

**IMPLEMENTASI ALGORITMA *K-NEAREST NEIGHBOR* UNTUK
SISTEM REKOMENDASI TOPIK PENELITIAN PADA JURUSAN ILMU
KOMPUTER UNIVERSITAS LAMPUNG BERBASIS KLASIFIKASI**

(SKRIPSI)

Oleh:

**NONIK AISYAH
1717051011**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
2021**

ABSTRAK

IMPLEMENTASI ALGORITMA *K-NEAREST NEIGHBOR* UNTUK SISTEM REKOMENDASI TOPIK PENELITIAN PADA JURUSAN ILMU KOMPUTER UNIVERSITAS LAMPUNG BERBASIS KLASIFIKASI

Oleh

NONIK AISYAH

Topik penelitian adalah landasan dasar yang dimiliki mahasiswa dalam melakukan sebuah penelitian. Topik penelitian sangat penting bagi mahasiswa tingkat akhir dalam menyelesaikan skripsi salah satunya di Jurusan Ilmu Komputer, maka untuk mempermudah dalam menentukan topik penelitian dengan memanfaatkan sistem rekomendasi. Tujuan dari penelitian ini untuk merekomendasikan topik penelitian pada mahasiswa Jurusan Ilmu Komputer Universitas Lampung menggunakan algoritma *K-nearest neighbor*. Algoritma *K-nearest neighbor* adalah suatu model pendukung keputusan yang dapat mengklasifikasikan data berdasarkan jarak terdekat dengan memanfaatkan data mahasiswa Jurusan Ilmu Komputer yang telah lulus. Metode yang digunakan pada sistem yaitu *extreme programming* dengan tahapan meliputi : *planning, design, coding, dan testing*. Data yang digunakan yaitu data yang diperoleh dari pengelola server siakadu, UPT TIK Universitas Lampung pada tahun 2016 – 2019. Variabel independen yang digunakan yaitu 35 data matakuliah dan variabel dependen yaitu 14 kategori penelitian mahasiswa yang telah lulus. *Output* dari sistem ini yaitu hasil prediksi kategori topik penelitian. Hasil pengujian menunjukkan model terbaik dengan tetangga terdekat yaitu 3 dan pendekatan jarak terbaik yaitu *manhattan distance* menghasilkan *accuracy* sebesar 82%, *precision* sebesar 82% dan *recall* sebesar 82%. Pengujian sistem menggunakan *black-box testing* menunjukkan setiap kasus uji berhasil menghasilkan *output* yang sesuai sehingga dapat membantu mahasiswa untuk menentukan topik penelitian.

Kata Kunci: *K-nearest neighbor*, Klasifikasi, Sistem Rekomendasi, *Extreme Programming*

ABSTRACT

IMPLEMENTATION OF K-NEAREST NEIGHBOR ALGORITHM FOR CLASSIFICATION-BASED RESEARCH TOPIC RECOMMENDATION SYSTEM OF THE DEPARTEMENT OF COMPUTER SCIENCE AT THE UNIVERSITY OF LAMPUNG

By

NONIK AISYAH

The research topic is the basic foundation that students have in conducting a research. This research topic is very necessary by final year students in completing their thesis, one of which is in the Department of Computer Science . This makes it easier to determine research topics by utilizing a recommendation system. The purpose of this study is to recommend research topics to students of the Department of Computer Science, University of Lampung using the K-nearest neighbor algorithm. The K-nearest neighbor algorithm is a decision support model that can classify data based on the closest distance by utilizing data from students of the Computer Science Department who have graduated. The method used in this system is extreme programming with stages including: planning, designing, coding, and testing. The data used is data obtained from the ICT server manager of UPT Siakadu, University of Lampung in 2016 – 2019. The independent variables used are 35 course data and the dependent variable is 14 research categories of students who have graduated. The output of this system is prediction of research topic category. The test results will show that the best model with the nearest neighbor is 3 and the best distance approach is the Manhattan distance, resulting in 82% accuracy, 82% precision and 82% recall, and system testing using black box testing which shows that each test case successfully produces the right output so that it can help students in determining their research topic.

Keywords: K-nearest neighbor, Classification, Recommendation System, Extreme Programming

**IMPLEMENTASI ALGORITMA *K-NEAREST NEIGHBOR* UNTUK
SISTEM REKOMENDASI TOPIK PENELITIAN PADA JURUSAN ILMU
KOMPUTER UNIVERSITAS LAMPUNG BERBASIS KLASIFIKASI**

Oleh

NONIK AISYAH

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA ILMU KOMPUTER

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2021**

Judul Skripsi : **IMPLEMENTASI ALGORITMA K-NEAREST
NEIGHBOR UNTUK SISTEM
REKOMENDASI TOPIK PENELITIAN
PADA JURUSAN ILMU KOMPUTER
UNIVERSITAS LAMPUNG BERBASIS
KLASIFIKASI**

Nama Mahasiswa : **Nonik Aisyah**

Nomor Pokok Mahasiswa : 1717051011

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam



Dr. Ir. Kurnia Muludi, M.S.Sc.
NIP 19640616 198902 1 001

Ir. Olgih Forda Nama, S.T., M.T.I.
NIP 19830712 200812 1 003

2. Ketua Jurusan Ilmu Komputer

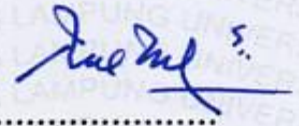
Didik Kurniawan, S.Si., M.T.
NIP 19800419 200501 1 004

MENGESAHKAN

1. Tim Penguji

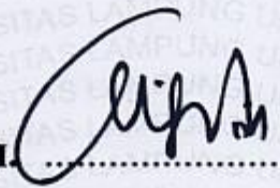
Ketua

: **Dr. Ir. Kurnia Muludi, M.S.Sc.**



Penguji I
Sekretaris

: **Ir. Gigih Forda Nama, S.T., M.T.I.**




Penguji II
Penguji Pembahas

: **Favorisen R. Lumbanraja, Ph.D.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam




Dr. Eng. Suripto Dwi Yuwono, S.Si., M.T.
NIP. 19740705 200003 1 001

Tanggal Lulus Ujian Skripsi : **13 Desember 2021**

PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama: Nonik Aisyah

NPM: 1717051011

Dengan ini menyatakan bahwa skripsi yang berjudul “IMPLEMENTASI ALGORITMA *K-NEAREST NEIGHBOR* UNTUK SISTEM REKOMENDASI TOPIK PENELITIAN PADA JURUSAN ILMU KOMPUTER UNIVERSITAS LAMPUNG BERBASIS KLASIFIKASI” adalah benar hasil karya sendiri dan bukan orang lain. Seluruh tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Jika dikemudian hari terbukti skripsi penulis adalah hasil penjiplakan atau dibuat oleh orang lain, maka penulis bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Bandar Lampung, 15 Desember 2021

Penulis



Nonik Aisyah

NPM. 1717051011

RIWAYAT HIDUP



Penulis dilahirkan di Dipasena pada tanggal 22 Agustus 1999 sebagai anak kedua dari dua bersaudara pasangan Bapak Kasno dan Ibu Suwarni. Penulis menyelesaikan Pendidikan Sekolah Dasar (SD) di SD Negeri 01 Bumi Dipasena Jaya pada tahun 2011. Kemudian pendidikan Sekolah Menengah Pertama (SMP) di SMP Negeri 1 Tulang Bawang Udik yang diselesaikan pada tahun 2014. Kemudian melanjutkan pendidikan Sekolah Menengah Atas (SMA) di SMA Negeri 1 Tumijajar yang diselesaikan pada tahun 2017.

Penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung pada tahun 2017 melalui jalur Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN). Selama menjadi mahasiswa, penulis melakukan beberapa kegiatan antara lain.

1. Menjadi anggota Adapter Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2017/2018.
2. Menjadi anggota pengurus di Bidang Keilmuan Himpunan Mahasiswa Jurusan Ilmu Komputer (Himakom) pada periode 2017/2018.
3. Menjadi Asisten Dosen dan Asisten Laboratorium Jurusan Ilmu Komputer untuk mata kuliah Keamanan Sistem Informasi pada periode semester ganjil tahun ajaran 2019/2020.

4. Menjadi Sekretaris di Bidang Keilmuan Himpunan Mahasiswa Jurusan Ilmu Komputer (Himakom) pada periode 2019.
5. Melaksanakan Kuliah Kerja Nyata (KKN) pada periode II tahun 2020 di Bumi Dipasena Jaya, Kecamatan Rawajitu Timur, Kabupaten Tulang Bawang.
6. Melaksanakan Kerja Praktik di Kantor Badan Pusat Statistik Tulang Bawang Barat pada tahun 2020.
7. Mengikuti ujian sertifikasi dan mendapatkan sertifikat *Junior Office Operator* (JOO) yang diselenggarakan oleh Badan Nasional Sertifikasi Profesi (BNSP) pada tahun 2020.

MOTTO

مَنْ خَرَجَ فِي طَلَبِ الْعِلْمِ فَهُوَ فِي سَبِيلِ اللَّهِ

“Barang siapa keluar untuk mencari ilmu maka dia berada di jalan Allah.”

“Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila engkau telah selesai (dari sesuatu urusan), tetaplah bekerja keras (untuk urusan yang lain). Dan hanya kepada Tuhanmulah engkau berharap.”

(QS. Al-Insyirah: 6-8)

“Allah akan meninggikan orang-orang yang beriman diantara kamu dan orang-orang yang berilmu pengetahuan beberapa derajat. Dan Allah Maha mengetahui apa yang kamu kerjakan”

(Al-Mujadillah: 11)

“Barangsiapa menempuh jalan untuk mendapatkan ilmu, Allah akan memudahkan baginya jalan menuju surga.”

(HR Muslim)

PERSEMBAHAN

Alhamdulillahirobbilalamin

Puji syukur kepada Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga telah memberikan kemudahan dalam menyelesaikan penulisan skripsi ini.

Shalawat serta salam selalu disanjungkan kepada Nabi Muhammad SAW.

Aku persembahkan karya ini kepada:

Bapak dan Mamak

Sebagai tanda bakti, hormat, dan terima kasihku kepada bapak dan mamak tercinta yang telah mendidik dan membesarkanku dengan kasih sayang, mendukung dan mendoakanku untuk meraih kesuksesan dunia dan akhirat. Terima kasih atas semua pengorbanan, perjuangan, dan kasih sayang tiada hentinya. Terima kasih Bapak dan Mamak untuk segalanya.

Kakakku tercinta Muhammad Wahyudi

Terima kasih telah memberikan semangat, dukungan dan do'a, serta kebahagiaan yang kalian berikan untukku.

Seluruh Keluarga Besar, Sahabat, dan Teman-teman yang selalu memberikan semangat dan dukungan.

Almamater Tercinta, Universitas Lampung

SANWACANA

Alhamdulillahirobbil alamiin, puji dan syukur ke hadirat Allah SWT yang telah memberikan rahmat dan hidayahNya sehingga saya dapat menyelesaikan skripsi dengan judul penelitian “Implementasi Algoritma *K-nearest neighbor* Untuk Sistem Rekomendasi Topik Penelitian Pada Jurusan Ilmu Komputer Universitas Lampung Berbasis Klasifikasi”. Shalawat dan salam senantiasa tercurahkan kepada Nabi Muhammad SAW.

Selama penyusunan skripsi ini tidak terlepas dari dukungan banyak pihak yang telah membantu, membimbing dan memberikan semangat kepada saya. Secara tertulis saya mengungkapkan terima kasih kepada:

1. Kedua orang tua tercinta Bapak Kasno dan Ibu Suwarni, Kakak Muhammad Wahyudi, serta Bude dan keluarga besar. Terima kasih atas semua jerih payah selama membesarkan saya, merawat, mencintai, membiayai, serta selalu mendoakan tanpa henti untuk keberhasilan dan kesuksesan. Kasih sayang dan ucapan terima kasih selalu diberikan untuk kalian.
2. Bapak Dr. Ir. Kurnia Muludi, M.S.Sc. sebagai pembimbing utama yang telah membimbing, memberikan arahan, kritik, saran, serta membina dalam menyelesaikan skripsi ini agar menjadi lebih baik.
3. Bapak Ir. Gigih Forda Nama, S.T., M.T.I. sebagai pembimbing kedua yang telah membimbing saya dalam memberikan arahan, kritik, saran sehingga penulisan skripsi ini dapat diselesaikan dengan baik.
4. Favorisen R. Lumbanraja, Ph. D. sebagai pembahas yang telah memberikan masukan yang sangat bermanfaat dalam penulisan dan perbaikan skripsi ini.
5. Bapak Dwi Sakethi, S.Si., M.kom. selaku pembimbing akademik penulis yang telah memberikan bimbingan, masukan, arahan, dan nasihat sehingga penulis dapat menyelesaikan proses belajar.
6. Bapak Dr. Suripto Dwi Yuwono, M.T. selaku Dekan FMIPA Universitas Lampung.

7. Ibu Astria Hijriani, S.Kom., M.Kom. selaku sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
8. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu dan pengalaman dalam hidup untuk menjadi lebih baik.
9. Ibu Ade Nora Maela, Kak Zainuddin dan Kak Ardi Novalia yang telah membantu segala urusan administrasi penulis di Jurusan Ilmu Komputer.
10. Keluarga besarku tercinta lainnya yang tidak dapat disebutkan satu per satu yang selalu memberi dukungan dan doa agar dapat menyelesaikan perkuliahan ini dengan baik dan lancar.
11. Diana Rohma selaku teman kosan, sahabat, serta *partner* dalam mengerjakan skripsi yang telah banyak membantu, mendukung, memberikan semangat serta doa kepada saya selama mengerjakan skripsi.
12. Intan Indah Larasati selaku teman kosan dan sahabat dari masa sekolah yang telah memberikan dukungan, semangat, nasihat, serta doa kepada saya selama pengerjaan skripsi ini.
13. Eki Tri Suenda dan Wildan Muttaqin yang banyak membantu dan memberi dukungan kepada saya dalam menyelesaikan skripsi ini.
14. Adam Malik dan Muhammad Bella BN yang selalu memberikan semangat dan dukungan dalam menyelesaikan skripsi ini.
15. Reka Amelia, Reda Meiningtyas, dan Noverina Ika Tama yang telah sama-sama berjuang dalam menjalani perkuliahan ini, saling mendukung serta memberi semangat dan do'a kepada penulis dalam menyelesaikan perkuliahan ini.
16. Teman-teman Goyang Squad yang telah menemani penulis dalam menjalani hari-hari dan saling memberi semangat selama masa perkuliahan.
17. Teman-teman pada masa sekolah yang telah mendengarkan keluh kesah dan telah memberikan doa kepada saya untuk menyelesaikan skripsi ini.
18. Teman-teman Jurusan Ilmu Komputer Angkatan 2017 yang menjadi keluarga satu angkatan selama menjalankan masa studi di Jurusan Ilmu Komputer Universitas Lampung.

Dalam proses penyusunan skripsi ini terdapat kekurangan yang disebabkan oleh keterbatasan pengalaman dan pengetahuan saya. Saya sangat mengharapkan kritik dan saran dari pihak pembaca sebagai pembelajaran untuk saya. Semoga isi dari skripsi ini dapat manfaat bagi pihak yang membaca.

Bandar Lampung, 15 Desember 2021

Penulis

Nonik Aisyah

NPM. 1717051011

DAFTAR ISI

Halaman

DAFTAR ISI	v
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
DAFTAR KODE PROGRAM	xi
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
II. TINJAUAN PUSTAKA	5
2.1 Klasifikasi	5
2.2 Data Mining	5
2.3 K- Nearest Neighbor	6
2.4 Data Cleaning	7
2.5 Data Transformation	8
2.6 Correlation-based Feature Selection	8
2.7 Synthetic Minority Over-Sampling Technique (SMOTE)	9
2.8 K-Fold Cross Validation	9
2.9 Grid Search	10
2.10 Confusion Matrix	10
2.11 Sistem Rekomendasi	12
2.12 Unified Modelling Language (UML)	12
2.13 Python	14

2.14 <i>Flask</i>	16
2.15 Metode <i>Extreme Programming</i>	17
2.16 <i>Balsamiq Mockup</i>	18
2.17 <i>Black-box Testing</i>	18
2.18 <i>Equivalence Partitioning</i>	19
2.19 Penelitian Terdahulu	20
III. METODOLOGI PENELITIAN	22
3.1 Waktu dan Tempat Penelitian	22
3.1.1 Tempat Penelitian	22
3.1.2 Waktu Penelitian	22
3.1.3 Data	23
3.1.4 Alat	23
3.2 Tahapan Penelitian	23
3.2.1 Studi Literatur	24
3.2.2 Pengumpulan Data	24
3.2.3 Implementasi <i>Machine Learning</i>	25
3.2.4 Implementasi Sistem	27
3.2.5 Penulisan Laporan	40
IV. HASIL DAN PEMBAHASAN	41
4.1 Hasil	41
4.1.1 <i>Data Cleaning</i>	41
4.1.2 <i>Data Transformation</i>	42
4.1.3 <i>Correlation-based Feature Selection</i>	43
4.1.4 <i>Sythetic Minority Over-Sampling Technique (SMOTE)</i>	44
4.1.5 Pembagian Data	46
4.1.6 <i>Error Rate</i> Nilai k	47
4.1.7 <i>Grid Search</i>	47
4.1.8 Evaluasi	48
4.2 Pembahasan	50
4.2.1 Implementasi <i>Machine Learning</i>	50
4.2.2 Dokumentasi Metode <i>Extreme Programming</i> Pada Tahap Pengembangan Sistem	58
4.2.3 Implementasi Sistem	66
4.3 Hasil Pengujian	72

V. SIMPULAN DAN SARAN.....	77
5.1 Simpulan	77
5.2 Saran	78
DAFTAR PUSTAKA	79

DAFTAR TABEL

Tabel	Halaman
Tabel 1. Pedoman Untuk Memberikan Interpretasi Koefisien Korelasi (Sugiono, 2016).	9
Tabel 2. Confusion Matrix (Rahman et al., 2017).	11
Tabel 3. Use Case Diagram (Satzinger et al., 2016).	13
Tabel 4. Activity Diagram (Satzinger et al., 2016).	14
Tabel 5. Pengujian Equivalence Partitioning (Setiawan et al., 2020)	19
Tabel 6. Penelitian Terdahulu	20
Tabel 7. User Stories	27
Tabel 8. Skenario Pengujian	38
Tabel 9. Data <i>Missing Value</i>	41
Tabel 10. Data Setelah Dilakukan Proses Data Cleaning	42
Tabel 11. Data Sebelum Dilakukan Proses Data Transformation	43
Tabel 12. Data Setelah Dilakukan Proses Data Transformation	43
Tabel 13. Perbandingan jumlah k pada SMOTE	45
Tabel 14. Hasil Perbandingan Pembagian Data	46
Tabel 15. Hasil grid search	47
Tabel 16. Hasil nilai Confusion Matrix	49
Tabel 17. Hasil Evaluasi	50
Tabel 18. Implementasi Tahap Pengembangan Sistem Metode Extreme Programming	59
Tabel 19. Hasil Pengujian Black-box	73

DAFTAR GAMBAR

Gambar	Halaman
Gambar 1. Representasi 10 folds cross validation	10
Gambar 2. Metode Extreme Programming (Pressman, 2020).	17
Gambar 3. Alur Penelitian	24
Gambar 4. Flowchart Model Machine Learning.....	25
Gambar 5. Use Case Diagram Sistem Rekomendasi Topik Penelitian.....	28
Gambar 6. Activity Diagram Login.	29
Gambar 7. Activity Diagram Mengelola Data Mahasiswa.	30
Gambar 8. Activity Diagram Prediksi.....	30
Gambar 9. Activity Diagram Kategori Penelitian.....	31
Gambar 10. Class Responsibility Collaborator Cards.....	31
Gambar 11. Rancangan Tampilan Login.	32
Gambar 12. Rancangan Tampilan Data Mahasiswa.	33
Gambar 13. Rancangan Tampilan Tambah Data	33
Gambar 14. Rancangan Tampilan Edit Data	34
Gambar 15. Rancangan Tampilan Beranda	34
Gambar 16. Rancangan Tampilan Tambah Prediksi.....	35
Gambar 17. Rancangan Tampilan Hasil Prediksi	36
Gambar 18. Rancangan Tampilan Kategori Penelitian.....	36
Gambar 19. Rancangan Tampilan Lihat Kategori Penelitian	37
Gambar 20. Nilai Korelasi.	44
Gambar 21. Data Sebelum Dilakukan Proses SMOTE.....	45
Gambar 22. Data Setelah Dilakukan Proses SMOTE.....	46
Gambar 23. Grafik Error Rate Nilai k.....	47
Gambar 24. Hasil Confusion Matrix.....	49
Gambar 25. Tampilan Data Mahasiswa.....	62
Gambar 26. Perubahan Tampilan Data Mahasiswa	62
Gambar 27. Tampilan Tambah Data Mahasiswa.....	63
Gambar 28. Tampilan Edit Data Mahasiswa	63
Gambar 29. Perubahan Tampilan Tambah Data Mahasiswa	64
Gambar 30. Perubahan Tampilan Edit Data Mahasiswa	64
Gambar 31. Tampilan Hasil Prediksi	65
Gambar 32. Tampilan Kategori Penelitian	65
Gambar 33. Tampilan Sweet Alert	66
Gambar 34. Tampilan Halaman Data Mahasiswa.....	67
Gambar 35. Tampilan Halaman Tambah Data.	67
Gambar 36. Tampilan Halaman Edit.	68
Gambar 37. Tampilan Halaman Hapus Data.	69

Gambar 38. Tampilan Halaman Beranda.....	69
Gambar 39. Tampilan Halaman Prediksi.....	70
Gambar 40. Tampilan Pop Up Judul Skripsi.	71
Gambar 41. Tampilan Halaman Login.	71

DAFTAR KODE PROGRAM

Kode Program	Halaman
Kode Program 1. Kode Proses Data Transformation.....	51
Kode Program 2. Kode Menampilkan Hasil Korelasi	53
Kode Program 3. Kode Menjalankan Proses SMOTE.....	54
Kode Program 4. Kode Melakukan Proses Pembagian Data.....	55
Kode Program 5. Kode Menampilkan Error Rate Nilai k.....	55
Kode Program 6. Kode Menjalankan Proses Grid Search	56
Kode Program 7. Kode Menampilkan Evaluasi.....	57

I. PENDAHULUAN

1.1 Latar Belakang

Topik penelitian merupakan landasan dasar yang dimiliki mahasiswa dalam melakukan sebuah penelitian. Topik penelitian ini sangat penting bagi mahasiswa tingkat akhir dalam menyelesaikan skripsi. Topik penelitian di setiap Jurusan pun berbeda-beda sesuai dengan bidangnya masing-masing salah satunya di Jurusan Ilmu Komputer Universitas Lampung.

Dalam menentukan topik penelitian, mahasiswa masih merasa kesulitan dan harus mengenali kemampuan yang dimilikinya untuk mempermudah dalam proses pengerjaan penelitian tersebut. Penentuan topik penelitian dilakukan dengan melihat beberapa kemiripan nilai mata kuliah yang sudah ditempuh dengan nilai mahasiswa sebelumnya yang sudah menyelesaikan perkuliahan. Adapun untuk mempermudah dalam menentukan topik penelitian dengan memanfaatkan sistem rekomendasi.

Sistem rekomendasi adalah suatu konsep yang menetapkan item untuk disarankan kepada pengguna. Sistem rekomendasi dapat mengidentifikasi topik penelitian yang sesuai dengan mahasiswa berdasarkan dari nilai mata kuliah. Sistem rekomendasi dapat membantu pengguna dalam pengambilan keputusan dengan memberikan rekomendasi spesifik.

Atas permasalahan tersebut, untuk membantu mahasiswa dalam memberikan wawasan tentang topik penelitian yang sesuai maka dibutuhkan sebuah analisis pengambilan keputusan yang dapat merekomendasikan mahasiswa dalam menentukan topik penelitian secara cepat dan tepat. Sistem yang akan dibuat menggunakan algoritma *K-nearest neighbor* (K-NN).

Algoritma ini dipilih karena K-NN merupakan suatu model pendukung keputusan yang dapat mengklasifikasikan data berdasarkan jarak terdekat. Algoritma *K-nearest neighbor* (K-NN) memiliki tujuan yaitu untuk mengklasifikasi objek baru berdasarkan atribut dan *training sample* atau data latih (Kartika et al., 2017).

Berdasarkan latar belakang di atas, maka dibuat sebuah penelitian dengan judul “Implementasi Algoritma *K-nearest neighbor* Untuk Sistem Rekomendasi Topik Penelitian Pada Jurusan Ilmu Komputer Universitas Lampung Berbasis Klasifikasi”. Penelitian ini membahas mengenai pemilihan topik penelitian yang ditentukan melalui nilai-nilai mata kuliah yang sudah diambil dan dilakukan kemiripan atau jarak terdekat dengan mahasiswa sebelumnya yang sudah lulus.

1.2 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah Bagaimana cara merekomendasikan topik penelitian menggunakan algoritma K-NN berdasarkan data nilai mahasiswa.

1.3 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah :

1. Dalam merekomendasikan topik penelitian ini menggunakan nilai mata kuliah mahasiswa yang akan menentukan topik penelitian berdasarkan nilai dan topik penelitian mahasiswa yang sudah lulus.
2. Pengimplementasian sistem rekomendasi metode K-NN dengan menggunakan bahasa pemrograman Python.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah merekomendasikan topik penelitian pada mahasiswa Jurusan Ilmu Komputer Universitas Lampung menggunakan algoritma K-NN.

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah :

1. Mengetahui rekomendasi topik penelitian pada mahasiswa berdasarkan penilaian jarak terdekat dengan nilai mahasiswa yang sudah lulus.
2. Mengetahui cara kerja metode K-NN dalam merekomendasikan topik penelitian.

II. TINJAUAN PUSTAKA

2.1 Klasifikasi

Klasifikasi merupakan suatu pengelompokan data yang mempunyai target atau kelas label (Wahyono et al., 2020). Klasifikasi diterapkan untuk menentukan suatu keputusan sesuai dengan pengolahan data lampau dan menggunakan perhitungan algoritma. Klasifikasi saat ini banyak digunakan dalam berbagai bidang seperti bidang kesehatan, pendidikan, komputer dan bidang lainnya. Algoritma yang digunakan dalam klasifikasi tersebut termasuk ke dalam *supervised learning* atau pembelajaran yang diawasi, di mana target atau data label berperan sebagai guru yang mengawasi pembelajaran untuk mencapai akurasi tertentu. Algoritma yang biasanya digunakan dalam menyelesaikan masalah klasifikasi, yaitu Naïve bayes, *K-nearest neighbor*, *Decision Tree* dan lainnya.

2.2 Data Mining

Data mining adalah suatu proses dalam mencari pengetahuan menggunakan pola dari data dalam jumlah yang besar. Sumber data yang digunakan berupa *database*, *data warehouse*, *web*, atau penyimpanan informasi lainnya. *Data mining* sering juga disebut dengan *Knowledge Discovery from Data* atau KD (Han et al., 2014). Adapun proses dalam KDD sebagai berikut :

- a. *Data cleaning* (untuk menghilangkan data yang tidak konsisten)
- b. *Data integration* (berbagai data dapat digabungkan)
- c. *Data selection* (data yang relevan diambil dari *database*)
- d. *Data transformation* (transformasi data untuk di proses pada *data mining*)
- e. *Data mining* (proses mencari pola menggunakan metode tertentu)

- f. *Pattern evaluation* (mengidentifikasi pola-pola yang menarik dan bakal menjadi pengetahuan)
- g. *Knowledge presentation* (teknik visualisasi dan representasi dalam menyajikan pengetahuan yang diperoleh kepada pengguna)

Data mining terbagi dalam beberapa tugas yang dapat dilakukan, diantaranya (Kantardzic, 2020) :

- a. *Classification* yaitu fungsi yang mengklasifikasikan suatu item menjadi beberapa kelas target yang telah ditentukan
- b. *Regression* yaitu fungsi yang mengkategorikan data ke dalam variabel prediksi
- c. *Clustering* yaitu fungsi yang mengidentifikasi data ke beberapa *cluster*
- d. *Summarization* yaitu fungsi tambahan untuk menemukan deskripsi *subset* data
- e. *Dependency modeling* yaitu suatu model menggambarkan hubungan antara variabel dalam data set
- f. *Change and deviation* yaitu untuk menemukan perubahan dalam data set.

2.3 K- Nearest Neighbor

K-nearest neighbor merupakan algoritma *supervised learning* di mana dalam melakukan klasifikasi berdasarkan data *training*. Setiap data *training* mewakili satu dari data kelas, ketika data uji tidak diketahui kelasnya maka K-NN akan mencari data yang terdekat dengan data uji. Kedekatan ini didefinisikan dalam istilah *euclidean distance*, di mana *euclidean distance* berada diantara dua titik, $X = (x_1, x_2, \dots, x_n)$ dan $Y = (y_1, y_2, \dots, y_n)$ dilambangkan oleh $d(X, Y)$. Klasifikasi nearest neighbor memberikan nilai atau bobot yang sama setiap masing-masing atribut, sehingga dapat digunakan untuk melakukan prediksi (Gorade et al., 2017). Algoritma K-NN merupakan suatu metode yang dilakukan untuk klasifikasi terhadap objek baru berdasarkan (K) tetangga terdekat. K-NN termasuk dalam algoritma *supervised learning*, di mana dalam melakukan klasifikasi berdasarkan mayoritas dari kategori K-NN. Langkah-langkah untuk menghitung algoritma K-NN yaitu sebagai berikut :

1. Menentukan nilai K (jumlah tetangga terdekat)
2. Menghitung jarak pada semua objek terhadap data *training*
3. Mengurutkan objek-objek ke dalam kelompok yang mempunyai jarak terdekat
4. Mengumpulkan kategori Y (klasifikasi nearest neighbors)
5. Memilih kategori mayoritas sebagai klasifikasi objek baru

Jumlah nilai K dan rumus jarak dari data pengujian ditentukan dengan menggunakan 4 rumus perhitungan yaitu, rumus *Euclidean Distance*, *Minkowski*, *Manhattan* dan *Chebyshev* (Rahmawati et al., 2021).

Rumus *Euclidean Distance* dapat dilihat pada persamaan 1 :

$$D_{(a,b)} = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2}$$

Rumus *Minkowski* dapat dilihat pada persamaan 2 :

$$D_{(a,b)} = \sqrt[p]{(a_1 - b_1)^p + (a_2 - b_2)^p + \dots + (a_n - b_n)^p}$$

Rumus *Manhattan* dapat dilihat pada persamaan 3 :

$$D_{(a,b)} = |a_1 - b_1| + \dots + |a_n - b_n|$$

Rumus *Chebyshev* dapat dilihat pada persamaan 4 :

$$D_{(a,b)} = \max (|a_1 - b_1| + \dots + |a_n - b_n|)$$

Keterangan :

- $D_{(a,b)}$ = Jarak dari kedua parameter a dan b
- a = Nilai data pada dataset
- b = Nilai data tes objek
- n = Batas jumlah banyaknya data

2.4 Data Cleaning

Data cleaning merupakan proses mengolah data untuk dilakukan analisis dengan cara mengatasi nilai yang hilang, *noise*, dan data yang tidak konsisten. *Data cleaning* dibagi menjadi dua yaitu *data cleaning* untuk *missing value* dan *data cleaning* untuk *noisy data*. *Missing value* dapat diatasi dengan mengabaikan *tuple*, mengisi nilai yang hilang secara manual, serta menggunakan nilai median atau modus untuk mengisi nilai yang hilang. Pada *noisy data* merupakan kesalahan dalam variabel yang diukur, cara mengatasi

noisy data dapat dilakukan dengan teknik *binning*, *regression*, dan *clustering* (Han et al., 2014).

2.5 *Data Transformation*

Data Transformation yaitu proses mengubah data ke dalam bentuk yang sesuai untuk meningkatkan kinerja pada proses selanjutnya sehingga lebih mudah untuk diklasifikasikan pada data mining (Xue et al., 2017). *Data Transformation* dilakukan dengan mengubah format atau bentuk data sehingga sangat memudahkan pengguna dalam melakukan proses *mining* dan memahami hasil yang didapatkan (Hadikristanto & Pungkas, 2019).

2.6 *Correlation-based Feature Selection*

Feature Selection yaitu suatu proses pengurangan pada jumlah *variable* yang dimasukkan atau *input* untuk meningkatkan kinerja pemodelan *machine learning*. Metode ini melibatkan hubungan antara *variable input* dan target dengan pengukuran statistik. Terdapat dua jenis dalam *feature selection* yaitu *supervised* dan *unsupervised*, pada *supervised* dapat dibagi menjadi tiga kelompok yaitu metode *wrapper*, *filter*, dan *intrinsic*. Pada metode pemilihan fitur berbasis *filter* menggunakan perhitungan statistik untuk menilai korelasi antara *variable* yang disebut dengan *Correlation-based Feature Selection* (CFS) (Brownlee, 2020). *Correlation-based Feature Selection* merupakan metode sederhana yang memilih fitur berdasarkan hasil dari korelasi. Pemilihan fitur dilakukan jika terdapat *variable* yang berkorelasi tinggi dengan kelas dan tidak terdapat korelasi antar *variable*. Fitur yang berlebihan dilakukan pengurangan karena akan berkorelasi tinggi dengan fitur yang tersisa (Hall, 1999). Penentuan rendah atau tingginya suatu hubungan korelasi antar *variable* dapat dilihat pada Tabel 1.

Tabel 1. Pedoman Untuk Memberikan Interpretasi Koefisien Korelasi (Sugiono, 2016).

Interval Koefisien	Tingkat Hubungan
0,00 – 0,199	Sangat Rendah
0,20 – 0,399	Rendah
0,40 – 0,599	Sedang
0,60 – 0,799	Kuat
0,80 – 1,000	Sangat Kuat



2.7 *Sythetic Minority Over-Sampling Technique (SMOTE)*

Sythetic Minority Over-Sampling Technique (SMOTE) merupakan teknik mensintesis sampel baru dari kelas minoritas dengan cara sampling ulang sampel kelas minoritas sehingga menyeimbangkan data. SMOTE bekerja dengan membuat replikasi dari data minoritas yang dikenal disebut dengan (*syntetic data*). SMOTE bekerja dengan algoritma *k-nearest neighbors* yaitu dengan tetangga terdekat sebanyak *k* pada setiap data kelas minoritas, kemudian dibuat data sintesis sebanyak presentase duplikasi yang diinginkan antara data minoritas dengan *k-nearest neighbors* yang dipilih secara acak (Siringoringo, 2018).

2.8 *K-Fold Cross Validation*

K-fold merupakan metode *Cross Validation* dilakukan menggunakan partisi secara acak pada data *training* sebanyak nilai *K* serta melakukan percobaan sebanyak nilai tersebut. Setiap percobaan menggunakan data *testing* sebanyak partisi ke-*K* dan sisa partisi lainnya digunakan sebagai data *training* (Syafira, 2017). Penelitian membagi data menjadi 10 bagian, 90% data *training* dan 10 lainnya dijadikan sebagai data *testing*. Proses ini dikenal dengan istilah *10 folds cross validation* yang banyak digunakan peneliti karena menghasilkan algoritma yang stabil (Indrayanti et al., 2017). Representasi *10 folds cross validation* dapat dilihat pada Gambar 1.

	Dataset dibagi menjadi 10 bagian secara <i>random</i> (acak)										Akurasi
	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%	100%
Percobaan 1	10%										a1
Percobaan 2		10%									a2
Percobaan 3			10%								a3
Percobaan 4				10%							a4
Percobaan 5					10%						a5
Percobaan 6						10%					a6
Percobaan 7							10%				a7
Percobaan 8								10%			a8
Percobaan 9									10%		a9
Percobaan 10										10%	a10

Keterangan gambar:
 = data testing
 = data training

Gambar 1. Representasi 10 *folds* cross validation

2.9 Grid Search

Grid search merupakan kelas library yang disediakan oleh scikit-learn untuk mencari kombinasi parameter yang menghasilkan performa terbaik. *Grid search* akan menyimpan parameter yang dicari kemudian akan dilakukan perhitungan model dengan semua parameter yang tersimpan dan mendapatkan parameter terbaik serta validasi silang terbaik pada semua data. Parameter terbaik disimpan pada atribut `bestparams_` dan skor terbaik disimpan pada atribut `best_score_attribute` (Shuai et al., 2019).

2.10 Confusion Matrix

Confusion matrix adalah metode yang digunakan untuk perhitungan akurasi pada data mining atau sistem pendukung keputusan. *Confusion matrix* dapat direpresentasikan dengan tabel yang menjelaskan jumlah data uji benar dan jumlah data uji salah (Rahman et al., 2017). Representasi *confusion matrix* dapat dilihat pada Tabel 2.

Tabel 2. *Confusion Matrix* (Rahman et al., 2017).

<i>Correct Classification</i>	<i>Classified as</i>	
	<i>Predicted “+”</i>	<i>Predicted “-”</i>
<i>Actual “+”</i>	<i>True Positives (TP)</i>	<i>False Negatives (FN)</i>
<i>Actual “-”</i>	<i>False Positives (FP)</i>	<i>True Negatives (TN)</i>

Berdasarkan tabel diatas :

1. *True Positives* (TP) yaitu jumlah data positif yang diklasifikasikan sebagai nilai positif
2. *False Positives* (FP) yaitu jumlah data negatif yang diklasifikasikan sebagai nilai positif
3. *False Negatives* (FN) yaitu jumlah data positif yang diklasifikasikan sebagai nilai positif
4. *True Negatives* (TN) yaitu jumlah data negatif yang diklasifikasikan sebagai nilai negatif

Perhitungan performa kinerja dari klasifikasi dapat dihitung menggunakan persamaan berikut (Hidayatullah et al., 2019):

1. *Accuracy* adalah pengukuran persentase yang dilakukan dengan membandingkan jumlah data yang diklasifikasikan benar oleh *classifier* dengan keseluruhan data. Rumus *accuracy* dapat dilihat pada persamaan 2 :

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \times 100\%$$

2. *Precision* adalah ukuran ketepatan dengan membandingkan sampel kelas positif yang diklasifikasikan dengan benar terhadap jumlah kelas positif. Rumus *precision* dapat dilihat pada persamaan 3:

$$Precision = \frac{TP}{TP+FP} \times 100\%$$

3. *Recall* adalah ukuran kelengkapan dengan membandingkan sampel kelas positif yang diklasifikasikan dengan benar terhadap sampel kelas positif yang salah diklasifikasikan ke kelas negative. Rumus *recall* dapat dilihat pada persamaan 4:

$$Recall = \frac{TP}{TP+Fn} \times 100\%$$

4. *F1-Score* adalah ukuran rata-rata kebalikan dari *precision* dan *recall*.

Rumus *f1-score* dapat dilihat pada persamaan 5:

$$F1-Score = \frac{2*Precision*Recall}{Precision+Recall} \times 100\%$$

2.11 Sistem Rekomendasi

Sistem rekomendasi adalah suatu konsep yang menetapkan item (buku, produk, film) untuk disarankan kepada pengguna (pelanggan, pengunjung halaman web, pembaca artikel online). Sistem memberikan rekomendasi berdasarkan informasi tentang pengguna, informasi ini diperoleh menggunakan data mentah dari berbagai sumber seperti internet, media dan database. Proses rekomendasi dibagi menjadi 3 tahap (Narke & Nasreen, 2020):

1. Tahap pengumpulan informasi : Tahap ini mengambil data pengguna dari berbagai sumber yang telah disebutkan di atas.
2. Tahap pembelajaran : Tahap ini menggunakan teknik dan algoritma pada informasi yang akan dikumpulkan untuk mendapatkan wawasan yang bermakna dari informasi yang dibutuhkan selanjutnya.
3. Tahap rekomendasi : Tahap ini adalah tahap terakhir di mana data yang diproses pada tahap pertama dan tahap selanjutnya, rekomendasi dibuat untuk pengguna dalam bentuk saran, konten atau layanan. *Feedback* yang diberikan setelah tahap rekomendasi yaitu tindakan pengguna seperti penilaian, komentar, dan link yang dikunjungi.

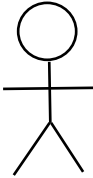



2.12 Unified Modelling Language (UML)

UML adalah suatu bahasa yang sudah menjadi standar untuk membuat desain dan analisis, mendefinisikan requirement, dan menggambarkan arsitektur dalam pemrograman berorientasi objek (Putra & Andriani, 2019). Tujuan dari penggunaan UML antara lain membuat suatu bahasa pemodelan sesuai kebutuhan dari berbagai proses rekayasa dan bahasa pemrograman, memberikan suatu model siap pakai dan bahasa pemodelan visual ekspresif sehingga mudah dimengerti secara umum, menyatukan dalam bentuk

pemodelan dari semua praktek terbaik (Wati & Kusumo, 2016). Diagram-diagram yang digunakan dalam UML antara lain *Use Case Diagram*, *Class Diagram*, *Sequence Diagram*, *Activity Diagram*, *Collaboration Diagram*, *Statechart Diagram*, dan *Deployment Diagram*.

- a. *Use Case Diagram* yaitu suatu aktivitas yang menggambarkan interaksi antara pengguna dengan sistem (Satzinger et al., 2016). Contoh bagian-bagian *use case diagram* dapat dilihat pada Tabel 3.




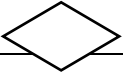
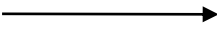

Tabel 3. Use Case Diagram (Satzinger et al., 2016).

No	Simbol	Keterangan
1.		<i>Actor</i> adalah pengguna yang menggunakan sistem
2.		Garis penghubung yang menghubungkan antara <i>actor</i> dengan objek
3.		Bentuk elips yang mempresentasikan kerja yang dilakukan oleh <i>actor</i>
4.		Bentuk kotak yang menampilkan proses kerja pada sistem

- b. *Activity Diagram*

Activity Diagram yaitu diagram yang menggambarkan alur kerja yang terjadi antara pengguna dengan sistem (Satzinger et al., 2016). Contoh bagian-bagian *activity diagram* dapat dilihat pada Tabel 4.

Tabel 4. *Activity Diagram* (Satzinger et al., 2016).

No	Simbol	Keterangan
1.		<i>Star node</i> yaitu titik awal dari suatu aktivitas
2.		<i>End node</i> yaitu titik akhir dari suatu aktivitas
3.		<i>Rounded rectangle</i> yaitu bentuk dari fungsi pada sistem yang spesifik
4.		<i>Decision diamonds</i> yaitu bentuk pilihan kerja yang memiliki cabang pilihan untuk mengambil keputusan
5.		<i>Arrows</i> atau tanda panah yang menunjukkan arah kerja pada sistem
6.		<i>Horizontal lines</i> yaitu garis yang menunjukkan aktivitas paralel

2.13 Python

Python merupakan bahasa pemrograman yang multiguna dengan sejarah perancangan yang berpusat terhadap keterbacaan kode. Python dijadikan sebagai suatu bahasa yang menggabungkan kapabilitas, kemampuan, dengan suatu sintaks kode yang jelas, dilengkapi dengan fungsionalitas pustaka yang besar dan komperhensif. Python pada dasarnya digunakan sebagai bahasa *script* walaupun dalam praktiknya penggunaan bahasa tersebut lebih luas. Python digunakan di berbagai bidang pengembangan perangkat lunak serta berjalan di berbagai platform sistem operasi. Platform sistem operasi yang dijalankan menggunakan kode Python diantaranya adalah :

- a. Linux/Unix
- b. Windows
- c. Mac OS X
- d. Java Virtual Machine
- e. Amiga
- f. Palm
- g. Symbian (produk-produk Nokia)

Python saat ini didistribusikan di berbagai lisensi yang berbeda-beda dari beberapa versi. Prinsip Python yaitu dapat diperoleh secara bebas bagi kepentingan komersial (Syahrudin & Kurniawan, 2018). Python merupakan bahasa pemrograman yang dianggap paling populer serta banyak diminati. Survei dari *stack overflow* menunjukkan bahwa python telah mengambil alih bahasa diantaranya Java, C, C++. Berikut ini alasan Python menjadi begitu populer, antara lain (Vinothraj et al., 2017) :

- a. *Simple and easy to learn* : Python bahasa yang sangat mudah dipahami karena mirip dengan bahasa inggris.
- b. *Supportive community* : Python telah berada selama 30 tahun, banyak komunitas yang mendukung perkembangan bahasa tersebut. Pelajar Python dari berbagai usia dan berbagai tingkat keahlian sangat mudah dalam menemukan referensi belajar seperti dari dokumentasi resmi tutorial YouTube.
- c. *Web development* : Pemrograman web Python memiliki *framework* untuk mengembangkan situs web seperti Django, *Flask*, pylon dan so on.
- d. *Use in big data and machine learning* : Big data dan machine learning merupakan trends di ilmu komputer saat ini yang dapat membantu perusahaan dalam mengubah proses dan alur kerja. Python termasuk ke dalam bahasa yang banyak digunakan oleh peneliti dan pengembang saat ini. *Library* Python seperti OpenCV untuk *computer vision* dan TensorFlow untuk jaringan saraf.
- e. *Efficiency* : Python menerapkan pemrograman yang berbeda dari bahasa Java dan C++. Python sering kali hanya menggunakan beberapa baris kode dalam menyelesaikan pekerjaan, sehingga memungkinkan menggunakan di berbagai lingkungan dari pengembangan web, pengembangan mobile, aplikasi desktop serta perangkat keras pemrograman.

Berikut beberapa *library* Python yang digunakan :

- a. *Scikit-learn*
Scikit-learn yaitu *library* yang dikembangkan untuk mengimplementasikan model algoritma pembelajaran mesin seperti klasifikasi dan klusterisasi.

Scikit-learn dapat berfungsi dengan *library scipy, numpy, pandas* dalam merancang model.

b. *Numpy*

Numpy merupakan struktur data yang mengambil data dalam bentuk array. *Numpy* array memiliki bentuk yang tetap $m \times n$ dengan m merupakan baris dan n yaitu kolom yang sama saat ditambahkan ke dalam matriks. Inisialisasi : `import numpy as np`

c. *Pandas*

Pandas yaitu *library* yang digunakan untuk analisis data yang disebut dengan *DataFrame*. *DataFrame* merupakan sebuah tabel seperti spreadsheet Excel. Setiap kolom pada *pandas* memiliki tipe yang terpisah seperti bilangan *integers, dates, floating-point*, dan *strings*. Berbagai file yang dapat digunakan pada *library pandas* diantaranya SQL, Excel, dan *comma-separated values (CSV)*. Inisialisasi : `import pandas as pd`

2.14 Flask

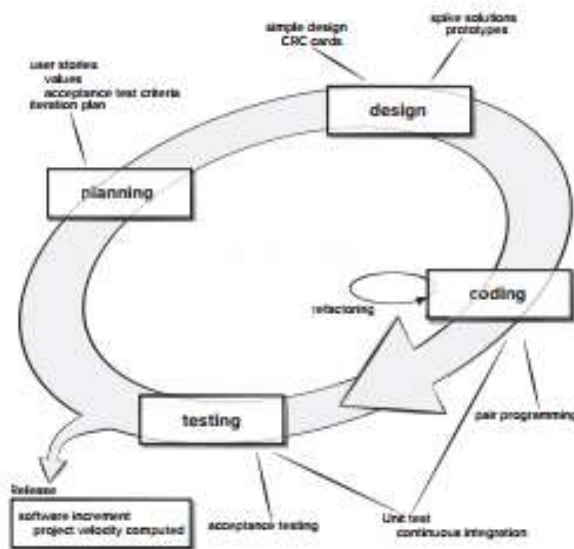
Flask merupakan microweb *framework* yang menggunakan bahasa Python. *Flask* banyak digunakan karena beberapa program dapat berjalan dengan memory yang rendah. *Flask* merupakan *framework* yang ringan tetapi fungsinya dapat berjalan sesuai dengan kebutuhan, serta berintegrasi dengan SQLALchemy sehingga berjalan optimal dengan database (Prehandayana et al., 2018). *Flask* adalah *framework* aplikasi web yang ditulis dengan bahasa Python. *Flask* menyediakan *tools, libraries*, dan teknologi yang dapat bekerja untuk membangun sebuah aplikasi web. Aplikasi web ini seperti halaman web, blog, wiki atau situs web komersial. *Flask* bagian dari kategori *microframework*. Fitur pada *flask* yaitu (Aslam et al., 2019) :

- a. Dukungan terintegrasi dalam pengujian unit
- b. Menggunakan *template* Jinja2
- c. Mendukung *secure cookie*
- d. Dokumentasi yang ekstensif
- e. Kompatibe dengan *Google app*

f. *Restful request dispatching*

2.15 Metode *Extreme Programming*

Extreme Programming salah satu *agile methods* yang paling banyak digunakan untuk pengembangan perangkat lunak. *Extreme Programming* memiliki empat tahap kerangka kegiatan : *planning*, *design*, *coding*, dan *testing* (Pressman, 2020). Berikut ini penjelasan dari tahap-tahap *Extreme Programming* dapat dilihat pada gambar 2.



Gambar 2. Metode *Extreme Programming* (Pressman, 2020).

1. *Planning*

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis sesuai dengan sistem yang akan dibangun dan melakukan wawancara serta diskusi dengan narasumber sehingga mendapatkan cerita dari pengguna yang akan dijadikan acuan dalam tahap selanjutnya.

2. *Design*

Desain sistem dilakukan dengan cara mengembangkan cerita dari pengguna dan merancang alur kerja sistem dengan pemodelan sistem dan pemodelan

arsitektur. Pada pemodelan sistem menggunakan UML dalam bentuk *use case diagram*, *activity diagram* dan *class diagram*. Adapun pemodelan arsitektur menggunakan Balsmiq Mockup untuk membuat rancangan sistem dalam bentuk *mockup* yang meminimalisir kesalahan desain awal sebelum masuk ketahap selanjutnya.

3. *Coding*

Rancangan yang telah dibuat pada tahap sebelumnya akan diterjemahkan ke dalam bahasa pemrograman yang telah ditentukan yaitu pemrograman bahasa Python dan framework Flask.

4. *Testing*

Hasil dari tahap *coding* akan diuji oleh pengguna dengan menggunakan *black-box*. Pengujian dilakukan untuk mengetahui apakah sistem sudah sesuai kebutuhan pengguna atau tidak, pengujian ini berfokus pada fitur dan fungsionalitas pada sistem.

2.16 Balsamiq Mockup

Balsamiq Mockup merupakan suatu alat desain antarmuka perngguna dalam membuat gambar rangka atau disebut prototipe low-fidelity (Sopandi & Rosdiana, 2020). *Balsamiq Mockup* yaitu software yang digunakan untuk membuat *prototyping* atau desain dalam pembuatan *user interface* dari sebuah aplikasi. *Balsamiq Mockup* menyediakan berbagai tools sehingga dapat memudahkan dalam proses pembuatan desain apikasi yang akan dibuat (Kriya, 2020).

2.17 Black-box Testing

Black-box Testing yaitu teknik pengujian perangkat lunak untuk menentukan fungsionalitas dari aplikasi. *Black-box Testing* berfokus pada input dan output tanpa mengetahui internal aplikasi. Metode pengujian ini didasarkan pada spesifikasi dan kebutuhan perangkat lunak (Verma et al., 2017). *Black-box*

Testing dilakukan untuk menemukan hal-hal berikut ini (Mustaqbal et al., 2015) :

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka
3. Kesalahan pada struktur data dan akses basis data
4. Kesalahan performansi
5. Kesalahan inialisasi dan terminasi.

Black-box Testing dibagi menjadi beberapa metode diantaranya *Equivalence partitioning*, *Boundary Value Analysis*, *Cause Effect Graphing*, *Decision Table Based Testing*, dan *Error Guessing*.

2.18 *Equivalence Partitioning*

Equivalence partitioning merupakan teknik pengujian dengan kasus uji yang membutuhkan semua nilai masukan dan membagi menjadi partisi-partisi. Setiap partisi berisi valid atau tidak valid untuk menemukan kesalahan (Verma et al., 2017). Pengujian *equivalence partitioning* merupakan pengujian yang berdasarkan dari masukan setiap menu yang terdapat pada sistem, setiap menu dilakukan pengujian melalui klasifikasi dan pengelompokan berdasarkan fungsinya (Hidayat & Muttaqin, 2018). Contoh pengujian *equivalence partitioning* dapat dilihat pada Tabel 5.

Tabel 5. Pengujian *Equivalence Partitioning* (Setiawan et al., 2020)

Id	Deskripsi Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
P01	Mengisi kolom ID Buku dengan "001", kolom Nis Siswa dengan "123456", dan kolom Tanggal Peminjaman dengan "01/10/2020" kemudian klik tombol Submit.	Sistem berhasil menyimpan penambahan data ke database dan menampilkan perubahan tampilan pada halaman sistem	Sistem menampilkan pop up pesan "Data Sukses Disimpan" serta kolom-kolom pada form tidak terisi atau sudah ter-reset.	Sesuai

2.19 Penelitian Terdahulu

Penelitian ini dilakukan tidak terlepas dari penelitian-penelitian sebelumnya yang berkaitan dengan latar belakang masalah yaitu mengenai implementasi algoritma *k-nearest neighbor*. Penelitian terdahulu dapat dijadikan sebagai acuan dan pertimbangan dalam penelitian yang akan dilakukan. Adapun penelitian terdahulu yang terkait diambil dari berbagai jurnal dapat dilihat pada tabel 6.

Tabel 6. Penelitian Terdahulu

Penelitian Terdahulu					
Judul	Fitur				Akurasi
	1	2	3	4	
Penerapan Metode <i>K-nearest neighbor</i> pada Aplikasi Penentu Penerima Beasiswa Mahasiswa di STMIK Sinar Nusantara Surakarta (Panoto, 2017).	Input Data	Seleksi Pemohon Beasiswa	Hasil Seleksi	Data Sampel	90.90%
Klasifikasi Sekolah Menengah Pertama/ Sederajat Wilayah Bireuen Menggunakan Algoritma <i>K- Nearest Neighbor</i> Berbasis Web (Dinata & Hasdyna, 2020).	Menu Uta-ma	Menu Data	Perhitungan Algoritma K-NN	Hasil Perhitungan Jarak	79.62%
Implementasi Metode <i>K-Nearest Neighbor</i> untuk Rekomendasi Keminatan Studi (Anshori et al., 2018).	Data	Scoring	Hitung	-	76.66%
Algoritma <i>K- Nearest Neighbor Classification</i> Sebagai Sistem Prediksi Predikat Prestasi Mahasiswa (Mustakim & Oktaviani F, 2016).	Home	Data Mahasiswa	Prediksi	-	82%
The Lao Text Classification Method Based on KNN (Zhou et al., 2020)	-	-	-	-	71,4%

Tabel 6. (Lanjutan)

Penelitian Terdahulu					
Judul	Fitur				Akurasi
	1	2	3	4	
Feature Extraction based Text Classification using K-Nearest Neighbor Algorithm (Azam et al., 2018).	-	-	-	-	78,67%
Penelitian yang akan dilakukan					
Judul	Fitur				Akurasi
	1	2	3	4	
Implementasi Algoritma <i>K-nearest neighbor</i> Untuk Sistem Rekomendasi Topik Penelitian Pada Jurusan Ilmu Komputer Berbasis Klasifikasi	Mengelola Data Mahasiswa	Kategori Topik Penelitian	Prediksi		-

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

3.1.1 Tempat Penelitian

Penelitian ini dilaksanakan di Gedung Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Jalan Soemantri Bojonegoro No. 1 Gedung Meneng Bandar Lampung.

3.1.2 Waktu Penelitian

Waktu penelitian dilaksanakan pada Semester tujuh bulan Desember 2020 hingga bulan Agustus 2021. Perencanaan waktu penelitian dimulai pada bulan Desember. Proses awal penelitian dimulai dengan mempelajari studi literatur pada bulan Desember sampai bulan Januari. Selanjutnya proses pengumpulan data dilakukan pada bulan Februari yang diperoleh dari UPT TIK Universitas Lampung. Kemudian dilakukan proses perancangan sistem rekomendasi topik penelitian pada bulan Maret sampai bulan April. Setelah proses perancangan sistem selesai, dilakukan proses penyusunan *draft* bab 1-3 pada bulan Januari sampai bulan April. Pada bulan Juni sampai bulan Juli dilakukan proses pembuatan model *machine learning* dengan algoritma *k-nearest neighbor*. Setelah proses pembuatan model selesai, dilanjutkan proses pembuatan sistem rekomendasi topik penelitian menggunakan *framework flask* pada bulan Agustus sampai September. Kemudian dilakukan proses pengujian sistem menggunakan metode *black-box* pada bulan Oktober. Selanjutnya proses penyusunan *draft* bab 4-5 dilakukan pada bulan Agustus sampai bulan Oktober.

3.1.3 Data

Data yang digunakan dalam penelitian ini merupakan data yang diperoleh dari pengelola server siakadu, UPT TIK Universitas Lampung pada tahun 2016 – 2019. Data yang digunakan berjumlah 240 data dengan atribut terdiri dari NPM mahasiswa, judul penelitian, kategori penelitian serta 35 data matakuliah dari 68 data mahasiswa Jurusan Ilmu Komputer yang telah lulus.

3.1.4 Alat

Alat yang digunakan dalam penelitian ini yaitu

a. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah Smartphone tipe Android dan Laptop Acer Aspire E1-472G dengan spesifikasi sebagai berikut:

1. *Processor* Intel Core i5.
2. RAM 8,00 GB.
3. Hardisk 500 GB.

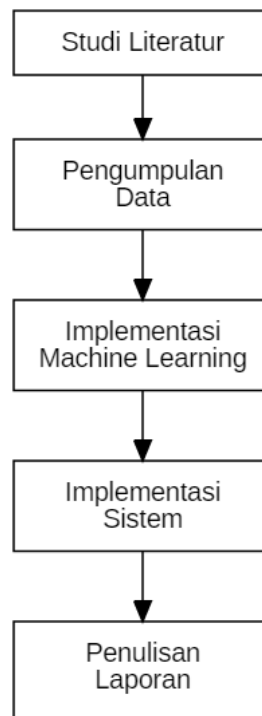
b. Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini sebagai berikut :

1. Sistem Operasi Windows 2010.
2. Xampp sebagai *local server*.
3. Notepad++ sebagai text editor.
4. Chrome sebagai aplikasi untuk menjalankan dan menguji sistem.
5. StarUML sebagai aplikasi untuk pemodelan UML.
6. Balsamiq Mockup3 sebagai aplikasi untuk membuat rancangan sistem.

3.2 Tahapan Penelitian

Tahapan penelitian merupakan cara bagaimana suatu penelitian dapat dilaksanakan. Adapun pada implementasi sistem rekomendasi topik penelitian menggunakan tahap penelitian yang dapat dilihat pada Gambar 3.



Gambar 3. Alur Penelitian

3.2.1 Studi Literatur

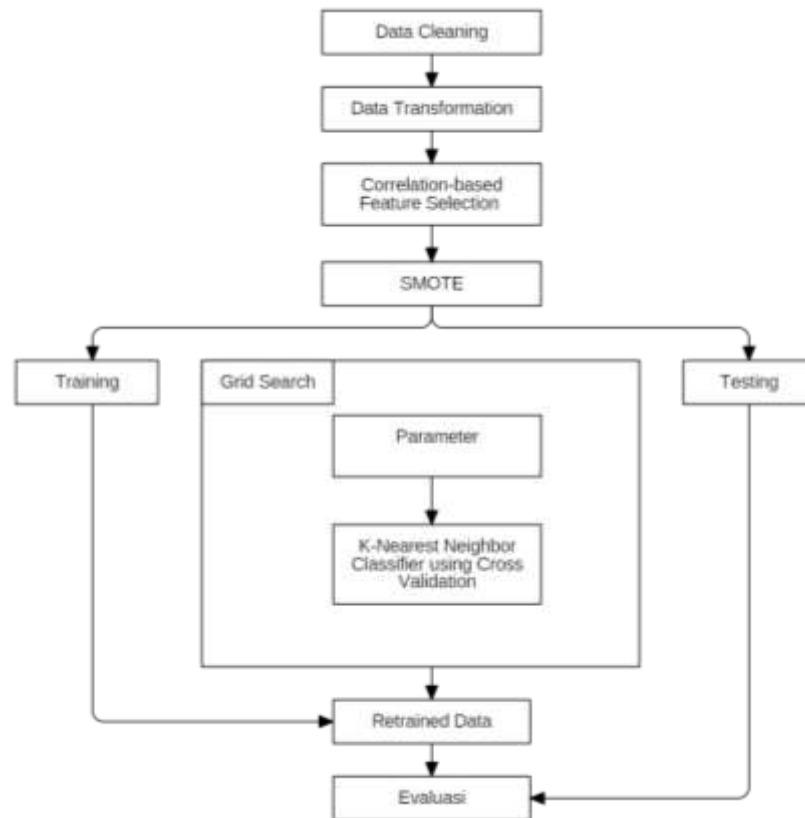
Studi literatur merupakan tahapan awal yang dilakukan dalam penelitian ini dengan mencari jurnal atau buku sebagai referensi maupun acuan dasar yang sesuai dengan permasalahan ini diantaranya sistem rekomendasi, algoritma *K-nearest neighbor*, serta metode *extreme programming*.

3.2.2 Pengumpulan Data

Pengumpulan data merupakan tahap kedua yang dilakukan berasal dari data SIAKAD UPT TIK Universitas Lampung serta berdiskusi dengan Dosen dan Sekretaris Jurusan Ilmu Komputer Universitas Lampung. Data berjumlah 240 yang terdiri dari 35 data independen dan 14 data dependen.

3.2.3 Implementasi *Machine Learning*

Implementasi *machine learning* merupakan tahap ketiga yang dilakukan dalam pembentukan model *machine learning*. Tahapan alur pembentukan model *machine learning* dapat dilihat pada gambar 4.



Gambar 4. Flowchart Model *Machine Learning*.

a. *Data Cleaning*

Data cleaning merupakan proses untuk mengatasi nilai yang hilang atau kosong pada data matakuliah mahasiswa.

b. *Data Transformation*

Feature transformation merupakan proses merubah data awal dari bentuk *string* menjadi bentuk numerik. Proses ini diimplementasikan dalam bentuk *coding*.

c. *Correlation-based Feature Selection*

Correlation-based Feature Selection merupakan pengukuran korelasi untuk menentukan pemilihan variabel antar data atribut yang saling berkorelasi tinggi.

d. *Sythetic Minority Over-Sampling Technique* (SMOTE)

Sythetic Minority Over-Sampling Technique (SMOTE) merupakan proses untuk menyeimbangkan data yang tidak seimbangan. Proses ini bekerja dengan *k-nearest neighbors* untuk mencari tetangga *k* terdekat pada setiap data di kelas minoritas kemudian dibuat data sintesis sebanyak presentase yang diduplikasi antara data minor dengan *k-nearest neighbors* yang dipilih dengan secara acak.

e. Pembagian Data

Pembagian data pada penelitian ini dibagi menjadi data *training* dan data *testing*. Data *training* digunakan untuk pembentukan model sedangkan data *testing* digunakan untuk evaluasi performa dari model tersebut.

f. *Grid Search*

Grid search merupakan teknik yang dilakukan untuk mencari kombinasi parameter yang menghasilkan performa terbaik. *Grid search* menyimpan parameter serta algoritma *K-nearest neighbor* dengan *cross validation*, kemudian *Grid search* akan melakukan perhitungan berdasarkan semua parameter yang tersimpan dan jumlah dari validasi silang untuk mendapatkan parameter yang terbaik.

g. Evaluasi

Evaluasi digunakan pada penelitian ini untuk mengevaluasi kinerja dari algoritma yang digunakan dengan menggunakan *confusion matrix* dan menghasilkan pengukuran akurasi, *precisian*, serta *recall*.

3.2.4 Implementasi Sistem

Implementasi sistem dalam penelitian ini menggunakan metode *extreme programming*. Adapun tahap-tahap dalam pengembangan sistem yaitu :

3.2.4.1 *Planning*

Tahap ini merupakan tahap awal yang dilakukan dengan mengumpulkan kebutuhan-kebutuhan sistem yang akan dibuat seperti fitur apa saja yang mendukung pada sistem rekomendasi ini. Kebutuhan-kebutuhan sistem tersebut ditulis dalam bentuk *user stories*. *User stories* dapat dijelaskan dengan judul, deskripsi dan *acceptance criteria* (Choudhari & Suman, 2012). Berikut penjelasan *user stories* dapat dilihat pada Tabel 7.

Tabel 7. *User Stories*

No.	Judul	Deskripsi Fitur	<i>Acceptance Criteria</i>
1.	Mengelola data mahasiswa	Admin dapat mengakses data mahasiswa dengan menambah, mengedit, dan menghapus data mahasiswa.	Terdapat fitur mengelola data mahasiswa untuk menambah, mengedit dan menghapus data mahasiswa.
2.	Melatih ulang data	Admin dapat melatih ulang data ketika data diperbarui.	Pada tampilan admin terdapat tombol latih ulang data untuk melatih ulang data <i>pickle</i> ketika data mahasiswa di perbarui
3.	Melakukan prediksi	Pengguna dapat memprediksi kategori topik penelitian.	Pada tampilan pengguna terdapat fitur prediksi untuk memprediksi topik penelitian berdasarkan nilai matakuliah.
4.	Melihat Kategori	Pengguna dapat melihat judul skripsi dari setiap kategori topik penelitian	Terdapat fitur kategori untuk melihat judul skripsi dari setiap kategori topik penelitian.

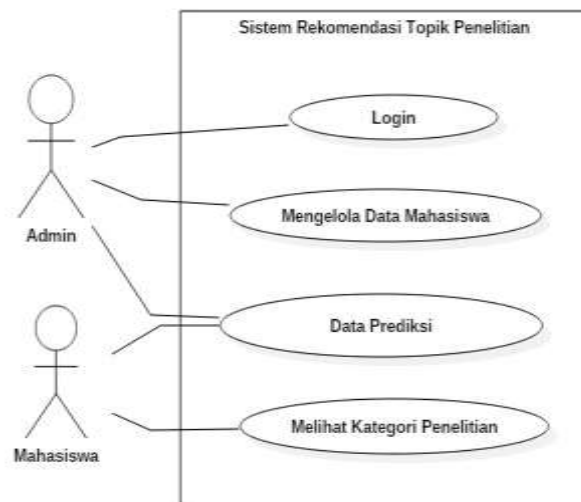
3.2.4.2 *Design*

Tahap kedua yaitu *design* atau tahap perancangan sistem, dilakukan dengan merancang tampilan sistem yang

diilustrasikan dengan *use case diagram*, *activity diagram*, dan *interface sistem*.

a. *Use case Diagram*

Use case Diagram pada Sistem Rekomendasi Topik Penelitian dirancang untuk mendeskripsikan interaksi antara admin dan mahasiswa serta untuk mengetahui fungsi *interface* dari masing-masing pengguna yaitu admin dan mahasiswa. *Use case Diagram* Sistem Rekomendasi Topik Penelitian dapat dilihat pada gambar 5.



Gambar 5. *Use Case Diagram* Sistem Rekomendasi Topik Penelitian.

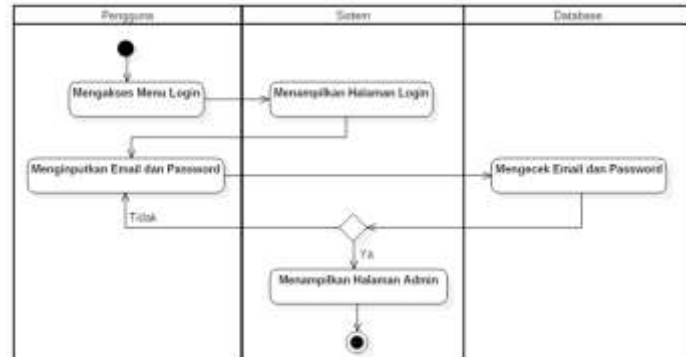
b. *Activity Diagram*

Activity Diagram digunakan untuk mendeskripsikan urutan aktivitas yang dibentuk dalam suatu operasi serta aktivitas lainnya. Pada Sistem Rekomendasi Penelitian terdapat 6 *activity diagram* yaitu sebagai berikut.

1. *Activity Diagram Login*

Pada *Activity Diagram Login* hanya admin yang dapat mengakses atau masuk untuk ke menu selanjutnya. Admin dapat memasukkan *username* serta password

pada *form* login. *Activity diagram login* dapat dilihat pada gambar 6.



Gambar 6. *Activity Diagram Login.*

2. *Activity Diagram* Mengelola Data Mahasiswa

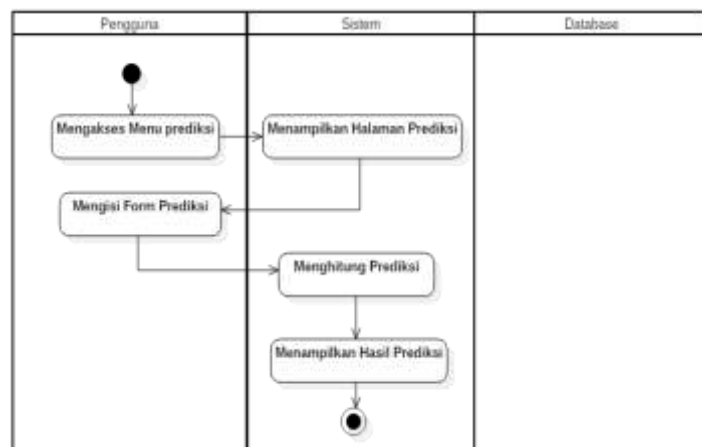
Pada *Activity Diagram* Mengelola Data Mahasiswa menggambarkan alur aktifitas yang menampilkan data mahasiswa. *Activity Diagram* Mengelola Data Mahasiswa dapat dilihat pada Gambar 7.



Gambar 7. Activity Diagram Mengelola Data Mahasiswa.

3. Activity Diagram Prediksi

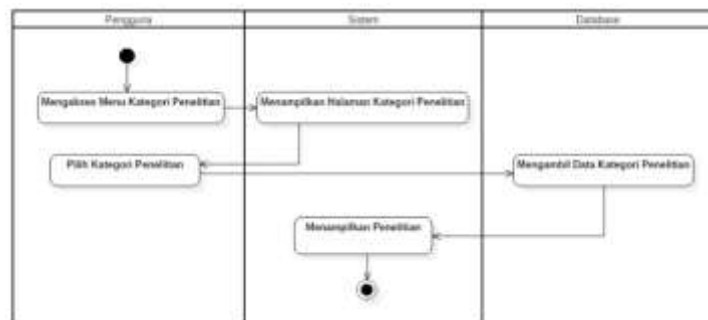
Activity Diagram Prediksi menggambarkan alur aktifitas untuk memprediksi topik penelitian dengan mengisi form yang ditampilkan. Activity Diagram Prediksi dapat dilihat pada Gambar 8.



Gambar 8. Activity Diagram Prediksi.

4. Activity Diagram Melihat Kategori Penelitian

Activity Diagram Melihat Kategori Penelitian menggambarkan alur aktifitas pada mahasiswa untuk melihat judul penelitian dari setiap kategori penelitian. *Activity Diagram* Melihat Kategori Penelitian dapat dilihat pada Gambar 9.



Gambar 9. Activity Diagram Kategori Penelitian

c. Class Responsibility Collaborator Cards (CRC Cards)

CRC Cards digunakan untuk menunjukkan interaksi antar objek serta fungsi yang dibutuhkan dan hubungannya dengan kelas lain. *CRC Cards* dapat dilihat pada Gambar 10.



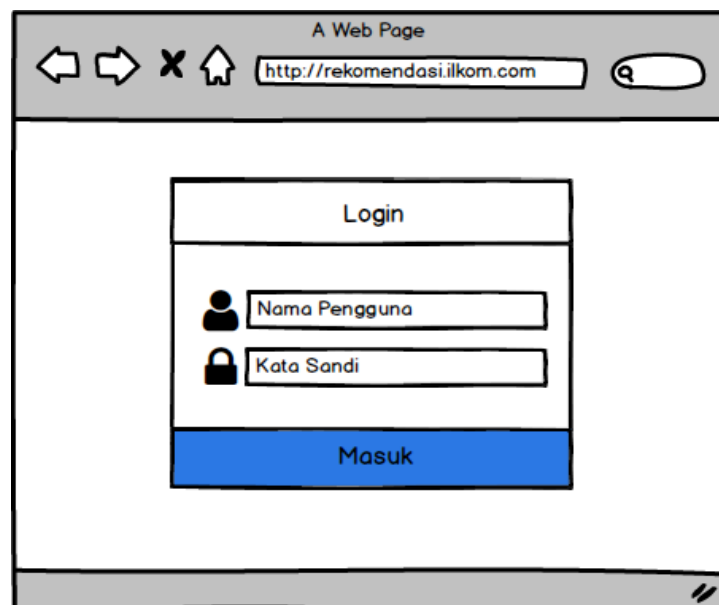
Gambar 10. Class Responsibility Collaborator Cards

d. Desain Antarmuka atau *Interface*

Desain *interface* dilakukan untuk merancang tata letak sistem sesuai dengan kebutuhan sistem. *Interface* yang dirancang yaitu sebagai berikut.

1. Rancangan Tampilan Halaman Login

Tampilan halaman login merupakan halaman khusus admin untuk mengelola data mahasiswa. Pada halaman login admin diminta untuk memasukkan *email* dan *password*. Rancangan tampilan login ditunjukkan pada gambar 10.

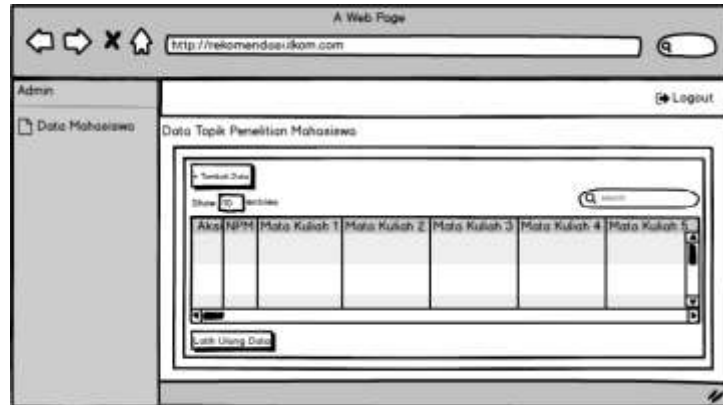


Gambar 11. Rancangan Tampilan *Login*.

2. Rancangan Tampilan Halaman Mengelola Data Mahasiswa

Tampilan halaman data mahasiswa berisikan data mahasiswa Jurusan Ilmu Komputer yang sudah lulus.

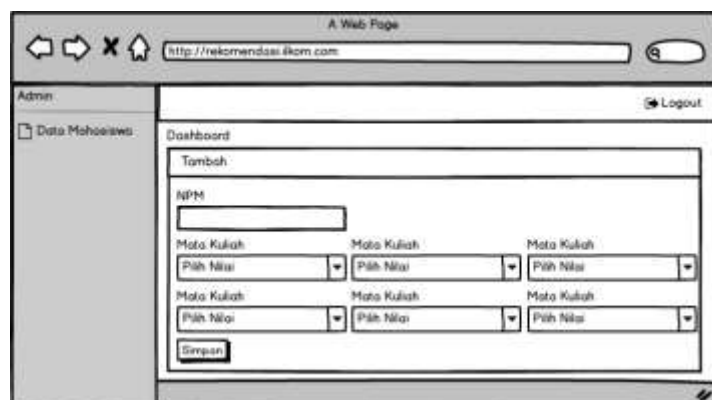
Pada halaman ini admin dapat melihat fitur tambah dan edit data mahasiswa serta dapat mengulang data latihan atau data *training*. Rancangan tampilan data mahasiswa ditunjukkan pada Gambar 11.



Gambar 12. Rancangan Tampilan Data Mahasiswa.

3. Rancangan Tampilan Halaman Tambah Data Mahasiswa

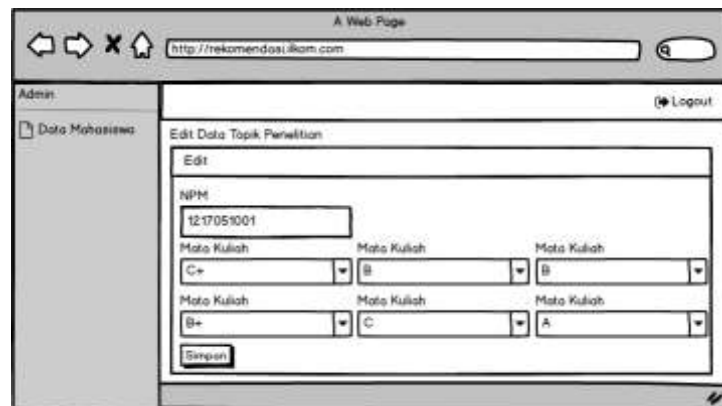
Tampilan halaman tambah data berisi *form* yang akan diisi oleh admin untuk menambahkan data mahasiswa. Rancangan tampilan tambah data mahasiswa dapat dilihat pada Gambar 13.



Gambar 13. Rancangan Tampilan Tambah Data

4. Rancangan Tampilan Halaman Edit

Tampilan halaman edit menampilkan *form* edit data yang sudah diisi sebelumnya. Admin dapat mengedit data mahasiswa jika ada kesalahan dan kekeliruan saat menambahkan data. Rancangan tampilan edit data mahasiswa dapat dilihat pada Gambar 14.



The screenshot shows a web browser window with the URL <http://rekomendasi.ikom.com>. The page title is "A Web Page". The user is logged in as "Admin" and has a "Logout" link. The main content area is titled "Edit Data Topik Penelitian" and contains a form with the following fields:

- NPM: 1217051001
- Mata Kuliah 1: C+
- Mata Kuliah 2: B
- Mata Kuliah 3: B
- Mata Kuliah 4: B+
- Mata Kuliah 5: C
- Mata Kuliah 6: A

A "Simpan" button is located at the bottom of the form.

Gambar 14. Rancangan Tampilan Edit Data

5. Rancangan Tampilan Halaman Beranda

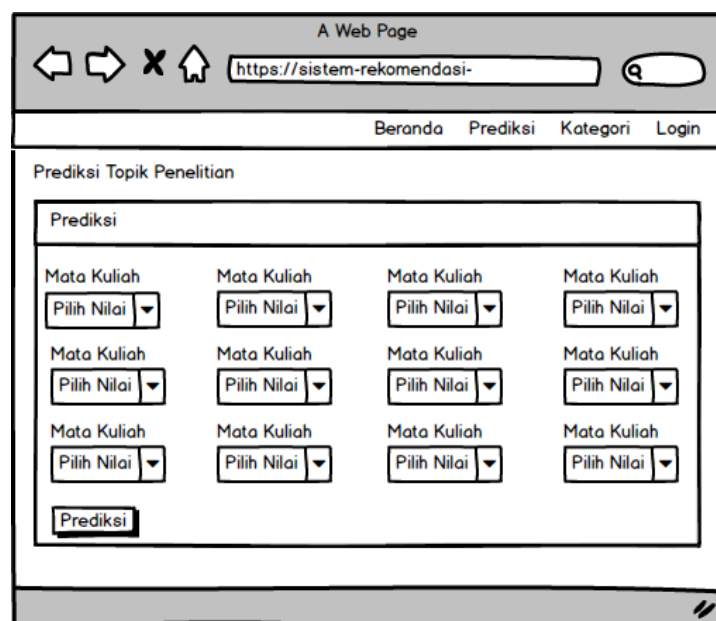
Tampilan halaman beranda menampilkan gambar dari gedung Jurusan Ilmu Komputer Universitas Lampung. Rancangan tampilan halaman beranda dilihat pada Gambar 15.



Gambar 15. Rancangan Tampilan Beranda

6. Rancangan Tampilan Halaman Prediksi

Tampilan halaman prediksi berisi *form* yang akan diisi oleh pengguna atau mahasiswa untuk memprediksi mengenai topik penelitian. Pengguna dapat mengisi mata kuliah yang sudah ditentukan pada *form* dan selanjutnya dapat pilih tombol prediksi untuk mengetahui hasil dari prediksi tersebut. Rancangan tampilan prediksi dapat dilihat pada gambar 16.

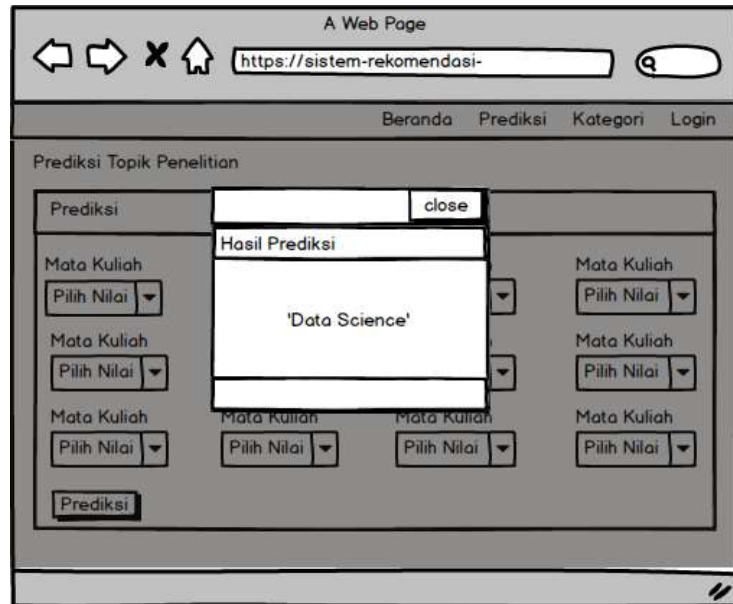


The image shows a web browser window with the title "A Web Page". The address bar contains "https://sistem-rekomendasi-". The page has a navigation menu with "Beranda", "Prediksi", "Kategori", and "Login". The main content area is titled "Prediksi Topik Penelitian" and contains a form with 12 dropdown menus labeled "Mata Kuliah" and "Pilih Nilai", and a "Prediksi" button.

Gambar 16. Rancangan Tampilan Tambah Prediksi

7. Tampilan Halaman Hasil Prediksi

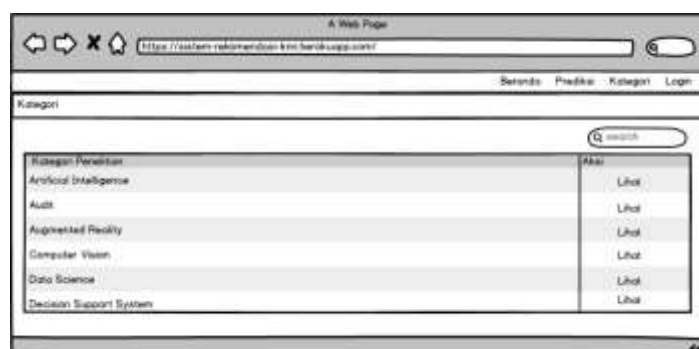
Tampilan halaman hasil prediksi menampilkan kategori topik penelitian yang sudah dihitung oleh sistem. Rancangan tampilan hasil prediksi dapat dilihat pada gambar 17.



Gambar 17. Rancangan Tampilan Hasil Prediksi

8. Tampilan Halaman Kategori Penelitian

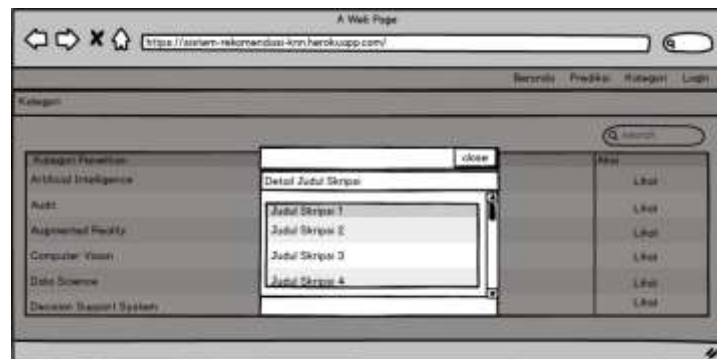
Tampilan halaman kategori penelitian menampilkan semua kategori penelitian, mahasiswa dapat memilih satu kategori penelitian untuk dilihat judul pebelitian dari kategori tersebut dengan mengklik tombol detail. Rancangan tampilan kategori penelitian dapat dilihat pada gambar 18.



Gambar 18. Rancangan Tampilan Kategori Penelitian

9. Tampilan Halaman Lihat Kategori

Tampilan halaman lihat kategori penelitian menampilkan semua judul penelitian dari setiap kategori yang dipilih. Rancangan tampilan lihat kategori penelitian dapat dilihat pada gambar 19.



Gambar 19. Rancangan Tampilan Lihat Kategori Penelitian

3.2.4.3 Coding

Tahap selanjutnya yaitu tahap *coding* yang dilakukan dengan membuat kode program menggunakan bahasa pemrograman Python dan *framework flask*. Pada *extreme programming* terdapat istilah *pair programming* yaitu proses penulisan program dilakukan secara berpasangan, dua orang programmer bekerjasama dalam menuliskan program (Juliyanto et al., 2020).

3.2.4.4 Testing

Tahap terakhir yaitu pengujian sistem menggunakan *Black-box* yang dilakukan dengan menguji fungsional dari setiap fitur. *Black-box testing* bekerja dengan cara mencari kesalahan fungsi dari sistem sehingga mengurangi tingkat kesalahan

pada nilai masukan. Skenario pengujian menggunakan *Black-box Testing* dapat dilihat pada Tabel 8.

Tabel 8. Skenario Pengujian

No.	Kelas Uji	Kasus Uji	Hasil yang Diharapkan
1	Mengakses halaman data mahasiswa	Mengakses halaman data dengan klik menu data mahasiswa.	Sistem dapat menampilkan data mahasiswa, <i>icon</i> edit, dan <i>icon</i> hapus.
2	Menambah data mahasiswa	Mengakses halaman tambah data dengan klik menu tambah.	Sistem dapat menampilkan halaman tambah data mahasiswa yang berisi <i>form</i> dan tombol simpan.
		Melakukan penambahan data:	Sistem dapat berhasil menyimpan data dengan menampilkan <i>popup</i> konfirmasi "Data Berhasil Ditambahkan" dan mengarahkan Admin ke halaman data mahasiswa.
		a. Mengisi seluruh <i>field</i> sesuai dengan format <i>input</i> .	
		b. Mengisi NPM dengan 10 digit angka.	
		c. Memilih nilai mata kuliah.	
		d. Mengisi judul penelitian.	
		e. Mengisi kategori penelitian	
		Mengisi <i>form</i> dengan <i>field</i> kosong kemudian klik simpan.	Sistem dapat menampilkan pesan error "Data Tidak Boleh Kosong".
		Mengisi NPM dengan huruf atau karakter selain angka.	Sistem dapat menampilkan pesan error, "Data Tidak Boleh Kosong dan Harus 10 Angka".
3	Mengubah data mahasiswa	Mengakses halaman edit data dengan klik tombol edit	Sistem dapat menampilkan halaman edit data mahasiswa yang dipilih.

Tabel 8. (Lanjutan)

No.	Kelas Uji	Kasus Uji	Hasil yang Diharapkan
		Mengubah data kemudian klik simpan dengan: <ol style="list-style-type: none"> Mengubah seluruh <i>field</i> input. Mengubah judul penelitian. Mengubah Kategori penelitian. 	Sistem dapat berhasil menyimpan data yang diubah dengan menampilkan <i>popup</i> konfirmasi “Data Berhasil Diubah” dan mengarahkan admin ke halaman data mahasiswa.
		Mengubah data dengan <i>field</i> kosong kemudian klik simpan.	Sistem dapat menampilkan pesan error “Data Tidak Boleh Kosong”.
4	Menghapus data pada halaman data mahasiswa	Menampilkan <i>pop up</i> notifikasi hapus dengan klik <i>icon</i> hapus.	Sistem dapat menampilkan <i>pop up</i> konfirmasi hapus data.
		Menghapus data dengan klik ok pada <i>pop up</i> .	Data dapat dihapus dan mengarahkan admin ke halaman data mahasiswa.
5	Mengakses halaman <i>login</i>	Mengakses halaman <i>login</i> dengan klik <i>login</i> .	Sistem dapat menampilkan <i>form</i> halaman <i>login</i> dan tombol masuk.
		Mengisi <i>field</i> nama pengguna dan kata sandi kemudian klik masuk.	Sistem dapat berhasil masuk dengan mengarahkan admin ke halaman data mahasiswa.
		Mengisi <i>form</i> dengan <i>field</i> kosong.	Sistem dapat menampilkan pesan error data tidak boleh kosong.
6	Mengakses halaman prediksi.	Mengakses halaman dengan klik prediksi.	Sistem dapat menampilkan halaman prediksi yang berisi <i>form</i> dan tombol prediksi.
		Mengisi <i>form</i> dengan memilih nilai matakuliah kemudian klik prediksi.	Sistem dapat menampilkan <i>pop up</i> hasil dari prediksi.

Tabel 8. (Lanjutan)

No.	Kelas Uji	Kasus Uji	Hasil yang Diharapkan
		Mengisi <i>form</i> dengan <i>field</i> kosong atau tidak memilih nilai matakuliah kemudian klik prediksi.	Sistem dapat menampilkan pesan error data tidak boleh kosong.
7	Mengakses halaman kategori.	Mengakses halaman dengan klik kategori.	Sistem dapat menampilkan data kategori.
		Mengakses tombol detail pada kategori yang dipilih.	Sistem dapat menampilkan <i>pop up</i> judul dari kategori yang dipilih.

3.2.5 Penulisan Laporan

Tahap terakhir yaitu penulisan laporan yang dilakukan apabila semua tahapan penelitian dan fungsi sistem sudah berjalan dengan baik. Tahap ini juga dilakukan pembahasan dari hasil data yang diperoleh dengan implementasi, pengujian serta mengambil kesimpulan dari penelitian yang telah dilakukan.

V. SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan penelitian yang telah dilakukan, maka hasil yang telah dicapai dapat disimpulkan sebagai berikut :

1. Analisis SMOTE (*Systhetic Minority Over-Sampling Technique*) untuk ketidakseimbangan data pada penelitian ini dengan membandingkan jarak k satu sampai empat didapatkan hasil terbaik yaitu jarak k tiga dengan akurasi sebesar 82%. Jumlah data sebelum dilakukan SMOTE yaitu 240 data dengan kelas terkecil yaitu *artificial intelligence* sebanyak 5 data dan kelas terbesar yaitu sistem informasi sebanyak 50 data. Jumlah data setelah dilakukan SMOTE yaitu 742 data dengan jumlah masing-masing kelas sebanyak 53 data.
2. Berdasarkan pencarian model terbaik dengan menggunakan tetangga terdekat 3, 5, 7, 9 dan jarak (*distance*) menggunakan *euclidean distance*, *minkowski distance*, *manhattan distance*, dan *chebyshev distance* pada penelitian ini didapatkan hasil terbaik dengan tetangga terdekat yaitu 3 dan jarak (*distance*) yaitu *manhattan distance*.
3. Hasil klasifikasi menggunakan *K-nearest neighbor* (KNN) pada penelitian ini didapatkan hasil rata-rata *accuracy* sebesar 82% , dengan hasil rata-rata *precision* sebesar 82% dan hasil rata-rata *recall* sebesar 82%.
4. Sistem rekomendasi yang telah dibuat dengan menggunakan algoritma K-Naerest Neighbor menghasilkan prediksi kategori topik penelitian dan setiap kategori terdiri dari kumpulan judul skripsi mahasiswa yang telah lulus.

5. Sistem rekomendasi topik penelitian ini dapat membantu mahasiswa untuk menentukan topik penelitian skripsi berdasarkan 34 nilai mata kuliah yang sudah ditentukan.

5.2 Saran

Berdasarkan hasil implementasi pada *machine learning* dan implementasi pada sistem yang telah dilakukan diperoleh beberapa saran yang diberikan yaitu sebagai berikut :

1. Perlunya penambahan jumlah data pada kategori yang tidak seimbang untuk mendapatkan nilai akurasi yang lebih baik.
2. Membandingkan algoritma *K-nearest neighbor* dengan algoritma lainnya untuk mengetahui hasil akurasi pada penelitian ini.
3. Mengganti nilai *missing value* dengan nilai rata-rata, perhitungan analisis dengan *p-value*, serta melihat matakuliah yang *trend* di Jurusan Ilmu Komputer.
4. Memperbaiki tampilan sistem yang kurang *user friendly* seperti *login*, latihan ulang data dan tampilan prediksi agar sistem lebih mudah dipahami oleh mahasiswa.
5. Memperbaiki tampilan prediksi pada sistem dengan menampilkan judul dari setiap kategori dan menambahkan presentase hasil prediksi dan riwayat hasil prediksi.
6. Membangun sistem rekomendasi lebih lanjut dengan aplikasi android.

DAFTAR PUSTAKA

- Anshori, L., Putri, R. R. M., & Tibyani. (2018). Implementasi Metode *K-nearest neighbor* untuk Rekomendasi Keminatan Implementasi Metode *K-nearest neighbor* untuk Rekomendasi Keminatan Studi (Studi Kasus : Jurusan Teknik Informatika Universitas Brawijaya). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(July).
- Aslam, F. A., Mohammed, H. N., Munir, J. M. M., & Gulamgaus, M. A. (2019). Efficient Way of Web Development Using Python And Flask. *International Journal of Advanced Research in ComputerScience*, 6(2), 54–57. <https://core.ac.uk/download/pdf/55305148.pdf>
- Azam, M., Ahmed, T., Sabah, F., & Hussain, M. I. (2018). Feature Extraction based Text Classification using *K-nearest neighbor* Algorithm. *IJCSNS International Journal of Computer Science and Network Security*, 18(12), 95–101. http://paper.ijcsns.org/07_book/201812/20181213.pdf
- Brownlee, J. (2020). Data Preparation for Machine Learning. In *Ambiguous Childhoods : Peer Socialisation, Schooling and Agency in a Zambian Village* (v1.1).
- Choudhari, J., & Suman, U. (2012). Designing RC story for software maintenance and evolution. *Journal of Software*, 7(5), 1103–1108. <https://doi.org/10.4304/jsw.7.5.1103-1108>
- Dinata, R. K., & Hasdyna, N. (2020). WILAYAH BIREUEN MENGGUNAKAN ALGORITMA *K-NEAREST NEIGHBORS* BERBASIS WEB. *Journal of Computer Engineering System and Science*, 5(1), 33–37.
- Doshi, M., & Chaturvedi, S. K. (2014). Correlation Based Feature Selection (CFS) Technique to Predict Student Perfomance. *International Journal of Computer Networks & Communications*, 6(3), 197–206. <https://doi.org/10.5121/ijcnc.2014.6315>
- Eliana, M., Nuryani, N., & Nugroho, A. S. (2019). Deteksi Fibrilasi Atrium Menggunakan FAM yang Dikombinasikan dengan Grid Search. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 8(2), 175.

<https://doi.org/10.22146/jnteti.v8i2.507>

- Gorade, S. M., Deo, P. A., & Purohit, P. P. (2017). Classifying into a class. *International Research Journal of Engineering and Technology (IRJET)*, 04(04), 3112–3115.
- Hadikristanto, W., & Pungkas, T. D. (2019). *Klasifikasi Status Gizi Orang Dewasa Menggunakan Algoritma Naive Bayes (Studi Kasus Klinik Bhakti Mulia Cikarang)*. 9, 46–53.
- Hall, M. A. (1999). *Correlation-based Feature Selection for Machine Learning* (Issue April).
- Han, J., Kamber, M., & Pei, J. (2014). Data mining: Data mining concepts and techniques. In *Proceedings - 2013 International Conference on Machine Intelligence Research and Advancement, ICMIRA 2013*. <https://doi.org/10.1109/ICMIRA.2013.45>
- Hidayat, T., & Muttaqin, M. (2018). Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis. *Jurnal Teknik Informatika UNIS JUTIS*, 6(1), 2252–5351. www.ccsenet.org/cis
- Hidayatullah, A. F., Fadila, A. A., Juwairi, K. P., & Nayoan, R. A. (2019). Identifikasi Konten Kasar Pada Tweet Bahasa Indonesia. *Jurnal Linguistik Komputasional*, 2(1), 1–5. <https://doi.org/10.26418/jlk.v2i1.15>
- Indrayanti, Sugianti, D., & Karomi, M. A. Al. (2017). OPTIMASI PARAMETER K PADA ALGORITMA K-NEAREST NEIGHBOUR UNTUK KLASIFIKASI PENYAKIT DIABETES MELLITUS. *Seminar Nasional Teknologi Dan Informatif*, 823–829.
- Juliyanto, T., Rusman, A., & Ramanda, K. (2020). Model Global Extreme Programming Pada Sistem Informasi Akademik MTs Al Muddatsiriyah. *Paradigma - Jurnal Komputer Dan Informatika*, 22(1), 1–6. <https://doi.org/10.31294/p.v22i1.7022>
- Kantardzic, M. (2020). Data Mining: Concepts, Models, Methods, and Algorithms. In *IIE Transactions* (Vol. 36, Issue 5). <https://doi.org/10.1080/07408170490426107>
- Kartika, J. I., Santoso, E., & Sutrisno. (2017). Penentuan Siswa Berprestasi Menggunakan Metode *K-nearest neighbor* dan Weighted Product (Studi

Kasus: SMP Negeri 3 Mejayan). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 1(5), 352–360.

Kriya, H. S. (2020). *Jurnal Ilmiah Manajemen Informatika*. 12(2).

Mustakim, & Oktaviani F, G. (2016). Algoritma *K-nearest neighbor* Classification Sebagai Sistem Prediksi Predikat Prestasi Mahasiswa. *Jurnal Sains, Teknologi Dan Industri*, 13(2), 195–202.

Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN). *Jurnal Ilmiah Teknologi Informasi Terapan*, 1(3), 31–36.

Narke, L., & Nasreen, A. (2020). A Comprehensive Review of Approaches and Challenges of a Recommendation System. *International Journal of Research in Engineering, Science and Management*, 3(4), 4–7.

Panoto, A. (2017). Penerapan Algoritma *K-nearest neighbors* untuk prediksi kelulusan Mahasiswa pada STMIK Sinar Nusantara Surakarta. *TIKomSiN*, 27–31.

Prehandayana, G., Yahya, W., & Nurwasito, H. (2018). Implementasi Struktur Data Dictionary Untuk Sistem Monitoring Perangkat Internet of Things. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIHK) Universitas Brawijaya*, 2(10), 3466–3473.

Pressman, R. S. B. R. M. (2020). *Software engineering is software engineering* (NINTH). McGraw-Hill Education, 2 Penn Plaza, New York, NY 10121. <https://doi.org/10.1049/ic:20040411>

Putra, D. W. T., & Andriani, R. (2019). Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD. *Jurnal Teknolf*, 7(1), 32. <https://doi.org/10.21063/jtif.2019.v7.1.32-39>

Rahman, M. F., Alamsah, D., Darmawidjadja, M. I., & Nurma, I. (2017). Klasifikasi Untuk Diagnosa Diabetes Menggunakan Metode Bayesian Regularization Neural Network (RBNN). *Jurnal Informatika*, 11(1), 36. <https://doi.org/10.26555/jifo.v11i1.a5452>

Rahmawati, D., I, M. P. P., Ulum, M., & Ulum, M. (2021). *Identification and Classification of Pathogenic Bacteria Using the K-nearest neighbor Method*

Identifikasi Dan Klasifikasi Bakteri Patogen Dengan. 5(1).

- Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2016). System Analysis and Design in a Changing World. In *In Course Technology Cengage Learning*.
- Setiawan, D., Fadhillah, M. A., Wibawa, A., Sugiarto, I., Mulyana, A., & Kusyadi, I. (2020). Pengujian Black Box pada Aplikasi Perpustakaan Berbasis Web Menggunakan Teknik Equivalence Partitioning. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 3(2), 95. <https://doi.org/10.32493/jtsi.v3i2.3955>
- Shuai, Y., Zheng, Y., & Huang, H. (2019). Hybrid Software Obsolescence Evaluation Model Based on PCA-SVM-GridSearchCV. *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, 2018-Novem*, 449–453. <https://doi.org/10.1109/ICSESS.2018.8663753>
- Sidoarjo, M., Sidoarjo, J. M., & Timur, J. (2017). DOI Link : <https://doi.org/10.21070/jicte.v1i1.1127> Original Research Article Analisis Korelasi untuk Mengetahui Keeratan Hubungan antara Keaktifan Mahasiswa dengan Hasil Belajar Akhir Cindy Cahyaning Astuti Dosen S1 Pendidikan TIK , Fakultas Keguruan dan. *Journal of Information and Computer Technology Education*, 1(April), 1–7.
- Siringoringo, R. (2018). Klasifikasi Data Tidak Seimbang Menggunakan Algoritma SMOTE dan *k-nearest neighbor*. *Jurnal ISD*, 3(1), 44–49.
- Sopandi, A., & Rosdiana, E. (2020). Rancang Bangun Sistem Informasi Serikat Pekerja Seluruh Indonesia (SPSI) Pada PT . Panarub Industry Berbasis Android. *Jurnal Informatika*, 1(2), 73–81.
- Sugiono. (2016). *Metode Penelitian Kuantitatif, Kualitatif dan R & D*. Alfabeta.
- Syafira, A. R. (2017). Sistem Deteksi Wajah Dengan Modifikasi Metode Viola Jones. *Emitor: Jurnal Teknik Elektro*, 17(1), 26–33. <https://doi.org/10.23917/emitor.v17i1.5964>
- Syahrudin, A. N., & Kurniawan, T. (2018). Input Dan Output Pada Bahasa. *Jurnal Dasar Pemrograman Python STMIK, January*, 1–7.
- Verma, A., Khatana, A., & Chaudhary, S. (2017). A Comparative Study of Black Box Testing and White Box Testing. *International Journal of Computer Sciences and Engineering*, 5(12), 301–304.

<https://doi.org/10.26438/ijcse/v5i12.301304>

- Vinothraj, T., Fareez, M., & Sayeth Saabith, A. . (2017). Python Current Trend Applications- an Overview. *International Journal of Advance Engineering and Research Development*, 2(2), 61–67.
- Wahyono, W., Trisna, I. N. P., Sariwening, S. L., Fajar, M., & Wijayanto, D. (2020). Comparison of distance measurement on k-nearest neighbour in textual data classification. *Jurnal Teknologi Dan Sistem Komputer*, 8(1), 54–58. <https://doi.org/10.14710/jtsiskom.8.1.2020.54-58>
- Wati, E. F., & Kusumo, A. A. (2016). Penerapan metode unified modeling language (UML) berbasis desktop pada sistem pengolahan kas kecil studi kasus pada PT indo mada yasa tangerang. *Jurnal Informatika*, 5(1), 24–36. <https://journal.unsika.ac.id/index.php/syntax/article/view/699>
- Xue, P., Zhou, Z., Fang, X., Chen, X., Liu, L., Liu, Y., & Liu, J. (2017). Fault detection and operation optimization in district heating substations based on data mining techniques. *Applied Energy*, 205(73), 926–940. <https://doi.org/10.1016/j.apenergy.2017.08.035>
- Zhou, L. J., Li, X. Da, Zhang, J. N., Huo, W. J., & Chen, Z. (2020). ScienceDirect The Lao Text Classification Method Based on KNN. *Procedia Computer Science*, 166, 523–528. <https://doi.org/10.1016/j.procs.2020.02.053>