

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **1.1 Darmajaya**

Darmajaya merupakan salah satu Perguruan Tinggi Swasta (PTS) terkemuka di Provinsi Lampung, berdiri pada tahun 1995, di bawah naungan Yayasan Pendidikan Alfian Husin. Nama “DARMAJAYA” yang berarti “Dharma Bhakti dan Jasa Yoenidar Alfian”. Kampus Terpadu Darmajaya dibangun di atas lahan seluas 2,5 hektar yang terletak di Jalan Zainal Abidin Pagar Alam No. 93B. Labuhan Ratu, Bandar Lampung, Provinsi Lampung, Indonesia. Program pendidikan dan pelatihan keilmuan yang telah berlangsung selama ini adalah Sekolah Tinggi Ilmu Komputer dan Ekonomi Bisnis program studi, masing-masing program meliputi: (S1 Teknik Komputer, S1 Sistem Komputer, S1 Sistem Informasi, S1 Manajemen, S1 Akuntansi) Darmajaya memiliki jumlah mahasiswa baru rata-rata 700 orang .

#### **2.2 Prodi Teknik informatika**

Berdasarkan data di Prodi Teknik Informatika yang diterima untuk program studi Teknik informatika pada tahun 2017 sebanyak 201 mahasiswa menyelesaikan studi mereka dengan rata-rata lebih dari satu tahun 3,5 tahun ataupun 4 tahun tentu saja, sedang pada tahun 2018 sebanyak 209 mahasiswa dengan rata-rata 3,5 tahun percepatan ini dimungkinkan jika pemangku kepentingan dalam program penelitian ini dapat memprediksi kelulusan mahasiswa teknik informatika penyelesaian awal mahasiswa yang terburuburu untuk lulus dari universitas tepat waktu pasti akan mendapatkan nilai bagus dengan sertifikasi kurikulum.[1]

#### **2.3. Mahasiswa**

Mahasiswa yang terkait dengan prestasi akademik, seperti nilai-nilai mata kuliah, absensi, dan aktivitas di dalam kampus. Data ini mencakup informasi penting yang dapat dijadikan dasar untuk menganalisis faktor-faktor yang berkontribusi terhadap kelulusan mahasiswa. Pengelompokan mahasiswa berdasarkan karakteristik tertentu dapat membantu perguruan tinggi dalam menyusun program-program pembelajaran yang lebih sesuai dengan

kebutuhan setiap kelompok. Hal ini dapat meningkatkan efektivitas pembelajaran dan membantu mahasiswa dalam mencapai tingkat kelulusan yang lebih tinggi.

#### **2.4 Kelulusan Mahasiswa**

Mahasiswa sering disebut sebagai sekelompok orang dengan kemampuan intelektual yang lebih luas dibandingkan dengan kelompok usia mereka yang bukan mahasiswa atau kelompok usia lain yang diantara mereka. Dengan kecerdasannya, siswa dapat bertemu dan mencari .Terapkan masalah secara sistematis dalam kehidupan sehari-hari untuk dapat bersaing didunia kerja. tingkat kelulusan siswa merupakan faktor penting untuk dipertimbangkan karena persentase adalah angka nilai mempengaruhi evaluasi dewan dan status akreditasi program. [2]

#### **2.5 Prediksi**

Memperkirakan disusun secara sistematis untuk memprediksi sesuatu dapat terjadi berdasarkan tanggal masalah dan sekarang, maka perbedaannya sesuatu terjadi dan hasil yang diharapkan dapat diminimalkan. Prediksi tidak harus dibuat jawaban pasti untuk acara mendatang, tetapi cobalah untuk menemukan jawabannya sedekat mungkin itu akan terjadi.[3]

#### **2.6 K-Means**

*K-Means Clustering* adalah sebuah algoritma *clustering*, algoritma *K-Means* dimulai dengan pemilihan acak K, di sini K jumlah tumpukan yang akan terbentuk. Kemudian tetapkan nilai K sementara dan secara acak nilai ini menjadi pusat *cluster*, atau biasa disebut dengan *centroid*, *mean* atau “*mean*” untuk menghitung jarak dari setiap data yang ada terhadap masing-masing *centroid* menggunakan rumus *Euclidian* hingga ditemukan jarak yang paling dekat / minimum dari setiap data dengan *centroid*. Klasifikasikan setiap data berdasarkan kedekatannya dengan *centroid*. Lakukan langkah tersebut hingga nilai *centroid* tidak berubah. Adapun perhitungan manual dan cara kerja dari metode *K-Means* adalah dijelaskan sebagai berikut :

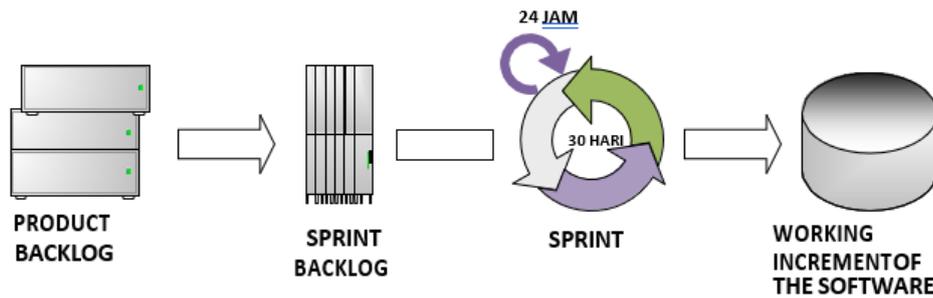
- a. Menentukan k sebagai jumlah *cluster* yang ingin di bentuk.
- b. Membangkitkan nilai random untuk pusat *cluster* awal (*centroid*)

- c. Menghitung jarak setiap data input terhadap masing – masing *centroid* menggunakan rumus jarak *Eucliden (Eucliden Distance)* hingga ditemukan jarak yang paling dekat dari setiap data dengan *centroid*. Dimana :  $x_i$  adalah data kriteria  $\mu_j$  adalah *centroid* pada kluster ke- $j$ .  
 $d(x_i, \mu_j) = \sqrt{(x_i - \mu_j)}$ .
- d. Mengklasifikasikan setiap data berdasarkan kedekatannya dengan *centroid* (jarak terkecil).
- e. Memperbaharui nilai *centroid*. Nilai *centroid* baru diperoleh dari rata-rata *cluster* yang bersangkutan dengan menggunakan rumus :
- $$\mu_i(t + 1) = \frac{\sum_j C_j X_j}{N_j}$$
- Dimana :  $\mu_j(t + 1)$  adalah *centroid* baru pada iterasi ke  $(t + 1)$   $N_j$  adalah banyak data pada cluster
- f. Melakukan perulangan dari langkah 2 hingga 5 , sampai anggota tiap cluster tidak ada yang berubah jika langkah f telah terpenuhi maka nilai pusat *cluster* pada iterasi terakhir akan digunakan sebagai parameter untuk menentukan klasifikasi data.[4]

## 2.7 Scrum

*Scrum* adalah salah satu metode rekayasa perangkat lunak dengan menggunakan prinsip-prinsip *agile* yang bertumpu pada kekuatan kolaborasi tim, *incremental product* dan proses iterasi untuk mewujudkan hasil akhir. *Scrum* merupakan sebuah kerangka kerja yang dapat mengatasi masalah kompleks adaptif, dimana pada saat bersamaan juga menghantarkan produk dengan nilai setinggi mungkin secara produktif dan kreatif[5]

Terdapat tiga buah tahapan dalam model Scrum yang berperan dalam proses pengembangan sistem, adapun penjelasan tahapan tersebut dapat dilihat pada gambar 2.1 berikut :



**Gambar 2.1 Scrum**

Tahap-tahapan metode *Scrum*

a. *Product Backlog*

Tahap *Product Backlog* ini lebih mengarah kepada teknik pengumpulan data berupa jadwal penelitian, hasil server dan wawancara tahapan ini merupakan tahapan awal yang bertujuan untuk mengumpulkan data-data mahasiswa yang diperlukan .

b. *Sprint Backlog*

Tahap *Sprint backlog* ini lebih megarah pada perancangan dan pemodelan agar pengembangan ini dapat sesuai dengan yang diharapkan *review*, hasilnya menjelaskan rancangan sistem seperti *dfd*, *erd*, desain *interface*, *pseuecode*

c. *Increment*

Tahap *Increment* merupakan hasil akhir dari tahap *Product Backlog* yang telah selesai dikembangkan pada saat tahapan *Sprint Log*. Pada tahap ini, diharapkan tahapan *Increment* telah selesai dilakukan dan sesuai dengan yang diinginkan sehingga mampu untuk digunakan sesuai dengan yang diinginkan.

## 2.8 Desain Sistem Data Base

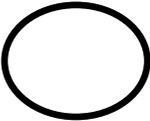
Sistem Manajemen Basis Data Relasional (RDBMS) adalah jenis sistem manajemen basis data yang didasarkan pada model data relasional. RDBMS digunakan untuk menyimpan, mengelola, dan mengakses data dalam bentuk tabel yang memiliki hubungan logis di antara mereka. Model relasional

menggunakan konsep tabel (atau entitas) yang terdiri dari baris dan kolom, dan setiap baris mewakili satu set data yang terkait.

### 2.8.1 *Relational Database Management System (RDMS)*

*Data Flow Diagram (DFD)* disebut juga dengan Diagram Arus Data (DAD). DFD adalah: suatu model logika data atau proses yang dibuat untuk menggambarkan: darimana asal data, dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut, dan interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut. DFD yang di dalam bahasa Indonesia disebut sebagai DAD (Diagram Arus Data) memperlihatkan gambaran tentang masukan- proses-keluaran dari suatu sistem/perangkat lunak, yaitu obyek-obyek data mengalir ke dalam perangkat lunak, kemudian ditransformasi oleh elemen-elemen pemrosesan, dan obyek-obyek data hasilnya akan mengalir keluar dari sistem/perangkat lunak.

**Tabel. 2.1 Simbol *Relational Data base Management System (RDMS)***

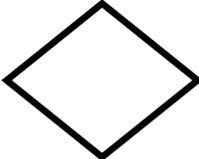
SIMBOL	KETERANGAN
<p><i>External Entity</i></p> 	<p>Merupakan sumber atau tujuan dari aliran data dari atau ke system</p>
<p>Arus Data (<i>Data Flow</i>)</p> 	<p>Menggambarkan aliran data</p>
<p>Proses</p> 	<p>Proses atau fungsi yang mentransformasikan data masukan menjadi keluaran.</p>

SIMBOL	KETERANGAN
Simpanan Data 	Komponen yang berfungsi untuk menyimpan data atau file.

### 2.8.2 Entity Relationship Diagram (ERD)

*Entity-Relationship Diagram* *Entity Relationship Diagram (ERD)* adalah diagram berbentuk notasi grafis yang berada dalam pembuatan *database* yang menghubungkan antara data satu dengan yang lain. Fungsi ERD adalah sebagai alat bantu dalam pembuatan database dan memberikan gambaran bagaimana kerja *database* yang akan dibuat.

Tabel 2.2 Simbol *Entity Relationship Diagram (ERD)*

NO	SIMBOL	KETERANGAN	DESKRIPSI
1		Simbol <i>Entity</i> / Objek data	Simbol ini adalah kumpulan objek atau suatu yang dapat dibedakan atau dapat diidentifikasi secara unik, kumpulan entitas yang sejenis
2		Simbol <i>relationship</i>	Simbol ini adalah hubungan yang terjadi antara satu entitas atau lebih, kumpulan relationship yang sejenis.
		Simbol garis	Simbol digunakan untuk menghubungkan antar tabel

## 2.9 MySQL

*MySQL (My Structured Query Language)* atau yang biasa dibaca mai-se-kuel adalah sebuah program pembuat dan pengelola *database* atau yang sering disebut dengan DBMS (*DataBase Management System*), sifat dari DBMS ini adalah *Open Source*. MySQL digunakan untuk data *warehousing* (gudang data), yaitu pengumpulan data terkait sistem cerdas untuk prediksi kelulusan mahasiswa.

## 2.10 PHP

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman open source yang sangat cocok atau dikhususkan untuk pengembangan web dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari. PHP adalah suatu bahasa pemrograman berbasis kode – kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke web *browser* menjadi kode HTML.

## 2.11 Website

*Website* atau situs dapat diartikan sebagai kumpulan dari halaman-halaman yang digunakan untuk menampilkan informasi berupa teks, gambar, animasi, suara, dan gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangun yang saling terkait dan biasanya dibuat untuk perorangan, organisasi, dan perusahaan.

## 2.12 UML (*Unified Modelling Language*)

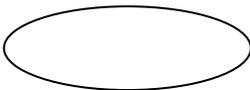
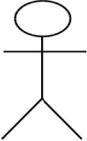
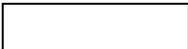
UML atau *Unified Modelling Language* adalah sebuah alat atau model yang digunakan untuk merancang pengembangan perangkat lunak yang berbasis pemrograman berorientasi objek. UML memberikan standar penulisan dalam membuat blueprint sistem yang mencakup proses bisnis, penulisan kelas-kelas dalam bahasa pemrograman yang spesifik, skema basis data, dan komponen yang diperlukan dalam perangkat lunak tersebut.

### 2.12.1 Use Case Diagram

*Use Case Diagram* adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor.

*Use case diagram* adalah gambaran grafis dari beberapa atau semua actor, use case, dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case diagram* tidak menjelaskan secara detail tentang penggunaan *use case*, tetapi hanya memberi gambaran singkat hubungan antara *use case*, aktor, dan sistem. Di dalam *usecase* ini akan diketahui fungsi-fungsi apa saja yang berada pada sistem yang dibuat.

**Tabel 2.3 Simbol-simbol *Use case Diagram***

No.	Simbol	Deskripsi
1	<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>usecase</i> .
3	<p><b>Aktor</b></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
4	<p><b>Assosiasi/association</b></p> 	Komunikasi antara aktor dan <i>Use case</i> yang berpartisipasi pada <i>Use case</i> atau <i>Use case</i>
5		Penjelasan tentang suatu paket yang

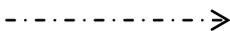
No.	Simbol	Deskripsi
		memperlihatkan sistem secara terbatas.

### 2.12.2 Activity Diagram

Diagram aktivitas adalah diagram perilaku penting lainnya dalam diagram UML untuk menggambarkan aspek dinamis dari sistem. Diagram aktivitas pada dasarnya adalah versi lanjutan dari diagram alir yang memodelkan aliran dari satu aktivitas ke aktivitas lainnya. Diagram Aktivitas menggambarkan bagaimana aktivitas dikoordinasikan untuk menyediakan layanan yang dapat berada pada tingkat abstraksi yang berbeda.

*Activity Diagram* (diagram aktivitas) menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi dan bagaimana mereka berakhir. Simbol-simbol yang digunakan dalam activity diagram dapat dilihat pada tabel 2.4.

**Tabel 2.4. Simbol Activity Diagram**

Simbol	Deskripsi
<p>StatusAwal/<i>InitialState</i></p> 	Status Awal atau <i>Initial State</i> adalah suatu keadaan awal pada saat system mulai hidup.
<p>Aktivasi</p> 	Aktivasi adalah suatu kegiatan yang dilakukan didalam sistem, biasanya diawali dengan kata kerja.
<p>StatusAkhir/<i>FinalState</i></p> 	Status Akhir atau <i>FinalState</i> adalah suatu keadaan akhir dari daur hidup.

Simbol	Deskripsi				
Percabangan/ <i>Decision</i> 	Percabangan adalah suatu kegiatan dimana terdapat pilihan kegiatan didalamnya.				
Penggabungan/Join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabung menjadi satu.				
Swimlane <table border="1" data-bbox="355 678 724 904"> <tr> <td>Actor 1</td> <td>Actor 2</td> </tr> <tr> <td></td> <td></td> </tr> </table>	Actor 1	Actor 2			Digunakan untuk memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Actor 1	Actor 2				

### 2.12.3 Use Case Diagram

*Use Case Diagram* adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor.

*Use case diagram* adalah gambaran grafis dari beberapa atau semua *actor*, *use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case diagram* tidak menjelaskan secara detail tentang penggunaan *use case*, tetapi hanya memberi gambaran singkat hubungan antara *use case*, aktor, dan sistem. Di dalam *use case* ini akan diketahui fungsi-fungsi apa saja yang berada pada sistem yang dibuat.

### 2.13 PenelitianTerkait

Dalam melakukan penelitian ini, peneliti terinspirasi dan mereferensi penelitian-penelitian terkait sesuai dengan masalah yang didapat dari latar belakang.

Tabel 2.5 Penelitian Terkait

No.	Judul	Penulis	Metode/Algoritma
1.	Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Algoritma Support Vector Machine	(RooyThaniket,Kusrini,Emha Taufik Luthf 2020)	Algoritma Support Vector Machine
2.	Pengembangan Sistem Cerdas Untuk Prediksi Daftar Kembali Mahasiswa Baru Dengan Metode Naive Bayes (Studi Kasus:Universitas Pendidikan Ganesha)	(Laila et al.,1978)	Metode Naive Bayes
3.	Case Based Reasoning Prediksi Waktu Studi Mahasiswa Menggunakan Metode Euclidean Distance Dan Normalisasi Min-Max	(kurniawan etal.,2021)	Metode Euclidean DistanceDan Normalisasi Min-Max
4	Prediksi Kelulusan Mahasiswa Berdasarkan Kinerja Akademik Menggunakan Pendekatan Data Mining Pada Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya	(Ryan Dwi Pambudi, Ahmad Afif Supianto, Nanang Yudi Setiawan 2019)	Pendekatan Data Data Mining

No.	Judul	Penulis	Metode/Algoritma
5	Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Metode Decision Tree Dan Artificial Neural Network	(Eko Prasetyo Rohmawan 2018)	Decision Tree Dan Artificial NeuralNetwork
6	Data mining untuk memprediksi kelulusan mahasiswa pendidikan matematika UIN Raden Intan Lampung menggunakan Naive Bayes	(Devi Heryana 2019)	Naive Bayes
7	Applikasi Prediksi Kelulusan Mahasiswa Berbasis K-Nearest Neighbord (K-N)	(Lalu Abd Hakim,Ahmad Ashril Rizal, Dwi Ratnasari 2019)	K-Nearest Neighbord (K-N)
8	Prediksi kelulusan mahasiswa dalam memilih Program Magister dengan Algoritma K-NN	(Aldo Sabothos Mananta, Green Arther Sandag 2018)	Algoritma K-NN
9	Optimal prediksi kelulusan mahasiswa tepat waktu menggunakan Naive bayes	(Hartatik 2020)	Naive bayes
10	Implementasi Desicion Tree untuk prediksi kelulusan mahasiswa	(Christin Nandari Dengen , Kusrini, Emha Tufiq Luthfi 2020)	Desicion Tree

#### 2.14. Algoritma *K-means*

*K-Means* adalah algoritma pengelompokan yang digunakan untuk mengelompokkan data ke dalam kategori yang disebut "kluster". Setiap kluster diwakili oleh pusatnya yang disebut "*centroid*". Algoritma ini mencoba untuk mengurangi variasi dalam kluster dengan menggerakkan *centroid* ke posisi yang mengurangi jarak antara data dalam kluster tersebut dengan *centroidnya*.

Langkah-langkah dalam algoritma *K-Means*:

1. Pilih jumlah kelompok yang ingin dibentuk (nilai "K") dan inisialisasi pusat-pusat awal untuk setiap kelompok secara acak atau dengan metode tertentu.
2. Setiap data dalam dataset diatribusikan ke kelompok terdekat berdasarkan jarak antara data dan pusat kelompok. Jarak ini biasanya diukur menggunakan metrik *Euclidean*, tetapi metrik lainnya juga bisa digunakan.
3. Jumlah kelompok atau klaster yang diinginkan harus ditentukan sebelumnya. Pada awalnya, titik-titik yang mewakili pusat dari masing-masing klaster secara acak dipilih dari data yang ada.
4. Setiap titik data kemudian diberi label berdasarkan jaraknya ke pusat-pusat klaster. Data akan dikelompokkan ke klaster yang memiliki pusat terdekat.
5. Setelah semua data dikelompokkan, pusat-pusat klaster yang baru dihitung berdasarkan rata-rata dari semua titik data yang termasuk dalam klaster tersebut.
6. Langkah 2 dan 3 diulang hingga tidak ada perubahan dalam pengelompokan atau hingga batas iterasi tertentu tercapai. Pada setiap iterasi, pusat klaster baru dihitung berdasarkan rata-rata data yang ada di dalam klaster.
7. Algoritma akan berhenti saat pusat-pusat klaster tidak berubah secara signifikan antara iterasi yang berurutan atau telah mencapai batas iterasi maksimum.

8. Algoritma akan berhenti saat pusat-pusat klaster tidak berubah secara signifikan antara iterasi yang berurutan atau telah mencapai batas iterasi maksimum.
9. Setelah konvergensi tercapai, data akan terbagi ke dalam kelompok-kelompok yang stabil. Setiap kelompok diwakili oleh pusat klaster, dan kita dapat mengidentifikasi anggota-anggota klaster tersebut