

BAB II

LANDASAN TEORI

2.1 Aturan Masa Studi Mahasiswa

Menurut Peraturan Menteri Pendidikan Dan Kebudayaan Republik Indonesia Tentang Standar Nasional Pendidikan Tinggi Bab 2 Bagian keempat Pasal 17 ayat 2 dan 3 menyebutkan bahwa ;

1. Untuk memenuhi capaian pembelajaran lulusan program sebagaimana dimaksud dalam Pasal 5, mahasiswa wajib menempuh beban belajar paling sedikit :
 - a. 36 sks untuk program diploma satu.
 - b. 72 sks untuk program diploma dua.
 - c. 108 sks untuk program diploma tiga.
 - d. 144 sks untuk program diploma empat dan program sarjana.
 - e. 36 sks untuk program profesi.
 - f. 72 sks untuk program magister, magister terapan, dan spesialis satu.
 - g. 72 sks untuk program doktor, doktor terapan, dan spesialis dua.
2. Masa studi terpakai bagi mahasiswa dengan beban belajar sebagaimana dimaksud pada ayat (2) sebagai berikut :
 - a. 1 (satu) sampai 2 (dua) tahun untuk program diploma satu.
 - b. 2 (dua) sampai 3 (tiga) tahun untuk program diploma dua.
 - c. 3 (tiga) sampai 4 (empat) tahun untuk program diploma tiga.
 - d. 4 (empat) sampai 5 (lima) tahun untuk program diploma empat dan program sarjana.

- e. 1 (satu) sampai 2 (dua) tahun untuk program profesi setelah menyelesaikan program sarjana atau diploma empat.
- f. 1,5 (satu koma lima) sampai 4 (empat) tahun untuk program magister, program magister terapan, dan program spesialis satu setelah menyelesaikan program sarjana atau diploma empat; dan paling sedikit 3 (tiga) tahun untuk program doktor, program doktor terapan, dan program spesialis dua.

2.2 Data Mining

Menurut Tan (2006:2) mendefinisikan data mining sebagai proses untuk mendapatkan informasi yang berguna dari gudang basis data yang besar. Data mining juga dapat diartikan sebagai pengekstrakan informasi baru yang diambil dari bongkahan data besar yang membantu dalam pengambilan keputusan. Istilah data mining kadang disebut juga *knowledge discovery* (Prasetyo, 2012). Pengolahan data mining terdiri dari beberapa metode pengolahan, yaitu (Kamagi, 2014) ;

2.2.1 *Predictive modelling* yang merupakan pengolahan data mining dengan melakukan prediksi/ peramalan. Tujuan metode ini untuk membangun model prediksi suatu nilai yang mempunyai ciri-ciri tertentu.

Contoh algoritmanya Linear Regression, Neural Network, Support Vector Machine, dan lain lain.

2.2.2 *Association* (Asosiasi) merupakan teknik dalam data mining yang mempelajari hubungan antar data. Contoh penggunaannya seperti untuk menganalisis perilaku mahasiswa yang datang terlambat. Contohnya jika mahasiswa memiliki jadwal dengan dosen A dan B, maka mahasiswa akan datang terlambat.

Contoh algoritmanya FP-Growth, A Priori, dan lain-lain.

2.2.3 *Clustering* (Klastering) atau pengelompokan merupakan teknik untuk mengelompokkan data ke dalam suatu kelompok tertentu. Contoh algoritmanya K-Means, K-Medoids, Self-Organisation Map (SOM), Fuzzy C-Means, dan lain-lain. Contoh untuk clustering: Terdapat lima pulau di Indonesia: Sumatera, Kalimantan, Jawa, Sulawesi dan Papua. Maka lima pulau tersebut dijadikan tiga klaster berdasarkan waktunya: Waktu Indonesia Barat (Sumatera, Kalimantan dan Jawa), Waktu Indonesia Tengah (Sulawesi) dan Waktu Indonesia Timur (Papua).

2.2.4 *Classification* merupakan teknik mengklasifikasikan data. Perbedaannya dengan metode clustering terletak pada data, dimana pada clustering variabel dependen tidak ada, sedangkan pada classification diharuskan ada variabel dependen.

Contoh algoritma yang menggunakan metode ini ID3 dan K Nearest Neighbors.

2.3 Metode K-Means

K-Means merupakan metode klasterisasi yang paling terkenal dan banyak digunakan di berbagai bidang karena sederhana, mudah diimplementasikan, memiliki kemampuan untuk mengklaster data yang

besar, mampu menangani data outlier, dan kompleksitas waktunya linier $O(nkT)$ dengan n adalah jumlah dokumen, k adalah jumlah kluster, dan T adalah jumlah iterasi. K-Means merupakan metode pengklusteran secara partitioning yang memisahkan data ke dalam kelompok yang berbeda. Dengan partitioning secara iteratif, K-Means mampu meminimalkan rata-rata jarak setiap data ke klasternya. Metode K-means dikembangkan oleh Mac Queen pada tahun 1967.

Sejalan dengan hal tersebut, disebutkan juga bahwa K-means merupakan suatu algoritma pengklusteran yang cukup sederhana yang mempartisi dataset kedalam beberapa kluster k . Algoritmanya cukup mudah untuk diimplementasikan dan dijalankan, relatif cepat, mudah disesuaikan dan banyak digunakan (Wu & Kumar, 2009). Prinsip utama dari teknik ini adalah menyusun k buah partisi/ pusat massa (*centroid*) / rata-rata (*mean*) dari sekumpulan data. Algoritma K-means dimulai dengan pembentukan partisi kluster di awal kemudian secara iteratif partisi kluster ini diperbaiki hingga tidak terjadi perubahan yang signifikan pada partisi kluster (Witten, Eibe, & Hall, 2011).

Karakteristik dari Algoritma K-means salah satunya adalah sangat sensitif dalam penentuan titik pusat awal kluster karena K-means membangkitkan titik pusat kluster awal random tersebut mendekati solusi akhir pusat kluster, K-means mempunyai kemungkinan yang tinggi untuk menemukan titik pusat yang tepat, sebaliknya jika awal titik pusat tersebut jauh dari solusi akhir pusat kluster, maka besar kemungkinan dapat menyebabkan hasil pengklusteran yang tidak tepat. Akibatnya k-means tidak

menjamin hasil pengklasteran yang unik. Hal inilah yang menyebabkan k-means sulit untuk mencapai optimum global, akan tetapi hanya optimum lokal. Selain itu, Algoritma K-means hanya bisa digunakan untuk data yang bernilai numerik.

Beberapa penelitian yang telah dilakukan mengenai masalah sensitifitas inisialisasi jumlah cluster (k), dan algoritma yang digunakan. Penelitian yang dilakukan oleh (Deelers & Auwatanamongkol, 2007) mengusulkan sebuah algoritma partisi data untuk menghitung awal pusat kluster. Partisi data mencoba membagi ruang data kedalam sel kecil atau kelompok, mana yang jarak interclusternya sebesar mungkin dan jarak intercluster sekecil mungkin. Sel partisi satu per satu sampai jumlah sel sama dengan jumlah klaser (k) yang telah ditetapkan, dan pusat-pusat sel k menjadi awal pusat kluster untuk K-means. Hasil percobaan menunjukkan bahwa algoritma partisi data bekerja lebih baik dibanding kan dengan inisialisasi pusat kluster secara acak dari sebagian kasus eksperimental dan dapat mengurangi waktu running algoritma K-means untuk dataset yang besar. (Yi et al, 2010), mengusulkan sebuah algoritma partisi data untuk memperbaiki awal pusat cluster yaitu Algoritma awal pusat cluster berbasis kepadatan (density). Algoritma ini menggunakan fungsi gaussian untuk memenuhi konsistensi global fitur clustering.

Algoritma yang diusulkan memilih titik kepadatan terbesar sebagai titik pusat awal pertama dari dataset, kemudian menentukan pusat awal kedua menggunakan metode yang sama dari dataset sehingga menghapus titik pertama dan tetangganya. Proses ini berlanjut sampai set M awal berisi k poin.

Hasil percobaan menunjukkan bahwa algoritma yang diusulkan sangat meningkatkan kualitas dan stabilitas.

Algoritma K-means, (Zhang & Fang, 2013) melakukan penelitian dalam perbaikan algoritma K-Means untuk mengoptimalkan inisialisasi pusat cluster. Dengan menentukan satu set data yang mencerminkan karakteristik distribusi data sebagai pusat awal cluster untuk mendukung pembagian data ke batas yang terbaik. Hasil percobaan didapatkan hasil akurasi algoritma perbaikan K-means meningkat secara signifikan dibandingkan dengan algoritma K-Means tradisional, dan algoritma yang diusulkan menunjukkan bahwa hasil setiap cluster lebih kompak.

2.4 Algoritma Cluster

Menurut Agusta (2007) *K-means clustering* merupakan salah satu metode data clustering non-hirarki yang mengelompokkan data dalam bentuk satu atau lebih cluster/kelompok. Data-data yang memiliki karakteristik yang sama dikelompokkan dalam satu *cluster*/kelompok dan data yang memiliki karakteristik yang berbeda dikelompokkan dengan *cluster*/kelompok yang lain sehingga data yang berada dalam satu *cluster*/kelompok memiliki tingkat variasi yang kecil (Ong, 2013).

Menurut Santosa (2007), langkah-langkah melakukan clustering dengan metode *K-Means* adalah sebagai berikut :

1. Pilih jumlah *cluster k*.
2. Inisialisasi *k* pusat *cluster* ini bisa dilakukan dengan berbagai cara. Namun yang paling sering dilakukan adalah dengan cara random. Pusat-pusat cluster diberi nilai awal dengan angka-angka random.

3. Alokasikan semua data/ objek ke cluster terdekat. Kedekatan dua objek ditentukan berdasarkan jarak kedua objek tersebut. Demikian juga kedekatan suatu data ke *cluster* tertentu ditentukan jarak antara data dengan pusat *cluster*. Dalam tahap ini perlu dihitung jarak tiap data ke tiap pusat *cluster*. Jarak paling antara satu data dengan satu *cluster* tertentu akan menentukan suatu data masuk dalam *cluster* mana.

Untuk menghitung jarak semua data ke setiap titik pusat cluster dapat menggunakan teori jarak Euclidean yang dirumuskan sebagai berikut :

$$D(i, j) = \sqrt{(X_{1i} - X_{1j})^2 + (X_{2i} - X_{2j})^2 + \dots + (X_{ki} - X_{kj})^2} \quad (1)$$

Penjelasan :

$D(i, j)$ = Jarak data ke i ke pusat cluster j

X_{ki} = Data ke i pada atribut data ke k

X_{kj} = Titik pusat ke j pada atribut ke k

4. Hitung kembali pusat *cluster* dengan keanggotaan *cluster* yang sekarang. Pusat *cluster* adalah rata-rata dari semua data/ objek dalam *cluster* tertentu. Jika dikehendaki bisa juga menggunakan median dari cluster tersebut. Jadi rata-rata (mean) bukan satu-satunya ukuran yang bisa dipakai.
5. Tugaskan lagi setiap objek memakai pusat *cluster* yang baru. Jika pusat cluster tidak berubah lagi maka proses *clustering* selesai. Atau, kembali ke langkah nomor 3 sampai pusat *cluster* tidak berubah lagi (Ong, 2013).

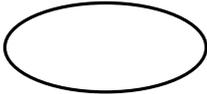
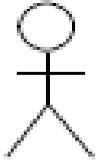
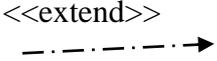
2.5 Unified Model Language(UML)

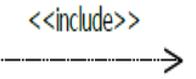
Unified Model Language(UML) merupakan sebuah bahasa untuk menguraikan spesifikasi yang sudah distandarisaikan untuk tujuan permodelan suatu objek (Salahuddin:2011). Berikut ini beberapa diagram yang digunakan dalam UML.

2.5.1 Use Case Diagram

Use Case Diagram merupakan permodelan untuk kelakuan (*Behaviour*) sistem informasi yangn akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Shalahuddin, 2011).

Tabel 2.1 Simbol *Use Case Diagram*

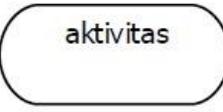
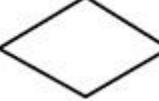
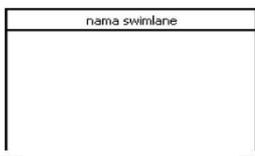
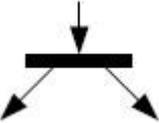
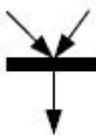
No	Nama	Simbol	Keterangan
1	Use Case		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2	Actor		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
3	Assosiation		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	Extend		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.

No	Nama	Simbol	Keterangan
5	<i>Generalation</i>		Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	<i>Include</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

2.5.2 Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Shalahuddin, 2011).

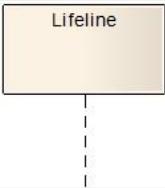
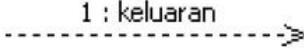
Tabel 2.2 Simbol *Activity Diagram*

No	Nama	Simbol	Keterangan
1	Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan/ <i>decision</i>		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
5	<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
6	<i>Fork</i>		<i>Fork</i> , digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel.
7	<i>Join</i>		<i>Join</i> , digunakan utk menunjukkan kegiatan yang digabungkan.

2.5.3 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup *objek* dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Shalahuddin, 2011).

Tabel 2.3 Simbol *Sequence Diagram*

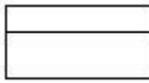
No	Nama	Simbol	Keterangan
1	<i>Actor</i>		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2	<i>Lifeline</i>		Merepresentasikan partisipasi dari objek dalam melakukan interaksi.
3	Waktu aktif		Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
4	Pesan tipe <i>call</i>		Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
5	Pesan tipe <i>message</i>		Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimkan.
6	Pesan tipe <i>return message</i>		Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

2.5.4 Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Shalahuddin, 2011).

- a. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
- b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Tabel 2.4 Simbol *Class Diagram*

No	Nama	Simbol	Keterangan
1	<i>Generalization</i>		Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang berada di atasnya objek induk (<i>ancestor</i>).
2	<i>Nary Association</i>		Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3	<i>Class</i>		Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4	<i>Collaboration</i>		Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5	<i>Realization</i>		Operasi yang benar-benar dilakukan oleh suatu objek.
6	<i>Dependency</i>		Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7	<i>Association</i>		Apa yang menghubungkan antara objek satu dengan objek lainnya.

2.6 Bahasa Pemrograman PHP

2.6.1. Pengertian PHP

PHP adalah bahasa pemrograman script yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. Contoh terkenal dari aplikasi PHP adalah forum (phpBB) dan MediaWiki (software di belakang Wikipedia). PHP juga dapat dilihat sebagai pilihan lain dari ASP.NET/C#/VB.NET Microsoft, ColdFusion Macromedia, JSP/Java Sun Microsystems, dan CGI/Perl. Contoh aplikasi lain yang lebih kompleks berupa CMS yang dibangun menggunakan PHP adalah Mambo, Joomla!, Postnuke, Xaraya, dan lain lain (Ratna, 2017).

PHP adalah (*PHP Hypertext Preprocessor*) adalah bahasa pemrograman *web* berupa *script* yang dapat diintegrasikan dengan HTML (Anhar, 2010).

2.6.2. Sejarah PHP

Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (Form Interpreted), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi open source, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP. Pada

November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan. Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP dirubah menjadi akronim berulang PHP: Hypertext Preprocessing. Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek (Ratna, 2017).

2.6.3. Kelebihan PHP dari bahasa pemrograman lain

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.

3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system. (Ratna, 2017).

2.7 MySQL

Database Management System (DBMS) adalah kumpulan program yang digunakan untuk mendefinisikan, mengatur, dan memproses database, sedangkan *database* adalah sebuah struktur yang dibangun untuk keperluan penyimpanan data. DBMS merupakan alat atau *tool* yang berperan untuk membangun struktur tersebut. Program DBMS saat ini banyak macamnya, seperti: MySQL, Oracle, Interbase/Firebird, IBM DB2, dan lain-lain.

Menurut Silberschatz (2006 : 2) *Database Management System* (DBMS) adalah suatu kumpulan data yang saling terkait dan kumpulan program untuk mengakses data tersebut. Kumpulan sebuah data biasanya disebut sebagai *database*, berisikan informasi yang relevan untuk disajikan. Tujuan utama DBMS adalah menyediakan sebuah cara untuk menyimpan dan menerima kembali informasi *database* secara mudah dan efisien.

MySQL merupakan *software* DBMS (atau *server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam

jumlah sangat besar, dapat diakses oleh banyak *user* (*multi-user*), data melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*).

SQL dibagi menjadi tiga bentuk *query*, yaitu :

1. DDL (*Data Definition Language*)

DDL adalah sebuah Metode *Query* SQL yang berguna untuk mendefinisikan data pada sebuah *database*. Berikut *query* yang termasuk DDL:

- a. *CREATE* digunakan untuk melakukan pembuatan tabel dan *database*.
- b. *DROP* digunakan untuk melakukan penghapusan tabel maupun *database*.
- c. *ALTER* digunakan untuk melakukan perubahan struktur tabel yang telah dibuat, baik menambah Field (*add*), mengganti nama Field (*change*) ataupun menamakannya kembali (*rename*), serta menghapus (*drop*).

2. DML (*Data Manipulation Language*)

DML adalah sebuah metode *query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *query* ini adalah untuk melakukan pemanipulasian *database* yang telah ada atau yang telah dibuat sebelumnya. Berikut *query* yang termasuk DML:

- a. *SELECT* digunakan untuk menampilkan data pada tabel.
- b. *INSERT* digunakan untuk melakukan penginputan / pemasukan data pada tabel *database*.

- c. *UPDATE* digunakan untuk melakukan perubahan atau peremajaan terhadap data yang ada pada tabel.
- d. *DELETE* digunakan untuk melakukan penghapusan data pada tabel.

3. DCL (*Data Control Language*)

DCL adalah sebuah metode *Query* yang dapat digunakan untuk mengendalikan eksekusi perintah. Biasanya berhubungan dengan pengaturan hak akses. Berikut *query* yang termasuk DCL:

- a. *GRANT* : Digunakan untuk memberikan hak akses (*privilage*) kepada user tertentu.
- b. *REVOKE* : Digunakan untuk mencabut hak akses dari *user* tertentu.

2.8 Kajian pustaka

Kajian pustaka berisi bahan referensi penulis berdasarkan penelitian-penelitian sebelumnya juga persamaan dan perbedaan antara penelitian sebelumnya dengan penelitian yang dilakukan oleh penulis untuk menghindari plagiat. Penelitian yang dilakukan oleh penulis berjudul Prediksi Masa Studi Mahasiswa Menggunakan Metode K-Means Clustering di Universitas Tunas Pembangunan (UTP) Surakarta. Variabel yang digunakan adalah Akumulasi jumlah SKS yang sudah ditempuh oleh mahasiswa selama 5 semester. Perhitungan dilakukan 2 kali yaitu, perhitungan pertama dilakukan setelah semester 3 dan perhitungan kedua dilakukan setelah semester 5.

Penelitian yang diambil sebagai bahan referensi seperti penerapan data mining dengan menggunakan metode clustering K-Means untuk mengukur tingkat ketepatan kelulusan mahasiswa program teknik informatika S1 Fakultas Ilmu Komputer Universitas Dian Nuswantoro Semarang oleh Rizal Akbar (2013), Implementasi Data Mining untuk Memprediksi Masa Studi Mahasiswa Menggunakan Algoritma C4.5 (Studi Kasus di Universitas Dehasen Bengkulu) oleh Siska Haryati, Aji Sudarsono, Eko Sunarya (2015), Implementasi Data Mining dengan Algoritma C4.5 untuk memprediksi tingkat Kelulusan Mahasiswa (Studi Kasus di Universitas Multimedia Nusantara) oleh David Hartanto Kamagi, Seng Hansum (2014), Data mining menggunakan algoritma k-means clustering untuk menentukan strategi promosi Universitas Dian Nuswantoro oleh Rima Dias Ramadhani (2013), Implementasi algoritma *k-means clustering* untuk menentukan strategi marketing president university oleh Johan Oscar Ong (2013), Perancangan aplikasi k-means sebagai penentu konsentrasi bagi mahasiswa informatika UMS oleh Dian Setiawan dan Yusuf Sulisty Nugroho, S.T,M.Eng (2015). Untuk lebih jelasnya bisa dilihat pada lampiran.