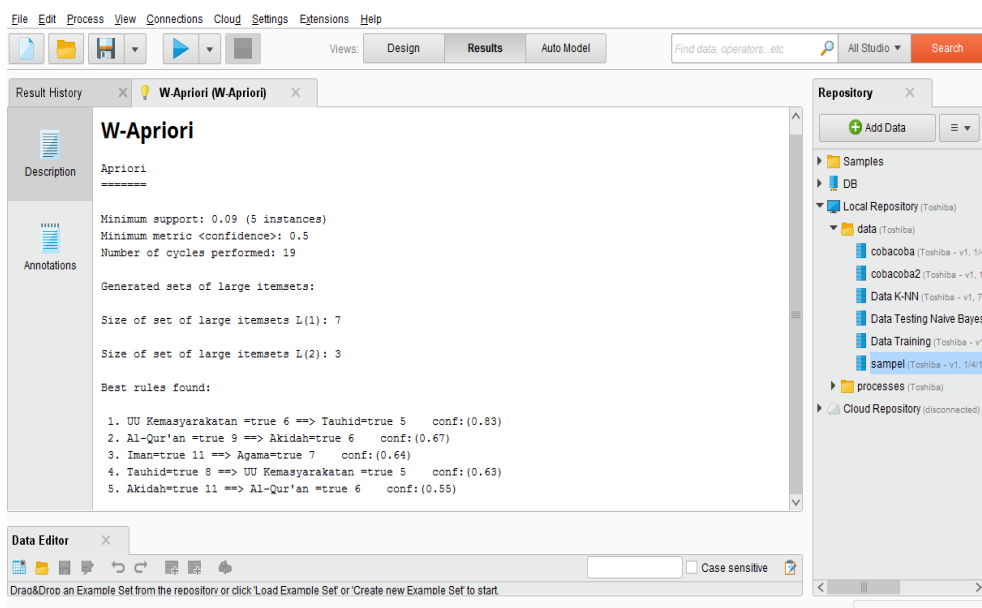
Gambar 2.13. Tampilan penghubung ketiga *operator*

10. Lalu jalankan proses dengan klik ikon *run* seperti pada Gambar 2.14.



Gambar 2.14. Tampilan *run* pada proses pengolahan data

11. Lalu akan muncul tampilan hasil dari data yang telah diolah seperti pada Gambar 2.15.



Gambar 2.15. Hasil perhitungan pada *RapidMiner*

2.3 Al-Quran

Menurut Akbar (2008) Al- Quran menurut bahasa (*etimologi*) adalah kata yang berbentuk *mashdar* (abstrak) dari kata kerja *qar'a* artinya dia telah membaca. Dari pengertian itu, maka Al-Quran berarti bacaan atau sesuatu yang dibaca dengan berulang-ulang oleh penganutnya. Kemudian kata *Qurun* itu sendiri adalah Kalam Allah SWT yang diturunkan kepada Nabi Muhammad SAW.

Menurut Akbar (2008) sedangkan pengertian Al-Quran menurut istilah (*terminologi*), menurut Imam Jalaluddin al-Suyuthy seorang ahli Tafsir dan Ilmu Tafsir di dalam bukunya “*Itmam al-Dirayah*” Al-Quran ialah firman Allah yang diturunkan kepada Nabi Muhammad SAW, dan berfungsi sebagai *mu’jizat* (melemahkan pihak-pihak yang menentangnya).

Menurut Akbar (2008) Allah SWT, menurunkan Al-Quran kepada Nabi Muhammad SAW, melalui perantara malaikat Jibril a.s. Al-Quran diturunkan secara berangsur-angsur selama 23 tahun sesuai dengan peristiwa-peristiwa dan kejadian-kejadian sejak Nabi Muhammad SAW diutus hingga wafat. Al-Quran diturunkan pada 17 Ramadhan tahun ke-40 dari usia Nabi Muhammad SAW, tiga belas tahun sebelum Hijrah, bertepatan pada tahun 610 M. Al-Quran diturunkan di Gua Hira’ yaitu sebuah gua di Jabal Nur yang terletak kira-kira tiga mil dari Kota Makkah. Al-Sya’bi menyebutkan bahwa Al-Qur’an mula-mula turun pertama kalinya pada malam qadar (lailatul qadar) dibulan Ramadhan. Pendapat ini didasarkan pada Firman Allah SWT, dalam Q.S Al-Qadr ayat 1:

إِنَّا أَنْزَلْنَاهُ فِي لَيْلَةِ الْقَدْرِ

“*Sesungguhnya Kami menurunkan Al-Quran pada suatu malam lailatul qadar*”

Firman Allah dalam Q.S Al-Dukhan ayat 3:

إِنَّا أَنْزَلْنَاهُ فِي لَيْلَةٍ مُبْرَكَةٍ إِنَّا كُنَّا مُنذِرِينَ

“*Sesungguhnya Kami menurunkannya pada malam yang diberkahi, dan sesungguhnya Kamilah yang memberi peringatan*”

Firman Allah dalam Q.S Al-Baqarah ayat 185:

شَهْرُ رَمَضَانَ الَّذِي أُنزِلَ فِيهِ الْقُرْآنُ هُدًى لِّلنَّاسِ وَبَيِّنَاتٍ مِّنَ الْهُدَىٰ وَالْفُرْقَانِ فَمَنْ شَهِدَ مِنْكُمُ الشَّهْرَ فَلْيَصُمْهُ وَمَنْ كَانَ مَرِيضًا أَوْ عَلَىٰ سَفَرٍ فَعِدَّةٌ مِّنْ أَيَّامٍ أُخَرَ يُرِيدُ اللَّهُ بِكُمُ الْيُسْرَ وَلَا يُرِيدُ بِكُمُ الْعُسْرَ وَلِتُكْمِلُوا الْعِدَّةَ وَلِتُكَبِّرُوا اللَّهَ عَلَىٰ مَا هَدَاكُمْ وَلَعَلَّكُمْ تَشْكُرُونَ

“*Bulan Ramadhan, bulan yang didalamnya diturunkan (permulaan) Al-Quran sebagai petunjuk bagi manusia dan penjelasan-penjelasan mengenai petunjuk itu dan pembeda (antara yang benar dan yang batil)*”.

Menurut Akbar (2008) ayat-ayat diatas menjelaskan tentang permulaan turunnya Al-Quran, yaitu pada malam *mubarakah* atau dinamai juga *lailatul qadar*, yakni salah satu malam pada Bulan Ramadhan. Malam tersebut dinamakan *lailah al-mubarakah* karena malam tersebut telah dipenuhi dengan berkah dan nikmat Allah yang tak ternilai, yaitu turunnya *Al-Quran al-Karim*, pembebasan umat manusia dari kesesatan, dan pembimbing mereka ke jalan yang benar, menuju kebahagiaan dunia dan akhirat. Dinamakan pula malam tersebut *lailatul qadr* karena ia mem-

punyai nilai yang tinggi lantaran pada malam itu diturunkan-Nya Kitab Suci kepada Nabi dan Rasul-Nya yang terakhir, dan akan menjadi pedoman bagi seluruh umat manusia sampai hari kiamat. Proses penurunan Al-Quran terbagi dua cara:

1. Penurunan Al-Quran secara sekaligus dari *Lauhul Mahfudz* ke *Baitul 'Izzah* dilangit dunia (langit lapisan pertama).
2. Penurunan Al-Quran secara berangsur-angsur, sedikit demi sedikit dan cara bertahap, sebagaimana menjelaskan bagian yang lain sesuai dengan fungsi dan kedudukannya, serta selaras pula dengan kepentingan-kepentingan yang dihadapi Rasulullah dan kaum muslim, yang diperkirakan dari permulaan sampai ayat yang terakhir turun.

Menurut Akbar (2008) Al-Quran terdiri dari 30 juz, 114 surat, yang dimulai dari surat Al-Fatihah dan ditutup dengan surat An-Nass. Sedangkan mengenai jumlah ayat dari seluruh Al-Quran tidak ada kesepakatan di kalangan para ulama, ada yang mengatakan tidak kurang dari 6000 ayat, sementara yang lain mengatakan lebih dari 6000 ayat. Namun pendapat yang dipandang terkuat adalah yang menyebutkan bahwa jumlah keseluruhannya 6236 ayat. Adapun mengenai perbedaan jumlah keseluruhan ayat-ayat Al-Quran tersebut, bukan karena adanya ayat-ayat Al-Quran yang diakui oleh sebagian dan tidak diakui oleh yang lainnya. Akan tetapi, perbedaan itu disebabkan, antara lain:

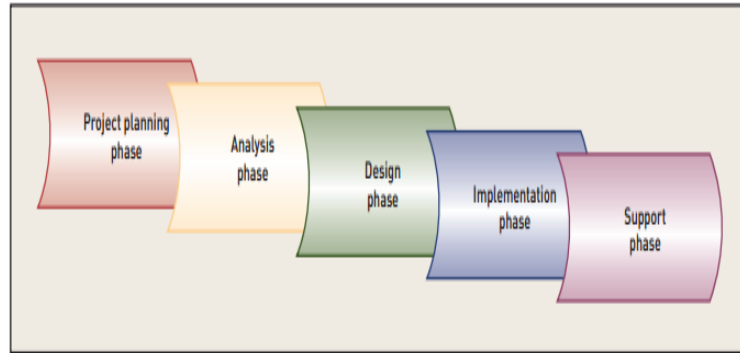
1. Perbedaan pendapat mereka tentang status "*Fatihatus Surat*" (pembuka surat) seperti *alif-lam-mim* dan sebagainya yang terdapat pada permulaan 29 buah surat dalam Al-Quran. Diantara ulama mengatakan bahwa *Fatihatus Surat* itu berkedudukan sebagai suatu ayat, sehingga ia dihitung sebagai ayat pertama dalam suratnya masing-masing, dan ada pula yang mengatakan tidak dihitung sebagai satu ayat.
2. Perbedaan pendapat tentang status *Bismillah* yang terdapat pada permulaan tiap-tiap surat kecuali surat al-Taubah. Dengan demikian, ada diantara mereka yang mengatakan bahwa *Bismillah* itu mempunyai kedudukan sebagai ayat pertama pada masing-masing surat, dan yang lain mengatakan tidak demikian halnya.
3. Perhitungan jumlah ayat-ayat Al-Quran itu dilaksanakan pada berbagai orang. Sebab itu tidak mengherankan jika muncul perbedaan dalam hasil perhitungan mereka itu.

Menurut Akbar (2008) para ulama telah mengemukakan keterangan mengenai keistimewaan Al-Quran, baik ditinjau dari bahasa, sifat, isi dan ajaran-ajaran maupun prinsip yang terkandung didalamnya. Sekalipun demikian, disini akan dikemukakan beberapa hal saja yang merupakan keistimewaan Al-Qur'an yang dapat dijangkau oleh pemikiran manusia yang terbatas, antara lain:

1. Al-Quran adalah Kitab Suci yang terakhir, yang diturunkan kepada Nabi dan Rasul-Nya.
2. Al-Quran adalah wahyu Allah, bukan karangan manusia atau ciptaan Nabi Muhammad atau pengikut-pengikutnya, baik yang ada pada masa Nabi Muhammad maupun yang ada pada masa sekarang.
3. Al-Quran itu setiap ayatnya disampaikan secara turun temurun sejak dari Nabi Muhammad sampai kepada kita sekarang ini dengan *mutawatir*, baik dalam bentuk hafalan ataupun tulisan sehingga ia terpelihara dari pemalsuan, penambahan dan pengurangan, karena telah dijamin oleh Allah SWT sepanjang zaman.
4. Al-Quran itu, bahkan setiap surat ataupun ayat adalah *mu'jizat* terbesar Nabi Muhammad SAW baik di tinjau dari segi lafadznya maupun isinya, sehingga tidak seorangpun yang sanggup menandinginya.
5. Membaca Al-Quran itu merupakan ibadah, bahkan diwajibkan membacanya dalam shalat.
6. Sebagai Kitab Suci untuk seluruh manusia, maka Al-Quran bersifat abadi (*absolut*) sepanjang zaman.

2.4 System Development Life Cycle (SDLC)

Menurut Satzinger dkk. (2011) metode *Systems Development Life Cycle* (SDLC/Siklus Hidup Pengembangan Sistem) atau *Systems Life Cycle* (Siklus Hidup Sistem), dalam rekayasa sistem dan rekayasa perangkat lunak, adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. Konsep ini umumnya merujuk pada sistem komputer atau informasi. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap: rencana (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*) dan pengelolaan (*support*). Berikut tahapan dalam SDLC dapat dilihat pada Gambar 2.16.



Gambar 2.16. Tahapan dalam SDLC
(Satzinger dkk., 2011)

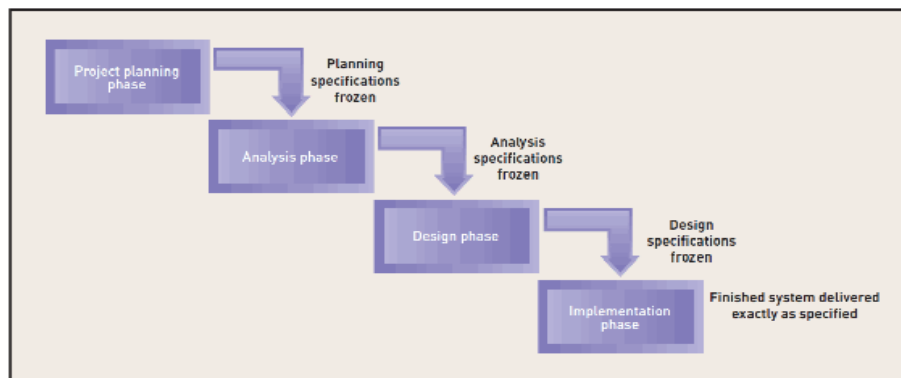
Menurut Satzinger dkk. (2011), ada beberapa tahapan dan tugas dalam SDLC, adapun tahapan dan tugas dalam SDLC di deskripsikan pada Tabel 2.1.

Tabel 2.1. Tahap dan tugas dalam SDLC

Fase	Tugas
<i>Project Planning</i>	Tahap ini digunakan untuk mengidentifikasi ruang lingkup sistem atau aplikasi baru, memastikan bahwa kegiatan pengembangan sistem atau aplikasi ini layak, dan mengembangkan jadwal, rencana sumber daya, dan anggaran dari kegiatan ini.
<i>Analysis</i>	Tahapan ini digunakan oleh seorang pengembang sistem informasi atau rekayasa perangkat lunak dalam memahami secara dalam dan merincikan semua kebutuhan-kebutuhan bisnis dan persyaratan-persyaratan yang berkaitan dengan pengolahan sistem atau rekayasa perangkat lunak baru agar lebih mudah dalam proses pengembangan sistem.
<i>Design</i>	Tahap ini digunakan untuk membuat perancangan pada sistem baru yang akan menghasilkan sebuah solusi berdasarkan persyaratan yang telah ditetapkan sebelumnya dan keputusan yang dibuat selama kegiatan analisis sistem baru.
<i>Implementation</i>	Tahap ini digunakan untuk membuat, membangun, melakukan <i>testing</i> atau pengujian, dan mengimplemenstasikan sistem informasi atau rekayasa perangkat lunak yang telah dibuat dimana setiap pengguna atau (<i>user</i>) akan dilakukan <i>training</i> atau pembelajaran siap untuk mendapatkan keuntungan seperti yang diharapkan dari penggunaan sistem baru.
<i>Support</i>	digunakan untuk memelihara sistem atau aplikasi yang telah dibuat agar mampu berjalan secara produktif, baik pada awalnya dan selama bertahun-tahun hidup sistem seperti yang diharapkan oleh pengembang sistem dan pemilik sistem.

2.5 Waterfal Model

Menurut Satzinger dkk. (2011) pada pengembangan penulis menggunakan metode Air terjun (*waterFall*). Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*). Gambar model SDLC *waterfall* dapat dilihat pada Gambar 2.17.



Gambar 2.17. Model SDLC air terjun (*waterfall*)
(Satzinger dkk., 2011)

Berdasarkan gambar Gambar 2.17 beberapa tahapan dalam SDLC model air terjun atau *waterfall* ini dijelaskan sebagai berikut:

1. Perencanaan

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Analisis

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Desain

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat

pada tahap desain.

4. Implementasi

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.6 Aplikasi

Aplikasi adalah suatu program siap pakai yang dapat digunakan untuk menjalankan perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut (Abdurahman dan Riswaya, 2014). Menurut Juansyah (2015) secara istilah pengertian aplikasi adalah satu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu tujuan yang diinginkan.

Menurut Nurcahyono (2017) aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu.