

**KLASIFIKASI SISWA BERPRESTASI DENGAN
MENGUNAKAN METODE
K-NEAREST NEIGHBOR
(KNN)**

(Studi Kasus di Smp Negeri 1 Kwandang)

Oleh

MUHAMAD REYNALDI

T3116310

SKRIPSI



**PROGRAM SARJANA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2020**

PENGESAHAN SKRIPSI
KLASIFIKASI SISWA BERPRESTASI DENGAN
MENGGUNAKAN METODE
K-NEAREST NEIGHBOR
(KNN)

(Studi Kasus Smp Negeri 1 Kwandang)

Oleh

MUHAMAD REYNALDI

T3116310

SKRIPSI

Untuk memenuhi salah satu syarat ujian

Guna memperoleh gelar sarjana

Program Studi Teknik Informatika

Ini telah disetujui oleh tim pembimbing

Gorontalo, 3 Desember 2020

Pembimbing Utama

for



Zohrahayatv, S.Kom, M.Kom

NIDN : 0908017702

Pembimbing Pendamping



Sumarni, S.Kom, M.Kom

NIDN : 0926018604

PERSETUJUAN SKRIPSI

**KLASIFIKASI SISWA BERPRESTASI DENGAN
MENGUNAKAN METODE
K-NEAREST NEIGHBOR
(KNN)**

(Studi Kasus Smp Negeri 1 Kwandang)

Oleh

MUHAMAD REYNALDI

T3116310

SKRIPSI

Skripsi ini telah memenuhi syarat yang disetujui oleh
Pembimbing pada tanggal 09 Desember 2020

1. Ketua Penguji

Sudirman S. Panna, M.Kom

2. Anggota

Irvan A Salihi, M.Kom

3. Anggota

Suhardi Rustam, M.Kom

4. Anggota

Zohrahayaty, M.Kom

5. Anggota

Sumarni, M.Kom

Dekan Fakultas Ilmu Komputer

Ketua Program Studi

Zohrahayaty, M.Kom

NIDN : 0912117702

Irvan Abraham Salihi M.Kom

NIDN : 0928028101

PERNYATAAN SKRIPSI

Dengan Ini Saya Menyatakan Bahwa :

1. Karya tulis (skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun diperguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain kecuali, arahan tim pembimbing.
3. Dalam karya tulis (Skripsii) saya ini tidak terdapat karya atau pendapat yang telah dipublikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan/situasi dalam naskah dan dicantumkan pula daftar pustaka.
4. Saya buat dengan sesungguhnya dan apabila dikemudian hari ter. Penyertaan ini saya dapat penyimpangan dan ketidak benaran dalam pernyataan ini, maka saya bersedia menerima sangsi lainnya sesuai norma-norma yang berlaku di Universitas Icshan Gorontalo.

Gorontalo, 03 Desember 2020

Dalam Membuat Pernyataan.



Muhamad Reynaldi

ABSTRAK

Prestasi adalah hasil atas usaha yang dilakukan seseorang yang diperoleh dari sekolah, universitas, perusahaan dan lembaga lain yang salah satunya yaitu Smp Negeri 1 Kwandang. Selama ini proses penyeleksian siswa berprestasi dilakukan dengan mengimput data siswa satu-persatu dalam excel. Pengolahan data yang dilakukan secara komputerisasi drngan memanfaatkan excel yang dilakukan oleh staf tersebut dapat meningkatkan terjadinya kesalahan, seperti terdapat data yang sama atau duplikasi data. Hal tersebut dapat mempengaruhi hasil klasifikasi siswa berprestasi. Metode K-Nearest Naeghbor (KNN) merupakan sebuah metode yang mengklasifikasikan data dan memiliki konsistensiyang kuat dengan melakukan klasifikasi berdasarkan menghitung kedekatan jarak antara kasus baru dan kasus lama sistem menghasilkan hasil penentuan siswa berprestasi berdasarkan kriteria yang telah ditentukan, yaitu nilai raport siswa. Dapat diambil kesimpulan bahwa penerapan metode k-Nearest Naeghbor (KNN) untuk menentukan siswa berprestasi memberikan hasil yang diharapkan serta dapat membantu mempermudah kinerja guru.

Kata Kunci : Klasifikasi Siswa Berprestasi Dengan Metode *K-neares Naeghbor*

ABSTRACT

Achievement is the result of efforts made by a person obtained from schools, universities, companies and other institutions, one of which is SMP Negeri 1 Kwandang. So far, the process of selecting outstanding students is carried out by collecting student data one by one in excel. Computerized data processing using excel performed by the staff can increase the occurrence of errors, such as the same data or duplication of data. This can affect the results of the classification of outstanding students. The K-Nearest Naeghbor (KNN) method is a method that classifies data and has strong consistency by classifying based on calculating the proximity of the distance between new cases and old cases, the system produces results of determining outstanding students based on predetermined criteria, namely student report cards. It can be concluded that the application of the k-Nearest Naeghbor (KNN) method to determine outstanding students gives the expected results and can help facilitate teacher performance.

Keywords: *Classification of Outstanding Students Using the K-neares Neighbor*

KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan sulan penelitian ini dengan judul ***“Klasifikasi Siswa Berorestasi Di Smp Negeri 1 Kwandang Dengan Menggunakan Metode K-Nearest Naeghbor. (Studi Kasus Smp Negeri 1 Kwandang)”*** untuk memenuhi salah satu syarat penyusunan Skripsi Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penyusunan menyadari sepenuhnya bahwa usulan penelitian ini tidak mungkin terwujud tanpa bantuan tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moral maupun materil. Untuk itu, dengan segala keikhlasan dan kerendahkan hati, penulis banyak mengucapkan terimakasih dan penghargaan setinggi-tingginya kepada :

1. Bapak Mohamad Ichsan Gaffar Latjokke, S.E M.Ak Selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo.
2. Bapak Dr. Abd Gaffar Latjoke, M.Si, selaku sebagai Rektor Universitas Ichsan Gorontalo.
3. Ibu Zohrahayaty, S.Kom, M.Kom selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.
4. Bapak Sudirman S. Panna, S.Kom, M.Kom Selaku Wakil Dekan I Bidang Akedemik Fakultas Ilmu Komputer.
5. Ibu Irma Surya Kumala Idris, S.Kom, M.Kom Selaku Wakil Dekan II Bidang Bidang Administrasi Umum Dan Keuangan Fakultas Ilmu Komputer.
6. Bapak Sudirman Melangi, S.Kom, M.Kom Selaku Wakil Dekan III Bidang Kemahasiswaan Fakultas Ilmu Kompuuter.
7. Bapak Irvan Abraham Salihi, S.Kom, M.Kom Selaku Ketua Prongram Studi Tehknik Informatika Fakultas Ilmu Komputer.
8. Ibu Zohrahayaty S.Kom M.Kom selaku pembimbing Utama yang telah membimbing penulis dalam menyusun penelitian ini.
9. Ibu Sumarni, S.Kom, M.Kom selaku Pendamping yang telah membimbing penulis dalam menyusun penelitian ini.
10. Bapak dan Ibu Dosen Fakultas Ilmu

Komputer Universitas Ichsan Gorontalo yang telah banyak membagikan ilmunya selama penulis melaksanakan studi.

11. Kepada Kedua Orang Tua dan seluruh keluarga yang selalu memberikan kasih sayang, bimbingan, perhatian dan doa keberhasilan studi ini.
12. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan moril yang sangat besar kepada penulis;
13. Kepada semua pihak yang ikut membantu dan menyelesaikan proposal ini yang tak sempat penulis sebutkan satu-persatu

Semoga Allah, SWT melimpahkan balasan atas jasa- jasa mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis mengharapkan adanya kritik dan saran yang konstruktif. Akhirnya penulis berharap semoga hasil ini di capai dapat bermanfaat bagi kita semua, Amiin.

Gorontalo, 03 Desember 2020

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PERSETUJUAN	iii
PERNYATAAN SKRIPSI.....	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xiv
DAFTAL TABEL.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	3
1.3 Rumusan Masalah	4
1.4 Tinjauan Penelitian.....	4
1.5 Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Tinjauan Studi	5
2.2 Tinjauan Teori.....	6

2.2.1 Prestasi Belajar.....	6
2.2.2 Data Mining	6
2.2.3 Algoritma K-Nearest Naighbor.....	8
2.2.4 Penerapan Algoritma Pada Sistem	10
2.2.5 Siklus Waterfall.....	12
2.2.6 Desain Sistem.....	15
2.2.7 Desain Sistem Secara Umum	16
2.2.8 Desain Sistem Secara Terinci.....	16
2.2.9 Implementai Algoritma Pada Sistem	22
2.2.10 Metode Pengujian.....	22
2.3 Framework	25
2.4 Perangkat Lunak Pendukung.....	26
2.5 Kerangka Berpikir	27
BAB III OBJEK DAN METODE PENELITIAN.....	28
3.1Objek Penelitian	28
3.2Metode Penelitian.....	28
3.3 Tahap Analisis.....	29
3.4 Tahap Desain.....	30
3.5 Tahap Pembuatan	31
3.6 Tahap Pengujian.....	31
3.7 Tahap Implementasi	32

BAB IV ANALISIS DAN DESAIN SISTEM	33
4.1 Hasil Pengumpulan Data.....	33
4.2 Hasil Pemodelan Sistem.....	33
4.2.1 Pemodelan Dengan Algoritma K-Nearest Naeghbor	33
4.3 Penerapan Algoritma K-Nearest Naeghbor	37
4.4 Diagram Konteks	37
4.5 Diagram Berjenjang	38
4.6 Diagram Arus Data	38
4.6.1 Diagram Arus Data Level 0	38
4.6.2 Diagram Arus Data Level 1	39
4.6.3 Diagram Arus Data Level 2	39
4.6.4 Entity Relationship.....	40
4.7 Kamus Data.....	40
4.8 Interface Design	41
4.8.1 Mekanisme User.....	41
4.8.2 Interface Desain Login.....	42
4.8.3 Tampilan Input Nilai	42
4.8.4 Tampilan Proses	43
4.8.5 Tampilan Hasil	43
4.9 Data Base	44
4.10 Desain Struktur Data	44

4.10.1 Desain Struktur Private	44
4.10.2 Desain Struktur Data Latih	45
4.10.3 Desain Struktur Data Hasil	45
4.11 Arsitektur Sistem.....	46
4.12 Pengujian White Box	46
4.13 Flowchart Program Untuk Pengujian.....	52
4.14 Flowgarph Untuk Pengujian White Box.....	53
4.15 Perhitungan Cyclomatic Complexity	53
4.16 Path Pada Pengujian White Box	54
4.17 Pengujian Black Box.....	54
BAB V PEMBAHASAN	56
5.1 Pembahasan Model	56
5.2 Pembahasan Sistem.....	56
5.2.1 Hasil Tampilan Window	56
5.2.2 Tampilan Login.....	56
5.2.3 Tampilan Home.....	57
5.2.4 Tampilan Input Nilai	57
5.2.5 Tampilan Proses	58
5.2.6 Tampilan Hasil	58

BAB VI KESIMPULAN DAN SARAN.....	59
6.1 Kesimpulan	59
6.2 SARAN	59
DAFTAR PUSTAKA	

DAFTAR GAMBAR

Gambar 2.1 Langkah-Langkah Proses Data Mining.....	6
Gambar 2.2 Algoritma K-Nearest Naeghbor	9
Gambar 2.3 Tahapan Metode Waterfal	13
Gambar 2.4 Ketentuan Notasi Luar DAD.....	21
Gambar 2.5 Notasi Arus Data DAD	21
Gambar 2.6 Notasi Arus Data DAD	21
Gambar 2.7 Notasi Simpan Data.....	22
Gambar 2.8 Bagan Air	23
Gambar 2.9 Grafik Alir	23
Gambar 2.10 Pengujian Kotak Hitam	25
Gambar 2.11 Kerangka Pikir.....	27
Gambar 3.1 Model Usulan	28
Gambar 4.1 Algoritma K-Nearest Naeghbor	37
Gambar 4.2 Diagram Konteks.....	37
Gambar 4.3 Diagram Berjenjang	38
Gambar 4.4 DAD Level 0	38
Gambar 4.5 DAD Level 1	39
Gambar 4.6 DAD Level 2	39
Gambar 4.7 Entity Relationship Diagram.....	40
Gambar 4.8 Desain Login	42

Gambar 4.9 Input Nilai	42
Gambar 4.10 Proses	43
Gambar 4.11 Hasil	43
Gambar 4.12 Data Base.....	44
Gambar 4.13 Flowchart Pengujian.....	52
Gambar 4.14 Flowgraph Pengujian.....	53
Gambar 5.1 Tampilan Window sistem	56
Gambar 5.2 Tampilan Login	56
Gambar 5.3 Tampilan Home	57
Gambar 5.4 Tampilan Input Nilai	57
Gambar 5.5 Tampilan Proses	58
Gambar 5.6 Tampilan Hasil	58

DAFTAR TABEL

Tabel 1.1 Dataset Siswa SMP Negeri 1 Kwandang.....	1
Tabel 2.1 Tinjauan Studi	5
Tabel 2.2 Nilai Siswa Kelas 7	10
Tabel 2.3 Data Setelah Mendapatkan jarak.....	11
Tabel 2.4 Pengurutan dan Ambil Jarak Terdekat.....	12
Tabel 2.5 Bagan Alir Sistem	18
Tabel 2.6 Perangkat Lunak Pendukung	26
Tabel 4.1 Hasil Pengumpulan Data.....	33
Tabel 4.2 Kriteria Siswa Berprestasi.....	34
Tabel 4.3 Data Trining Siswa Smp 1 Negeri Kwandang.....	34
Tabel 4.4 Data Sampel Smp 1 Negeri Kwandang.....	35
Tabel 4.5 Hasil Jarak Terdekat	36
Tabel 4.6 Kamus Data Penilaian.....	40
Tabel 4.7 Data Perhitungan Bobot Kategori.....	40
Tabel 4.8 Mekanisme User	41
Tabel 4.9 Struktur Data Private.....	44
Tabel 4.10 Data Latih.....	45
Tabel 4.11 Struktur Data Hasil.....	45
Tabel 4.12 Path Pengujian White Box	54
Tabel 4.13 Pengujian Black Box.....	54

BAB I

PENDAHULUAN

1. 1 Latar Belakang

Pendidikan mempunyai peranan yang sangat penting untuk kemajuan bangsa ini. Salah satu hal yang tidak dapat dilepaskan dari pendidikan yaitu prestasi dan perilaku peserta didik. Prestasi dan perilaku setiap siswa merupakan sebuah hal yang sudah menjadi sesuatu yang penting bagi semua pihak yang terkait secara langsung maupun tidak langsung. Saat ini, lembaga pendidikan setiap tahunnya terus berusaha meningkatkan mutu pendidikan khususnya siswa Sekolah Menengah Pertama (SMP) dengan harapan lulusannya dapat memiliki potensi yang memuaskan dibandingkan sekolah sederajat, hal tersebut dilakukan demi meningkatkan kualitas lulusan. Dalam rangka mencapai tujuan yang dimaksud, sekolah sebagai institusi pendidikan, mengembangkan berbagai sistem pembinaan yang sifatnya memotivasi dan mengembangkan potensi para siswa. Salah satu kegiatan untuk proses pengembangan potensi setiap siswa adalah melalui pemilihan siswa berprestasi.

Sekarang, proses penentuan siswa berprestasi seharusnya tidak dinilai berdasarkan aspek akademik saja, melainkan aspek non-akademik juga seperti ekstrakurikuler, kepribadian, dan kedisiplinan karena kemampuan setiap siswa itu berbeda-beda. Sehingga di dalam sebuah sekolah siswa dinyatakan berprestasi tidak hanya dinilai dari hasil-hasil nilai mata pelajaran saja melainkan kemampuan diri juga harus menjadi pendukung penilaian dimana siswa tersebut dapat dikatakan berprestasi.

Permasalahan pada Smp Negeri 1 Kwandang yaitu belum seimbang dalam melakukan penilaian dikarenakan nilai akademik masih menjadi pertimbangan yang utama dibandingkan nilai non-akademik. Nilai non-akademik hanya dijadikan data penunjang yang tidak jelas dalam pembobotan nilai. Sehingga dirasa kurang adil dalam penentuan siswa berprestasi. Dalam sekolah siswa-siswa yang berprestasi itu tidak hanya ditimbang dari aspek akademik, tapi aspek non-akademik juga harus diper-timbangkan. Dengan memiliki banyak kriteria dalam penentuan siswa berprestasi akan muncul permasalahan baru yaitu susah untuk

memasukkan nilai dengan pembobotan yang optimal, sehingga dalam melakukan perhitungan siswa berprestasi akan memakan waktu yang tidak singkat.

Menyikapi hal tersebut diatas, pada penelitian ini penyusun berusaha untuk membantu Smp Negeri 1 Kwandang dengan menerapkan sistem yang mampu bekerja secara cepat, tepat, dan obyektif dalam pengelompokan siswa berprestasi sehingga hasil yang dikeluarkan tersebut valid dan benar-benar bisa dikatakan bahwa siswa tersebut merupakan siswa yang berprestasi. Salah satu metode pendukung yang digunakan dalam pengklasifikasian tersebut adalah *K-Nearest Neighbor* (KNN) karena metode ini merupakan sebuah metode yang menggunakan algoritma *supervised* dimana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada KNN yang tujuannya untuk mengklasifikasikan data siswa yang berprestasi dan tidak berprestasi.

Tabel 1.1 Dataset Siswa Smp Negeri 1 Kwandang

NO	NAMA	BHS.Indo	Matematika	IPA	BHS.Ingggris
1	QONITA AZZAHIR	98	98	98	97
2	AINTIYA	95	97	84	95
3	VERA ISMAIL	95	97	90	95
4	CRISANDY	95	97	82	95
5	WAHYU	95	97	84	95
6	ANDRE SALEH	89	91	82	89
7	RIZKI MALE	89	91	98	89
8	SRI FADILA	89	91	82	89
9	RAHMAT WIJAYA	89	91	82	89
10	MOHAMAD RAHIM	82	84	84	82

Data Mining (Witten, 2011) didefinisikan sebagai proses penemuan pola dalam data. Berdasarkan tugasnya, *data mining* dikelompokkan menjadi deskripsi, estimasi, prediksi, klasifikasi, *clustering* dan asosiasi (Larose, 2005).

Algoritma *K-Nearest Neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek yang berdasarkan dari data pembelajaran yang jaraknya paling dekat dengan objek tersebut. KNN merupakan algoritma *supervised learning* dimana hasil dari *query instance* yang baru diklasifikan berdasarkan mayoritas dari kategori pada algoritma KNN (Asahar Johar T, 2016).

Berdasarkan penelitian sebelumnya dilakukan oleh Ari Sulistyo menggunakan metode *K-Nearest Neighbor* dengan judul penentuan jurusan disekolah menengah atas menggunakan metode *K-Nearest Neighbor* mempunyai keakuratan sebesar 79,68 % dengan menggunakan tujuh kriteria (Ari Sulistyo, 2014).

Adapun penelitian yang dilakukan oleh Mursyid menggunakan metode *K-Nearest Neighbor* dengan judul Aplikasi Website Interface, hasil pengujian yang mempunyai keakuratan 85% yang berisi diagnosa saran bagi orang tua (Mursyid, 2015).

Berdasarkan latar belakang yang telah diuraikan, maka dibuatlah penelitian dengan judul “**KLASIFIKASI SISWA BERPRESTASI PADA SMP NEGERI 1 KWANDANG YANG MENGGUNAKAN METODE K-NEAREST NEIGHBOR (KNN)**”.

1. 2 Identifikasi Masalah

Belum adanya sistem yang bekerja secara cepat, tepat, dan obyektif dalam klasifikasi siswa berprestasi. Untuk mengetahui siswa berprestasi pihak sekolah masih menggunakan komputerisasi untuk melakukan perhitungan dengan memanfaatkan aplikasi *microsoft excel*.

1. 3 Rumusan Masalah

Dari beberapa uraian yang telah dijelaskan pada latar belakang di atas, maka penulis dapat menyimpulkan rumusan masalah yaitu :

1. Bagaimana pembuatan sistem klasifikasi siswa berprestasi di Smp Negeri 1 Kwandang.?
2. Bagaimana implementasi sistem klasifikasi siswa berprestasi di Smp Negeri 1 Kwandang.?

1. 4 Tujuan Penelitian

Adapun tujuan dari penelitian ini antara lain adalah sebagai berikut :

Menerapkan sistem klasifikasi siswa berprestasi di Smp Negeri 1 Kwandang.

Mengimplementasikan sistem klasifikasi siswa berprestasi di Smp Negeri 1 Kwandang.

1. 5 Manfaat Penelitian

Adanya penelitian ini diharapkan dapat memberikan manfaat baik secara teoritis maupun secara praktis:

Manfaat teoritis

Hasil dari penelitian ini diharapkan dapat memberikan sumbangsi berupa kontribusi pemikiran maupun kajian tentang pembuatan sistem klasifikasi siswa berprestasi.

Manfaat praktis

- a). Memberikan kontribusi positif bagi lembaga pendidikan khususnya Smp Negeri 1 Kwandang.
- b). Menambah ilmu dan pengetahuan tentang bagaimana cara membuat sistem klasifikasi siswa berprestasi.

BAB II

TINJAUAN PUSTAKA

2. 1 Tinjauan Studi

Tabel 2. 1 Tinjauan Studi

No	Penelitian	Judul	Tahun	Metode	Hasil
1	Jodi Irjaya Kartika Edy Santoso Sutrisno	Penentuan Siswa Berpre stasi	2017	<i>K-Knearest Neighbor dan Weighted Product</i>	pengujian dimana penulis merubah nilai bobot. Dengan tujuan untuk memperole h akurasi yang lebih baik.
2	Raymundus Nandi Irawan Wawan Laksito YS Sri siswanti	Sistem Pendukung keputusan Untuk menentukan Status Pestasi Siswa	2013	<i>K-Knearest Neighbor</i>	Pengujian dilakukan dengan membandin gkan antara perhitunga n manual dengan komputeris as.
3	Moenawar Kholil Kusrini Henderi	Penerapan Metode K- Knearest Neighbor Dalam Proses Seleksi Penerima Beasiswa	2018	<i>K-Knearest Neighbor</i>	Pengujian ini dilakukan untuk membagi semua data tering menjadi 10 blok data, yang masing- masing blok terdapat 50 blok data.

2. 2 Tinjauan Teori

2.2. 1 Prestasi Belajar

Prestasi merupakan hasil akhir dari pekerjaan yang telah dilakukan. Prestasi tidak akan diperoleh sebelum seseorang melakukan kegiatan. Bagi siswa prestasi merupakan sesuatu yang amat penting lebih khusus lagi dalam hal prestasi belajar, karena nilai yang dicapai dalam proses belajar adalah prestasi yang dapat dilihat secara nyata (Setiawan, 2009).

2.2. 2 Data Mining

Data mining (Witten, 2011), didefinisikan sebagai proses penemuan pola dalam data. Berdasarkan tugasnya, *data mining* dikelompokkan menjadi deskripsi, estimasi, prediksi, klasifikasi, *clustering* dan asosiasi (Larose, 2005). Proses dalam tahap *data mining* terdiri dari tiga langkah Utama (Sumathi, 2006), yaitu :

a. *Data Preparation*

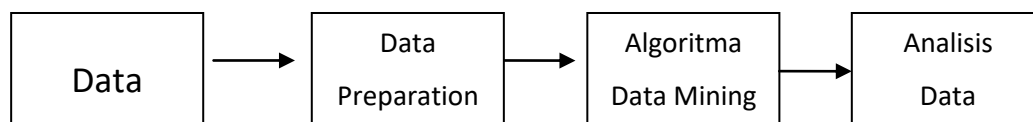
Pada langka ini, data dipilih, dibersihkan, dan dilakukan *preprocessed* mengikuti pedoman dan *knowledge* dari ahli domain yang menangkap dan mengintegrasikan data internal dan eksternal ke dalam tinjauan organisasi secara menyeluruh.

b. Algoritma *Data Mining*

Penggunaan algoritma *data mining* dilakukan pada langka ini untuk mengenali data yang terintegrasikan untuk memudahkan identifikasi informasi bernilai.

c. Fase Analisis Data

keluaran dari *data mining* evaluasi untuk melihat apakah *knowledge* domain ditemukan dalam bentuk *rule* yang telah diekstrak dari jaringan.



Gambar 2. 1 Langkah-Langkah Dalam Proses Data Mining

Sumber Dari (Maimon dan Rokach 2010)

Data mining merupakan bidang dari beberapa bidang keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistik, database, dan visualisasi untuk pengenalan permasalahan pengambilan informasi dari database yang besar (Larose, 2005). Data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan machine learning untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari database yang besar.

Menurut Larose, data *mining* dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu (Larose, 2005) :

1. Deskripsi

Untuk mencari cara yang menggambarkan pola dan kecenderungan yang terdapat dalam data. Deskripsi dari pola dan kecenderungan yang sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan

2. Estimasi

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

3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang.

4. Klasifikasi

Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, sedang dan rendah.

5. Pengklusteran

Pengklusteran merupakan pengelompokan *record*, pengamatan atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah kumpulan record yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan *record* dalam kluster lain. Algoritma pengklusteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), yang mana

kemiripan *record* dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal

6. Asosiasi

Tugas asosiasi dalam data mining adalah menemukan atribut yang muncul dalam suatu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja.

2.2. 3 Algoritma *K-Nearest Neighbor*

External variable secara langsung akan mempengaruhi persepsi manfaat dan persepsi kemudahan dari pengguna. Persepsi kemudahan penggunaan dipengaruhi oleh variabel eksternal yang berkenaan dengan karakteristik sistem yang meningkatkan penggunaan dari teknologi, seperti *mouse*, *touch screen*, *menu* dan selain itu, pelatihan individu juga akan mempengaruhi kemudahan penggunaan. Semakin banyak pelatihan yang diterima individu, semakin besar tingkat kemudahan dalam penggunaan.

Tujuan algoritma KNN adalah mengklasifikasikan obyek baru berdasarkan atribut dan *training sample*. *Classifier* tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik *query*, akan ditemukan sejumlah k obyek atau (titik training) yang paling dekat dengan titik *query*. Klasifikasi menggunakan voting terbanyak diantara klasifikasi dari k obyek. Algoritma KNN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari *query instance* yang baru. Algoritma metode KNN sangatlah sederhana, bekerja berdasarkan jarak terpendek dari *query instance* ke *training sample* untuk menentukan KNN-nya.

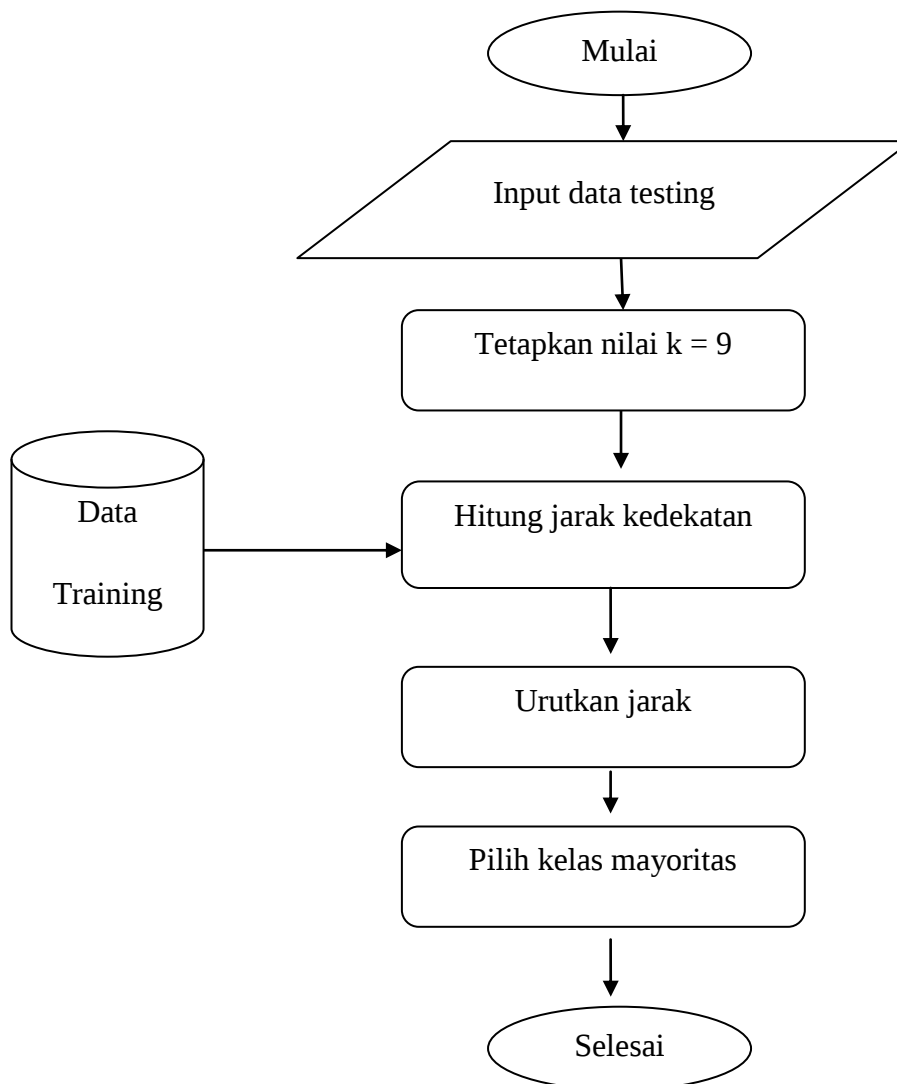
Nilai k yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus dimana klasifikasi diprediksikan berdasarkan training data yang paling dekat (dengan kata lain, $k=1$) disebut algoritma Nearest Neighbor.

Kelebihan KNN (*K-Nearest Neighbor*):

1. Tangguh terhadap *training* data yang memiliki banyak *noise*.
2. Efektif apabila *training* datanya besar.

Kelemahan KNN (*K-Nearest Neighbor*):

1. KNN perlu menentukan nilai dari parameter k (jumlah dari tetangga terdekat).
2. *Training* berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan.
3. Atribut mana yang harus digunakan untuk mendapatkan hasil terbaik.
4. Biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap *query instance* pada keseluruhan *training sample*.



Gambar 2. 2 Algoritma K-NN

1. Tentukan parameter K
2. Hitung jarak antara data yang akan dievaluasi dengan semua pelatihan
3. Urutkan jarak yang terbentuk (urut naik)
4. Tentukan jarak terdekat sampai urutan K
5. Pasangkan kelas yang bersesuaian
6. Cari jumlah kelas dari tetangga yang terdekat dan tetapkan kelas tersebut sebagai kelas data yang akan dievaluasi

Rumus KNN:

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2} \quad (1)$$

Keterangan:

x_1 = Sampel Data

x_2 = Data Uji / Testing i = Variabel Data

d = Jarak

p = Dimensi Data

2.2.4 Penerapan Algoritma Pada Sistem

Penerapan algoritma K-nearest neighbor menggunakan web. Pada algoritma tersebut dalam sistem pengolahan data yang menggunakan klasifikasi sebagai berikut.

Tabel 2. 2 Nilai Siswa Kelas 7

No	Nama	Rata ² ManelDiknas	Rata ² ManelMulo	Rata ² Penge mbangan diri	Prestasi kejuaraan
1	Qonita A	60	60	66	20
2	Aintiva Y	65	75	75	20
3	Rika K	64	68	75	10
4	Sarimah	75	68	65	10
5	Sasha	68	67	65	10
6	Marcel	87	68	61	10
7	Vera I	78	66	77	20

8	Melani	78	69	80	10
9	Bisma	67	64	87	10
10	Crisandy	78	65	78	20
11	Billy	75	80	85	10
12	Riska	88	71	65	10
13	Andreas	78	65	68	10
14	Wahyu	87	75	78	20
15	Puspa	98	61	87	10
16	Izzatun	78	64	68	30
17	Iqbal	78	66	78	10
18	Jefri M	67	67	75	10
19	Natalio	78	67	75	20
20	Cecep	65	68	77	10
21	Lanang	66	62	87	10
22	Muh toni	77	64	77	10
23	Jafrie	63	67	78	10
24	Sri Fadila	67	69	80	20
25	Kristan P	64	77	80	10
26	Rahmawaty	67	71	74	20
27	Fikri	62	81	75	10
28	David	67	67	75	10
29	Wita Rahim	61	78	77	20
30	Iqsan	60	77	68	10

Sumber Raymundus Nandi Irawan 2013

- a. Hitung jarak setiap data dengan data uji siswa bernama billy.

$$\begin{aligned}
 d_1 &= \sqrt{(75 - 60)^2 + (80 - 60)^2 + (85 - 66)^2 + (10 - 20)^2} \\
 &= \sqrt{1086} \\
 &= 32.9545
 \end{aligned}$$

Tabel 2. 3 Data Setelah Mendapatkan Jarak

No	Nama	Jarak	No	Nama	Jarak
1	Qonita A	3.295.451	15	Anita	3.088.689
2	Aintiya	1.802.776	16	Iqbal	1.593.738
3	Rika K	1.910.497	17	Jefri M	1.824.829
4	Sarimah	2.332.381	18	Romlah	2.156.386
5	Sasha	2.485.961	19	cecep	1.754.993
6	Marcel	2.939.388	20	Lanang	2.022.375
7	Indah	1.920.937	21	Muh toni	18
8	Melani	124.499	22	Jafrie	190.263
9	Bisma	18	23	Daniel	1.760.682
10	C.Cahya	1.957.039	24	Kristan P	124.499
11	Billy	0	25	Yohanes	1.913.113
12	Riska	254.951	26	Fikri	1.643.168
13	Andreas	2.286.919	27	David	1.824.829
14	Amelia	1.783.255	28	Wijanarko	1.907.878
			29	Iqsan	2.286.919

Misalkan jarak dikatakan tinggi ≥ 81 , dikatakan sedang jika $51 < \text{jarak} \leq 80,9999$, dikatakan rendah jika $\leq 50,9999$.

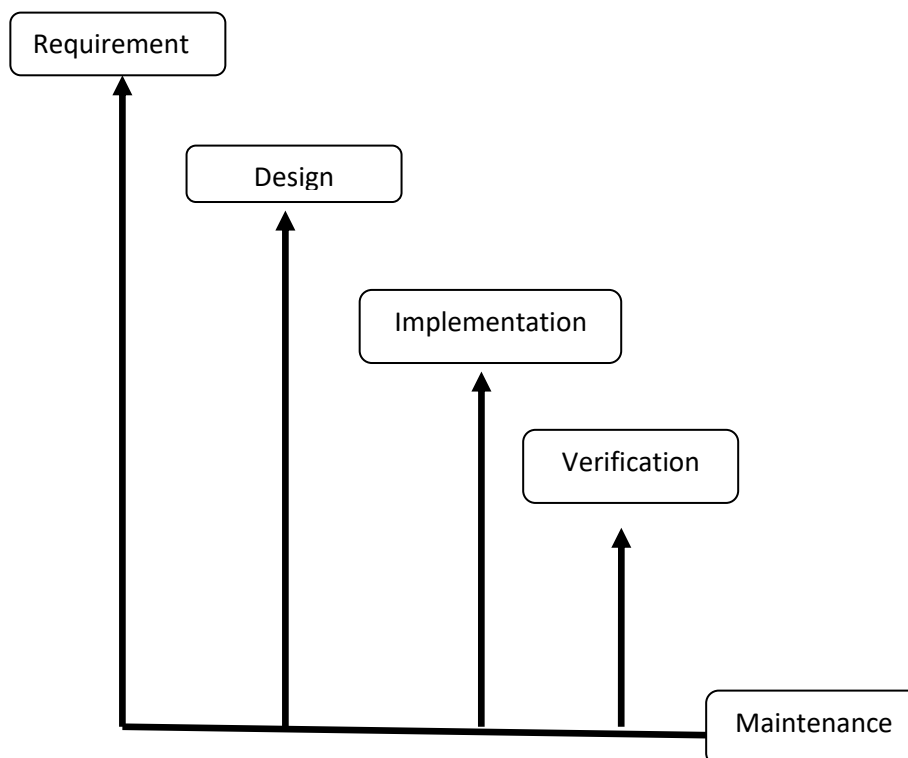
Tabel 2. 4 Pengurutan & Ambil Jarak Terkecil

No	Nama	Jarak	Status
1	Billy	0	Rendah
2	Melani	124.499	Rendah
3	Kristan		Rendah
4	Iqbal	1.593.738	Rendah
5	Fikri	1.643.168	Rendah
6	Cecep	1.754.993	Rendah
7	Daniel	1.760.682	Rendah

2.2.5 Siklus Sistem Waterfal

Pengertian metode *waterfall* metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada

pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perancangan (*planning*), permodelan (*modeling*) konstruksi (*construcion*), serta penyerahan sistem ke para pelanggan atau pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Pressman, 2012). Tahapan metode *waterfall* dapat dilihat pada gambar dibawah ini.



Gambar 2. 3 Tahapan Metode Waterfall

Sumber (rizaldi, 2017)

Tahapan metode *waterfall* dalam pengembangannya memiliki beberapa tahapan yang berurut yaitu *requirement* (analisis kebutuhan), *design system* (desain sistem), *coding* (pengkodean) dan *testing* (pengujian), penerapan program, pemeliharaan. Tahapan-tahapan dari metode *waterfall* adalah sebagai berikut :

1. *Requirement Analisis*

Tahapan ini pengembangan sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan

perangkat lunak tersebut. Informasi ini biasanya diperoleh melalui Wawancara, diskusi untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. Sistem Design

Spesifikasi kebutuhan dari tahapan sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3. implementation

Pada tahap ini, sistem pertama kali dikembangkan diprogram di program kecil yang disebut *unit* dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai *unit testing*.

4. integration & testing

Seluruh *unit* yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing *unit*. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

5. Operation & Maintenance

Tahapan akhir dalam model waterfall. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

2.2. 5 Desain Sistem

Setelah tahap analisis sistem selesai dilakukan, maka analisis sistem telah mendapat gambaran dengan jelas apa yang harus dilakukan. Tiba waktunya sekarang bagi analisis sistem untuk memikirkan bagaimana membentuk sistem tersebut, tahap ini dengan desain sistem (*system design*).

1. Requirement Analisis

Tahapan ini pengembangan sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya diperoleh melalui Wawancara, diskusi untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. *Sistem Design*

Spesifikasi kebutuhan dari tahapan sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3. *implementation*

Pada tahap ini, sistem pertama kali dikembangkan diprogram di program kecil yang disebut *unit* dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai *unit testing*.

4. *integration & testing*

Seluruh *unit* yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing *unit*. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

5. *Operation & Maintenance*

Tahapan akhir dalam model waterfall. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

2.2. 6 **Desain Sistem**

Setelah tahap analisis sistem selesai dilakukan, maka analisis sistem telah mendapat gambaran dengan jelas apa yang harus dilakukan. Tiba waktunya sekarang bagi analisis sistem untuk memikirkan bagaimana membentuk sistem tersebut, tahap ini dengan desain sistem (*system design*).

Whitten, et al. (2004:34) menyatakan *system design* adalah spesifikasi atau intruksi solusi yang teknis dan berbasis komputer untuk persyaratan bisnis yang diidentifikasi dalam analisis sistem.

Desain sistem adalah spesifikasi atau intruksi solusi yang teknis dan berbasis komputer untuk persyaratan bisnis yang didefinisikan dalam analisis sistem. Driver masa kini dan masa yang akan datang paling berimpak pada proses

dan keputusan desain system, banyak organisasi mengidensifikasikan arsitektur teknologi informasi umum yang didasarkan pada drive-drive teknologi ini.

1. Untuk memenuhi kebutuhan kepada pemakai system
2. Untuk memberikan gambaran yang jelas dan rancangan bangun yang lengkap kepada pemrograman komputer dan ahli-ahli teknik lainnya.

Perancangan system adalah suatu keinginan membuat desain teknis yang berdasarkan evaluasi yang dilakukan pada kegiatan analisis. Perancangan disini dimaksud sebagai proses pemahaman dan perancangan suatu system berbasis komputer yang akan menghasilkan komputerisasi.

2.2. 7 Desain Sistem Secara Umum

Tujuan dari desain system secara umum adalah untuk memberikan gambaran secara umum kepada user tentang system yang baru, yang mana merupakan periapan dari desain system secara rinci. Desain secara umum dilakukan oleh analisis system untuk mendefinisikan komponen-komponen system informasi yang didesain secara rinci oleh programer komputer dan ahli teknik lainnya.

Pada tahap ini komponen-komponen system informasi dirancang untuk dikomunikasikan kepada user. Komponen system informasi yang akan didesain adalah model, output, input, database, teknologi dan kontrol.

2.2.8 Desain Sistem Secara Terinci (Detailed System Design)

1. Desain Output Terinci

Desain output secara terinci dimaksud untuk mengetahui bagaimana dan seperti apa bentuk output-output dari system yang baru. Desain output terinci terbagi atas dua yaitu, desain output berbentuk laporan di media kertas dan desain output dalam bentuk dialog di layar terminal.

a. Desain Output Dalam Bentuk Laporan

Desain ini dimaksud untuk menghasilkan output dalam bentuk laporan di media kertas. Bentuk laporan yang paling banyak digunakan adalah dalam bentuk table dan bentuk grafik atau bagan.

b. Desain Output dalam bentuk dialog di layar terminal

Desain ini merupakan rancang bangun dari percakapan antara pemakai system (*user*) dengan komputer. Percakapan ini dapat terdiri atas proses memasukan data ke system, menampilkan output informasi kepada *user*, atau keduanya.

Beberapa strategi dalam pembuatan layar dialog terminal

Dialog pertanyaan atau jawaban

1. Menu

Menu banyak digunakan karena merupakan jalur pemakai yang mudah dipahami dan mudah digunakan. Menu berisi beberapa alternative atau pilihan yang disajikan kepada user. Pilihan menu akan lebih baik bila dikelompokkan berdasarkan fungsinya.

2. Desain Input Terinci

Masukan merupakan awal dimulainya proses informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi-transaksi yang dilakukan oleh organisasi. Data hasil transaksin adalah masukan untuk system informasi. Hasil dari system informasi tidak lepas dari data yang dimasukasn. Desain input terinci dimulai dari desain dokumen dasar apabila tidak didesain dengan baik kemungkinan input yang tercatat bisa salah atau kurang.

Fungsi dokumen dasar dalam penanganan arus data :

- a. Dapat menunjukan macam dari data yang harus dikumpulkan dan ditangkap.
- b. Data dapat dicatat dengan jelas, konsisten dan akurat.
- c. Dapat mendorong lengkapnya data, disebabkan alat yang dibutuhkan disebut satu persatu didalam dokumen dasar.

3. Desain Database

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasi datanya. Database merupakan salah satu komponen yang penting disystem informasi, karena berfungsi sebagai penyedia informasi bagi para pemakainya. Penerapan database dalam system informasi disebut *database system*.

4. Desain Teknologi

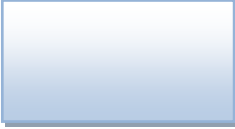
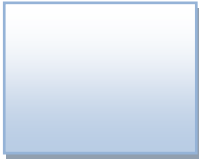
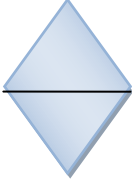
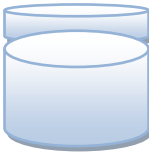
Pada tahap ini yang dilakukan adalah menentukan teknologi yang akan dipergunakan dalam menerima dan menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari system secara keseluruhan.

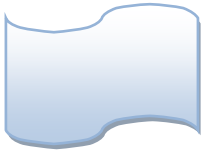
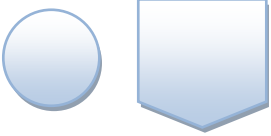
5. Tahapan Desain

Tahap desain terbagi menjadi dua, yaitu desain model secara umum dan terinci. Tahap desain model secara umum berupa desain sistem secara fisik dan logika. Desain fisik dapat digambarkan dengan bagian alir sistem, bagian alir dokumen, dan desain secara logika digambarkan dengan diagram dan arus data (DAD), pada tahap desain model terinci. Modelan didefinisikan secara terinci. Urutan-urutan langkah proses ini diwakili oleh suatu program computer.

Tabel 2. 5 Bagan Alir Sistem

NO	NAMA SIMBOL	SIMBOL	KETERANGAN
1	Simbol Dokumen		Menunjukkan dokumen <i>input</i> dan <i>output</i> baik itu proses manual, mekanik, atau komputer
2	Simbol Kegiatan Manual		Menunjukkan pekerjaan manual
3	Simbol Simpanan Offline		Menunjukkan file non-komputer yang diarsip urut angka (<i>numerical</i>), huruf (<i>alphabetical</i>), atau tunggal (<i>chronological</i>)
4	Simbol Kartu Plong		Menunjukkan <i>input</i> dan <i>output</i> yang menggunakan kartu plong (<i>punched card</i>)

5	Simbol Proses		Menunjukkan kegiatan proses dari operasi program komputer
6	Simbol Operasi Luar		Menunjukkan operasi yang dilakukan diluar proses operasi komputer
7	Simbol Pengaturan Offline		Menunjukkan proses urut data diluar proses komputer. Operasi luar, menunjukkan operasi yang dilakukan diluar proses operasi komputer
8	Simbol Pita Magnetik		Menunjukkan <i>input</i> dan <i>output</i> menggunakan pita <i>magnetic</i>
9	Simbol Hard Disk		Menunjukkan input dan output menggunakan harddisk
10	Simbol Diskette		Menunjukkan input dan output menggunakan diskette
11	Simbol Drum Magnetik		Menunjukkan input dan output menggunakan drum magnetic

12	Simbol Pita Kertas Berlubang		Menunjukkan input dan output menggunakan pita kertas berlubang
13	Simbol Keyboard		Menunjukkan input yang menggunakan online keyboard
14	Simbol Display		Menunjukkan output yang ditampilkan dimonitor
15	Simbol Pita Kontrol		Menunjukkan penggunaan pita kontrol (control tape) dalam batch control total untuk pencocokan di proses batch processing
16	Simbol Hubungan Komunikasi		Menunjukkan proses transmisi data melalui channel komunikasi
17	Simbol Garis Air		Menunjukkan arus dari proses
18	Simbol Penjelasan		Menunjukkan penjelasan dari suatu proses
19	Simbol Penghubung		Menunjukkan penghubung ke halaman yang masih sama atau halaman yang lain

Sumber jogianto, 2005 : 805

Untuk mempermudah penggambaran suatu sistem yang ada atau sistem yang baru yang akan dikembangkan secara logika tanpa memperhitungkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data

tersebut akan disimpan, maka digunakan diagram arus data (DAD) atau data *flow diagram* (DFD). Dalam menggambarkan sistem perlu dilakukan pembentukan simbol, berikut ini simbol-simbol yang sering digunakan dalam (DAD) :

1. *external entity* (*kesatuan luar*) atau *boundary* (batas sistem)

Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima input dan menghasilkan output kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain yang berada di lingkungan luarnya yang akan memberi input serta menerima output dari sistem. (Jogianto HM. 2005 701)



Gambar 2. 4 Ketentuan Notasi Luar DAD

2. Data flow (arus data)

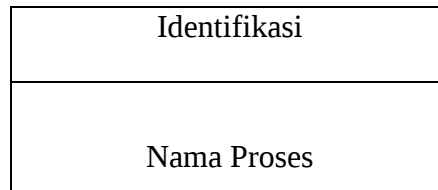
Arus data ini menunjukkan arus atau aliran data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem. (Jogianto HM. 2005 701)



Gambar 2. 5 Notasi Arus Data di DAD

3. *process* (proses)

Suatu proses adalah kegiatan atau kerja yang dilakukan orang, mesin atau komputer dari hasil suatu arus data yang masuk kedalam proses untuk dihasilkan arus data yang akan keluar dari proses. (Jogianto HM. 2005 705)



Gambar 2. 6 Notasi Proses Di DAD

4. Data store (simpanan Data)

Simpanan data pada DFD dapat disimbolkan dengan pasangan horisontal paralel yang tertutup di salah satu ujungnya. (Jogianto HM. 2005 707)



Gambar 2. 7 Notasi Simpan Data

2.2. 9 Implementasi Algoritma Pada Sistem

Tahapan implementasi merupakan tahapan dimana dilakukan transformasi atau penerjemahan dari bahasa modeling ke suatu bahasa pemrograman. Hal ini tugas dari pemrograman, pada pengembangan sistem atau perangkat lunak berorientasi objek penerjemahan dari setiap diagram-diagram DAD yang telah dirancang pada tahap perancangan harus diterjemahkan kedalam bahasa pemrograman sama persis dengan diagram-diagram yang ada guna menghindari terjadinya perubahan fungsi atau tujuan dari pengembangan sistem atau perangkat lunak.

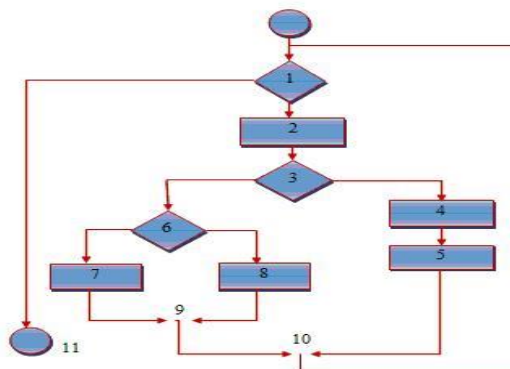
2.2. 10 Metode Pengujian

1. White Box Testing

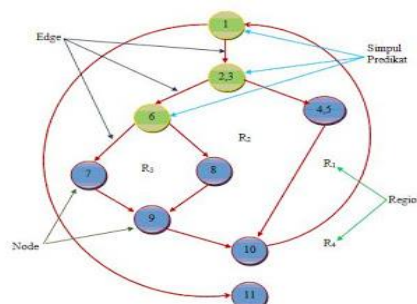
White box testing adalah metode desain test case yang menggunakan struktur control desain procedural untuk mendapatkan test case. Dengan menggunakan metode white box analisis sistem akan dapat memperoleh test case yang meliputi :

- Menjamin seluruh independen path didalam modul yang dikerjakan sekurang-kurangnya sekali.
- Mengerjakan seluruh keputusan logical.
- Mengerjakan seluruh loop sesuai dengan batasannya.
- Mengerjakan seluruh struktur data internal untuk menjamin validitas.

Untuk melakukan pengujian *test case* terlebih dahulu dilakukan . penerjemahan *Flowchart* kedalam notasi *Flowgraph*.



Gambar 2. 8 Bagan Alir



Gambar 2. 9 Bagan Grafik

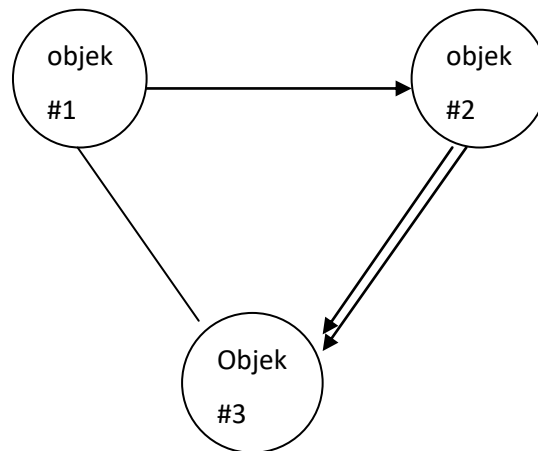
Ada beberapa istilah saat pembuatan flowgrap, yaitu :

1. *Note* yaitu lingkaran pada flowgrap yang menggambarkan satu atau lebih perintah procedural.
2. *Edge* yaitu tanda panah yang menggambarkan aliran kontrol dan setiap node harus mempunyai tujuan node.
3. *Region* yaitu daerah yang dibatasi oleh node dan edge dan untuk menghitung region daerah luar flowgrap juga harus dihitung.
4. *Predicat* node yaitu kondisi yang terdapat pada node dan mempunyai karakteristik dua atau lebih edge lainnya.
5. *Cylomatic complexity* yaitu metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kekompleksan logical program dan dapat digunakan untuk mencari jumlah path dalam satu flowgrap.
6. *Independen path* yaitu jalur melintas atau melalui program dimana sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru.

Dalam pelaksanaanya menggunakan teknik pengujian black box ini mempunyai tiga langkah yaitu :

1. Menggambar flowgraph yang ditransfer dari flowchart.
 2. Menghitung Cylomatic complexity untuk flowgraph yang telah dibuat
 3. Menentukan jalur pengujian dari flowgraph yang berjumlah sesuai dengan Cylomatic complexity yang telah ditentukan
- ## 2. Black Box Testing

Menurut Ladjamudin (2006 : 379) pengujian black box berfokus pada persyaratan fungsional perangkat lunak. konsep pengujian black box kotak hitam) digunakan untuk merepresentasikan sistem yang cara kerja didalamnya tidak tersedia untuk diinspeksi karena item-item yang diuji dianggap gelap karena logikanya tidak diketahui, yang diketahui hanya apa yang dimasuk dan apa yang dikeluarkan dari kotak hitam.



Gambar 2. 10 Pengujian Kotak Hitam

2.3 Framework

Framework merupakan kerangka kerja yang dapat membantu pengembangan aplikasi dalam menangani suatu masalah pemrograman. *Codeigniter* dan *CakePHP*, merupakan framework yang menggunakan konsep MVC(*Model-View-Control*), permasalahan dari dua *framework* adalah belum mengetahui kelebihan dan kekurangan satu *framework* dengan *framework* lain. Untuk itu akan lakukan penelitian mengenai perbandingan dua buah *framework* dengan metode perbandingan dan metode MVC (*Model-View-Control*) dalam menangani suatu permasalahan. Metode MVC (*Model-View-Contro*) dapat di terapkan oleh semua *framework* untuk membangun suatu aplikasi, sedangkan dengan penerapan metode perbandingan diharapkan dapat membandingkan kinerja dari performa, arsitektur, ,dan komunitas. (Dewi Rosmala dan Muhammad Ichwan, 2011).

framework adalah suatu kerangka kerja atau juga dapat di artikan sebagai kumpulan *script* yang dapat membantu pengembangan aplikasi dalam menangani berbagai masalah *pemrograman* seperti koneksi ke *database*, , pem anggilan *variable*, dan *file*. sehingga *developer* lebih cepat membangun aplikasi. Macam- macam *framework* seperti *codeigniter*, *cakephp*, *yii*, *spring*, *zend*, *panda*, dan *hibernate*. Di sini penulis menggunakan dua *framework* yaitu

codeigniter dan *cakephp*, di karenakan dua *framework* *open source*, mendukung *php 5* dan *php 4* komunitas penggunanya besar dan forum yang cukup baik.

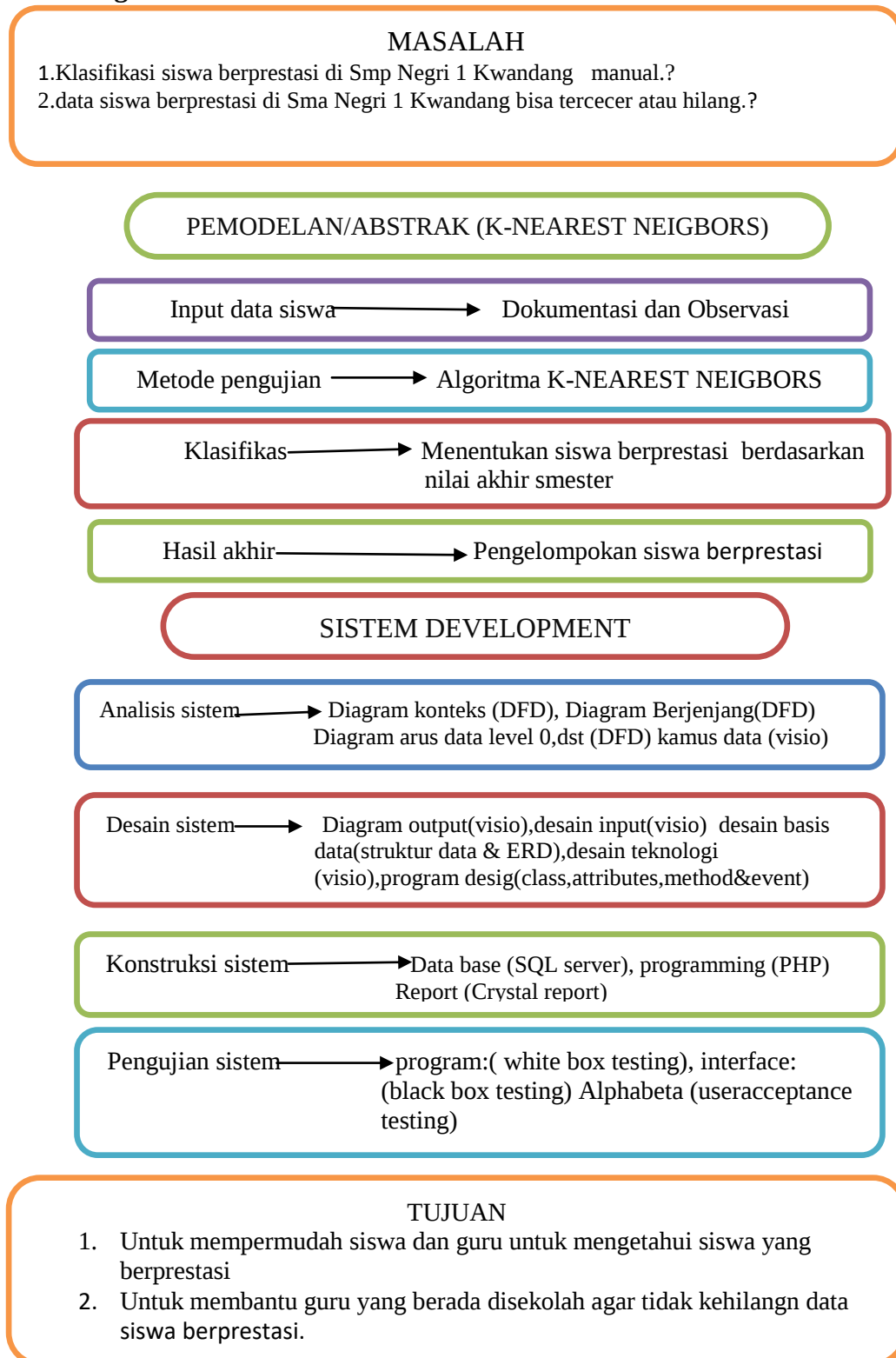
Metode yang di gunakan oleh kedua *framework* adalah metode MVC (*Model- View-Control*). Tetapi dari *framework* yang di berikan masih ada permasalahan yaitu belum diketahui perbedaan dan keunggulan satu *framework* dengan *framework* lain. Cara mengetahui perbedaan dan keunggulan dua *framework* dengan cara Membandingkan dua *framework* dengan menggunakan teknik perbandingan dan metode MVC(*Model- View-Control*). dapat di terapkan oleh semua *framework* untuk membangun suatu aplikasi, dan perbandingan di gunakan untuk membandingkan dari segi *Performa*, *Arsitektur*, dan *Komunitas*.

2.4 Software pendukung

Tabel 2. 6 Perangkat Lunak Pendukung

NO	TOOLS	KEGUNAAN
1	PHP	Sebuah bahasa scripting yang terpasang pada html, yang bertujuan untuk memungkinkan perancang web menulis halaman web dinamik dengan cepat.
2	MySQL	Salah satu pengolah database yang menggunakan SQL (<i>Strukture Query Language</i>) sebagai bahan mengakses databasenya. Yang memiliki keuntungan seperti <i>open source</i> dan memiliki kemampuan menampung kapasitas yang besar

2.5 Kerangka Pikir



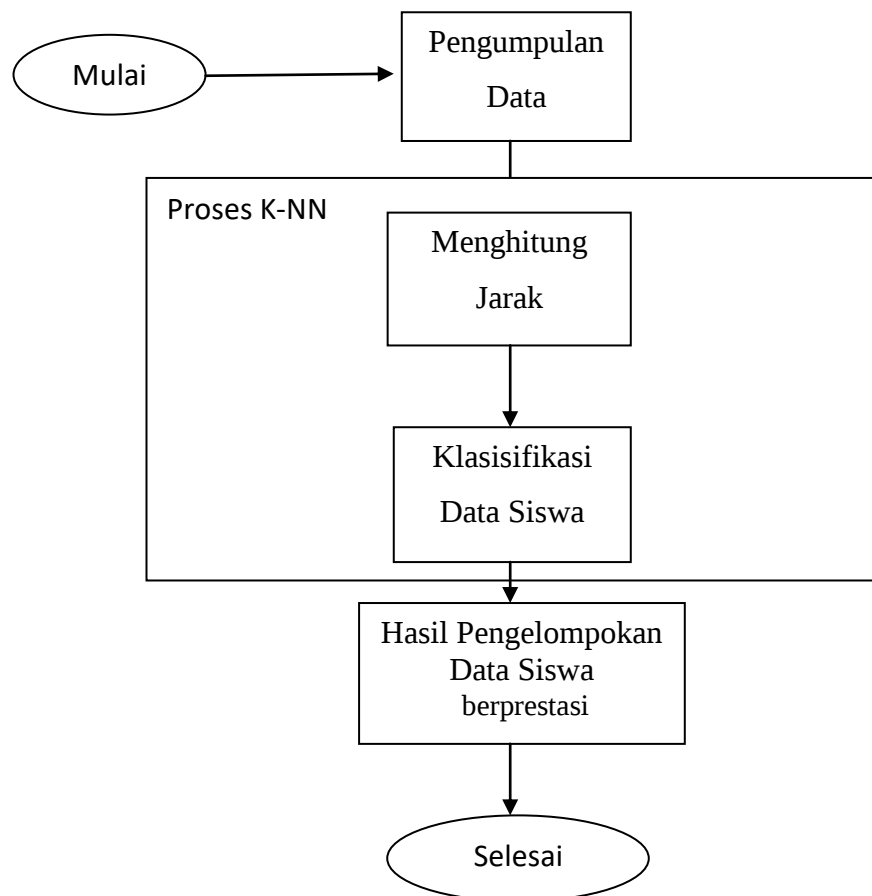
Gambar 2.11 Kerangka Pikir

BAB III

METODE PENELITIAN

3. 1 Objek Penelitian

Berdasarkan latar belakang dan kerangka pemikiran seperti yang telah diuraikan dalam Bab I dan Bab II, maka yang menjadi objek penelitian adalah *Klasifikasi Data Siswa Berprestasi*.



Gambar 3. 1 Model Usulan

3. 2 Metode Penelitian

Metode yang digunakan pada penelitian ini mengembangkan metode *K-Nearest Naighbor* untuk klasifikasi siswa berprestasi. penelitian ini adalah proses pengelompokan data-data yang telah dikumpulkan sebelumnya dengan tujuan

untuk menentukan variabel-variabel yang akan digunakan beserta himpunan-himpunan yang termasuk kedalam variabel-variabel yang digunakan.

a. Sumber Data

Data sekunder merupakan data yang diperoleh dari studi literatur yang dilakukan. Sumber dari studi literatur yaitu jurnal, makalah ilmiah, atau buku yang membahas tentang penelitian algoritma k-nearest neighbor.

Data primer merupakan data yang diperoleh dari penelitian. Data primer pada penelitian ini yaitu berupa data siswa berprestasi dalam proses pengelompokan data-data yang telah dikumpulkan.

b. Cara Pengumpulan Data

Pada penelitian ini digunakan beberapa cara untuk mengumpulkan data diantaranya

1. Observasi

Observasi merupakan salah satu teknik pengumpulan data yang cukup efektif untuk mempelajari dan mengamati secara langsung.

2. Wawancara

Wawancara dilakukan dengan pihak yang terkait yakni pihak sekolah yang berada di SMP Negeri 1 Kwandang sebagai objek penelitian untuk mendapatkan informasi mengenai siswa berprestasi.

3. 3 Tahap Analisis

Tahap analisis penerapan algoritma k-nearest neighbor untuk klasifikasi siswa berprestasi, sebagai berikut :

a. Analisis Sistem Berjalan

Selama ini SMP Negeri 1 Kwandang belum memiliki aplikasi atau sistem. Sehingga pihak sekolah yang berada di Smp Negeri 1 Kwandang. Hanya memanfaatkan *microsoft excel* untuk melakukan perhitungan nilai siswa untuk mengetahui siswa berprestasi dan menyimpan data.

b. Analisis Yang Diusulkan

Berdasarkan analisis yang sedang berjalan maka dengan demikian perlu adanya suatu system yang dapat mendukung pihak sekolah yang berada di SMP Negeri 1 Kwandang dengan menggunakan teknik data mining. Sistem ini terdiri dari :

1. Entry Data : Enteri data atribut siswa SMP Negeri 1 Kwandang
2. Proses : Proses klasifikasi data siswa berprestasi pada algoritma K-Nearest Neighbor
3. Laporan : Laporan atribut siswa berprestasi dan laporan hasil klasifikasi siswa berprestasi menggunakan metode K-Nearest Neighbor.

3. 4 Tahapan Desain

a. Desain Model

Merupakan tahapan yang lebih fokus pada spesifikasi detail berbasis komputer. Sistem yang digunakan adalah *Model Driven Design*, yaitu sebuah pendekatan desain sistem yang menekankan penggambaran model sistem untuk mendokumentasikan aspek teknis dan implementasi dari sebuah sistem. Dimana pada tahap ini kita melakukan pertimbangan-pertimbangan mengenai bagaimana suatu sistem akan diterapkan, baik teknologi dan lingkungan implementasi pada tahap ini digunakan DFD, dimana kita memodelkan persyaratan bisnis logi dari sistem informasi. DFD memodelkan keputusan-keputusan teknik dan desain manusia untuk diimplementasikan sebagai bagian dari suatu sistem informasi.

b. Desain Output

dimaksudkan untuk mengetahui bagaimana dan seperti apa bentuk output-output dari system yang akan dibuat. Desain output terinci terbagi atas dua yaitu desain output berbentuk laporan media kertas dan desain output dalam bentuk dialog delay terminal (monitor).

c. Desain Input

Masukan merupakan awal dimulainya proses informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi-transaksi yang dilakukan oleh organisasi. Data hasil transaksi adalah masukan untuk sistem informasi.

Hasil dari system informasi tidak lepas dari data yang dimasukkan. Desain input terinci dimulai dari desain dokumen dasar apabila tidak didesain dengan baik kemungkinan input yang tercatat bisa salah atau kurang.

d. Desain Database

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasi datanya. Database merupakan salah satu komponen yang penting di system informasi, karena berfungsi sebagai penyedia informasi bagi para pemakainya. Penerapan database dalam system informasi disebut *database system*.

e. Desain Teknologi

Pada tahap ini yang dilakukan adalah menentukan teknologi yang akan dipergunakan dalam menerima dan menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari system secara keseluruhan.

3. 5 Tahap Pembuatan

Tahap pembuatan ini merupakan tahapan dimana dimana kita melakukan pengembangan, melakukan tahap produksi system hasil analisa dan desain sistem sebelumnya. Termasuk didalamnya membangun sebuah system penerapan Algoritma K-Nearest Neighbor untuk klasifikasi siswa berprestasi, menginstal paket tambahan untuk menjalankan program , menulis *listing* program dan membangunnya dalam bentuk sebuah antar muka dan tersusun dalam sebuah sistem menu sehingga dapat dijalankan oleh pengguna sistem.

3. 6 Tahap Pengujian

Tahap ini dilakukan setelah melakukan model selesai dibuat. Dan program dapat berjalan, dimana seluruh perangkat lunak , program tambahan, dan semua program yang terlibat dalam pembangunan sistem diuji untuk memastikan sistem dapat berjalan dengan rancangan atau belum. Pengujian yang dilakukan dengan dua teknik yaitu :

a. Whait Box

Dalam pengujian *white box* ini dengan membuat bagan alir program, listing program, grafik alir, pengujian *basispath* serta perhitungan *Ciclomatic Complexity*.

b. Black Box

Pengujian black box yang termasuk dalam tahap ini yaitu memuji antar muka sistem. Apakah sebuah system setelah diberikan kepada pengguna dapat dioperasikan atau tidak.

3. 7 Tahap Implementasi

Tahap implementasi sistem (*System Implementation*) merupakan tahap meletakkan sistem supaya siap dioperasikan pada pihak sekolah yang berada di SMP Negeri 1 Kwandang. Pada tahap ini akan dilakukan pengetesan sistem secara bersamaan anantara analisis sistem (*System Analist*), pemograman (*Programmer*), dan pemakai sistem (*user*).

Adapun beberapa langka yang dilakukan pada tahap ini adalah :

a. Penerapan Atau Penggunaan Program

penerapan instalasi dari program yang telah dibangun ini nantinya akan diterapkan pada pihak sekolah yang berada di Smp Negeri 1 Kwandang.

b. Instalasi Program

langkah selanjutnya adalah menginstal program. Proses penginstalan tidak memakan waktu yang lama.

c. Pelatihan pengguna

Langka berikut tidak kalah pentingnya dengan langkah-langkah sebelumnya, yakni kita harus melatih penggunaan program pada pihak sekolah yang berada di SMP Negeri 1 Kwandang, yang nantinya menggunakan program ini.

d. Enteri Data

Setelah pelatihan pengguna dilakukan, maka hal yang selajutnya kita lakukan adalah memasukan data. Ini dilakukan agar nantinya program yang telah dibangun dapat dinilai oleh pengguna.

BAB IV

ANALISIS DAN DESAIN SISTEM

4.1 Pengumpulan Data

Berikut Ini adalah contoh Perhitungan manual dari Metode *K-Nearest Nighbor* diambil dari beberapa dataset sebagai sampel seperti yang terdapat pada tabel berikut

Tabel 4.1 Hasil Pengumpulan Data

NAMA	BHS.INDO	MATEMATIKA	IPA	BHS.ING
RAHMAN ISHAK	90	80	83	85
AZIZAH NUR KHUMAIRAH	95	85	80	80
NUR AYIN VITRIA RAHMAN	75	85	80	75
IRENE GABRIELA ROMBE	80	83	78	79
RAFIA OLI'I	80	75	80	77
SRI WILIS ISHAK	80	75	75	75
LUTHFIYAH RAMADHANI KABANGUNAN	78	80	70	75
MAGVIRA TAHIR	85	80	78	80
SITI MULFATIA UNO	90	88	78	88
ALFANDI FARID TOYILI	89	84	86	87

4.2 Hasil Pemodelan Sistem

4.2.1 Pemodelan Dengan Algoritma K-Nearest Neighbor

Berikut ini adalah pemodelan Algoritma K-Nearest Neighbor:

1. Menentukan parameter K (jumlah tetangga paling dekat).
2. Menghitung kuadrat jarak *euclid* (*queri instance*) masing-masing objek terhadap data yang diberikan.
3. Kemudian mengurutkan objek-objek tersebut kedalam kelompok yang mempunyai jarak *euclid* terkecil.
4. Mengumpulkan kategori Y (*klasifikasi nearest neighbor*).
5. Memakai kategori *K-Nearest Neighbor* yang paling banyak muncul maka dapat dijadikan sebagai prediksi.

Langkah perhitungan pada sistem klasifikasi siswa berprestasi menggunakan metode *K-Nearest Neighbor* ini adalah sebagai berikut :

Tabel 4.2 Kriteria Siswa Berprestasi

NO	Nama	Kriteria				
		MP_1	MP_2	MP_3		MP_10
		75	75	75	75	75

Adapun data training yang digunakan merupakan data nilai siswa untuk dilakukan klasifikasi siswa berprestasi berikut contoh perhitungan pada klasifikasi siswa berprestasi pada tabel berikut ini :

Tabel 4.3 Data Trining Siswa Smp Negeri 1 kwandang

NO	NISN	NAMA	BHS INDO	MATE"KA	IPA
1	2785	AKRIANTO TUNIO	85	78	80
2	2855	MOH. AGIL SAPUTRA HAS	80	80	85
3	2811	DIAN BAKARI	80	80	80
4	2953	FAUZAN PANJU	80	84	85
5	2906	MOHAMAD SAHRIL MOH	83	80	75
6	2806	ARSAN KADIR	80	83	80
7	2812	DIKI SALEH	85	85	90
8	2987	SRI MILANTI PIO	85	77	80
.....					
50	2862	MOHAMAD ABDUL OLAN	88	78	70

Dalam penelitian ini ada 50 data sebagai data training dan 35 data sebagai data sampel serta menggunakan $K=8$ untuk menentukan banyaknya tetangga terdekat

Tabel 4.4 Data Sampel Smp Negeri Kwandang

NO	NISN	NAMA	BHS.INDO	MATEMATIKA	IPA
1	2888	RAHMAN ISHAK	90	80	83
2	2808	AZIZAH NUR KHUMAIRAH	95	85	80
3	2878	NUR AYIN VITRIA RAHMAN	75	85	80
4	2837	IRENE GABRIELA ROMBE	80	83	78
5	2958	RAFIA OLI'I	80	75	80
6	2916	SRI WILIS ISHAK	80	75	75
7	2850	LUTHFIYAH RAMADHANI KABANGUNAN	78	80	70
8	2957	MAGVIRA TAHIR	85	80	78
.....					
50	2787	ALFANDI FARID TOYILI	89	84	86

Kemudian Membandingkan setiap data sampel dengan data trining untuk menghitung jarak terdekat dengan menggunakan rumus *Euclidean* sebagai Berikut:

$$\begin{aligned}
 d^1 &= \sqrt{(90-80)^2 - (85-85)^2 - (87-80)^2 - (80-78)^2 - (83-80)^2 - (85-85)^2 - (87-80)^2 - (95-80)^2 - \\
 &\quad (88-80)^2 - (90-80)^2} \\
 &= \sqrt{(100) + (0) + (49) + (4) + (9) + (0) + (49) + (225) + (64) + (100)} \\
 &= \sqrt{600} \\
 &= 24,49489
 \end{aligned}$$

$$\begin{aligned}
 d^2 &= \sqrt{(95-78)^2 - (88-80)^2 - (82-76)^2 - (85-80)^2 - (80-85)^2 - (80-75)^2 - (85-80)^2 - (90-80)^2 - \\
 &\quad (80-80)^2 - (90-80)^2} \\
 &= \sqrt{(289) + (64) + (36) + (25) + (25) + (25) + (25) + (100) + (0) + (100)} \\
 &= \sqrt{689} \\
 &= 25,8457
 \end{aligned}$$

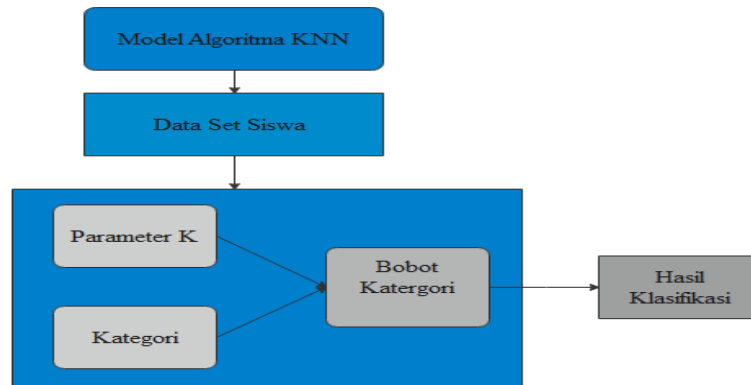
Hasil pengklasifikasikan jenis klasifikasi siswa berprestasi diperoleh setelah melakukan pengurutan jarak dimana jarak terkecil merupakan jarak paling dekat. Data kategori mayoritas yang muncul akan dijadikan kategori terpilih. Hasil klasifikasi tersebut seperti tabel pada tabel berikut.

Tabel 4.5 Hasil Jarak Terdekat

NISN	JARAK	KETERANGAN
2888	24,4949	BERPRESTASI
2808	25,8457	BERPRESTASI
2878	25,33772	BERPRESTASI
2837	19,13113	TIDAK BERPRESTASI
2958	9	TIDAK BERPRESTASI
2916	21,33073	TIDAK BERPRESTASI
2850	26,94439	BERPRESTASI
2957	16,7332	TIDAK BERPRESTASI

4.3 Penerapan *Algorithm K-Nearest Neighbor*

Berikut adalah penerapan *Algorithm K-Nearest Neighbor* :

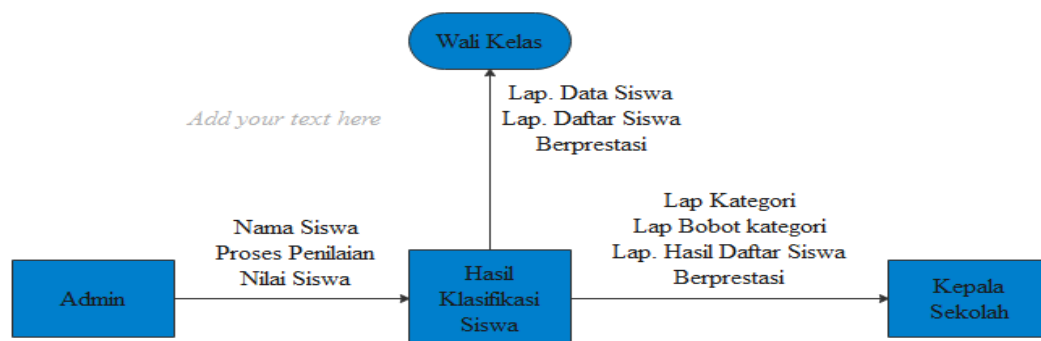


Gambar 4.1 Algoritma K-Nearest Naeghbor

Algoritma *K-Nearest Neighbor* adalah sebuah algoritma untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Kasus khusus dimana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat (dengan kata lain $K=1$) disebut algoritma *K-Nearest Neighbor*.

4.4 Diagram Konteks

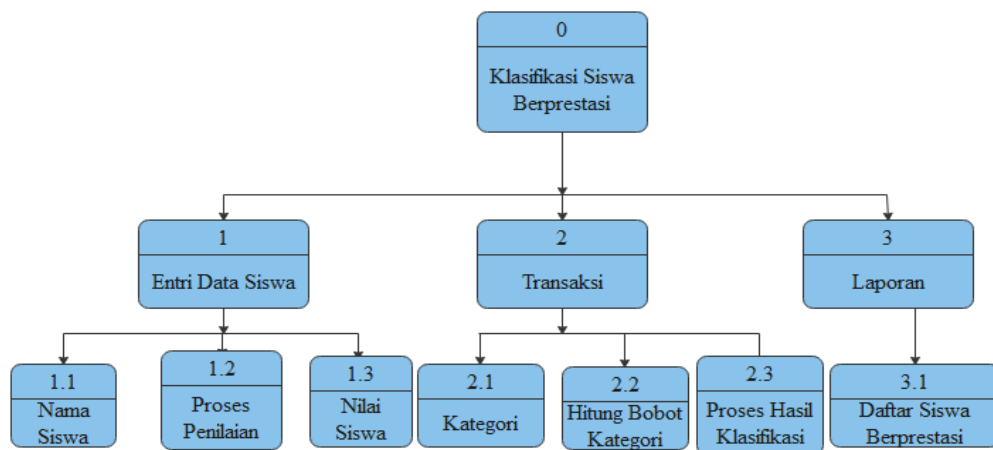
Diagram konteks atau disebut dengan model sistem fundamental merepresenniasikan seluruh elemen sistem sebagai sebuah bubble dengan data input, output yang ditunjukkan anak panah yang masuk dan keluar secara berurutan.



Gambar 4.2 Diagram kontek

4.5 Diagram Berjenjang

Diagram Berjenjang merupakan perancangan suatu sistem yang dapat menampilkan seluruh proses yang terdapat pada suatu sistem tertentu yang jelas dan terstruktur.

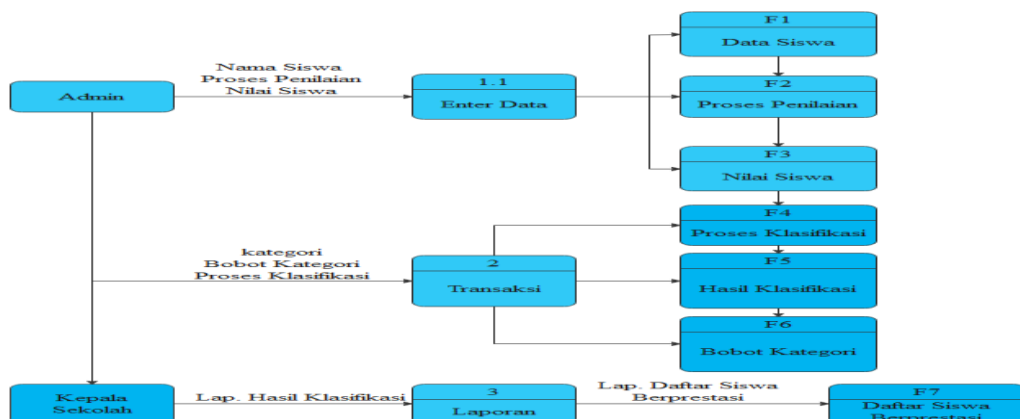


Gambar 4.3 Diagram Berjenjang

4.6. Diagram Arus Data

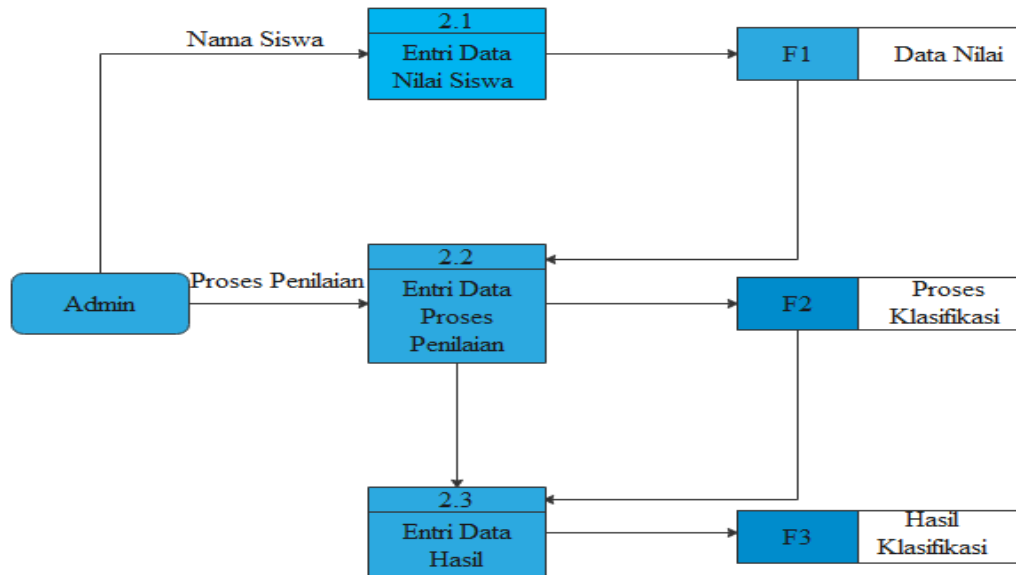
Diagram Arus Data merupakan suatu diagram yang menggambarkan aliran data dari sebuah proses atau sistem.

4.6.1 Diagram Arus Data Level 0



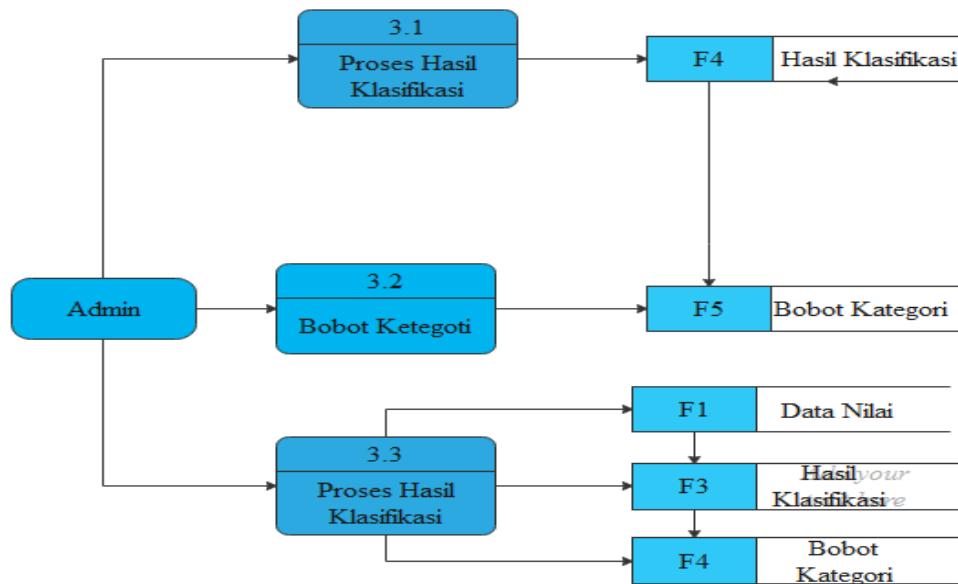
Gambar 4.4 DAD level 0

4.6.2 Diagram Arus Data Level 1



Gambar 4.5 DAD Level 1

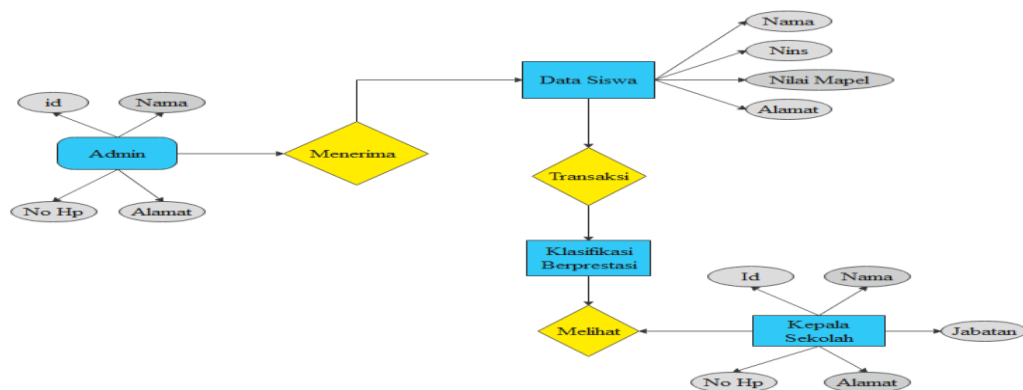
4.6.3 Diagram Arus Data Level 2



Gambar 4.6 DAD Level 2

4.6.4. Entity Relationship Diagram

ERD merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh system analysts dalam tahap analisis persyaratan proyek pengembangan system.



Gambar 4.7 Entity Relationship Diagram

4.7 Kamus Data

Kamus Data atau *Data Dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data digunakan untuk merancang input, file-file database dan output, kamus data dibuat berdasarkan arus yang mengalir pada DAD, dimana didalamnya terdapat struktur dari arus data secara detail.

Tabel 4.6 Kamus Data Penilaian

Kamus Data : Data Penilaian				
Nama Arus Data : Penilaian Penjelasan : Input Data Penilaian Periode : Setiap ada penambahan Data Penilaian				Bentuk Data : a-1, 1.1-f2, f2-2, 2.1-f1, f2- 2.2, 2.3-f3 Dokumen
No	Field Name	Type	Size	
1	Nis	Integer	8	
2	Nama_Siswa	Varchar	25	
3	Nilai	Integer	7	

Tabel 4.7 Data Perhitungan Bobot Kategori

Kamus Data : Data Perhitungan Bobot Kategori				
Nama Arus Data: Hasil Penjelasan : Hasil sistem Periode : Setiap ada penambahan data Proses				Bentuk Data : a- 1.1-f5, f5-2.1-2.2, 2.2-f2, f2-2.3, 2.3- f3 Dokumen
No	Field Name	Type	Size	Ket
1	Nisn	Integer	5	Nomor Induk Siswa
2	Nama_Siswa	Varchar	30	Nama
3	Nilai_mp_1	Integer	5	Nilai mp_1
4	Nilai_mp_2	Integer	5	Nilai mp_2
5	Nilai_mp_3	Integer	5	Nilai mp_3
6	Nilai_mp_4	Integer	5	Nilai mp_4
7	Nilai_mp_5	Integer	5	Nilai mp_5
8	Nilai_mp_6	Integer	5	Nilai mp_6
9	Nilai_mp_7	Integer	5	Nilai mp_7
10	Nilai_mp_8	Integer	5	Nilai mp_8
11	Nilai_mp_9	Integer	5	Nilai mp_9
12	Nilai_mp_10	Integer	5	Nilai mp_10
13	Hasil	Varchar	30	Hasil

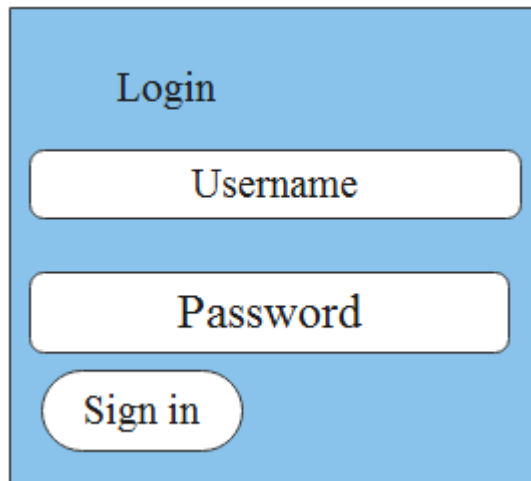
4.8.Interface Design

4.8.1 Mekanisme User

Tabel 4.8 Mekanisme User

User	Kategori	Akses Input	Akses Output
Admin	Wali Kelas	Data Siswa	Data Siswa

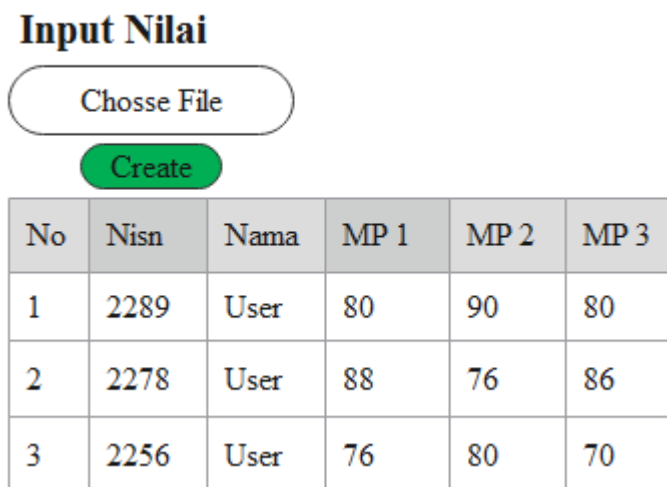
4.8.2 Interface Desain Login



The login interface consists of a light blue rectangular container. At the top center is the text 'Login'. Below it are two white rounded rectangular input fields, the first labeled 'Username' and the second labeled 'Password'. At the bottom center is a white rounded rectangular button labeled 'Sign in'.

Gambar 4.8 Desain Login

4.8.3 Tampilan Input Nilai



The 'Input Nilai' interface features a title 'Input Nilai' at the top. Below the title are two buttons: 'Chosse File' (with a typo) and 'Create'. Below the buttons is a table with 6 columns: 'No', 'Nisn', 'Nama', 'MP 1', 'MP 2', and 'MP 3'. The table contains 3 rows of data.

No	Nisn	Nama	MP 1	MP 2	MP 3
1	2289	User	80	90	80
2	2278	User	88	76	86
3	2256	User	76	80	70

Gambar 4.9 Input Nilai

4.8.4 Tampilan Proses



No	Nisn	Nama	MP 1	MP 2	MP 3
1	2289	User	80	90	80
2	2278	User	88	76	86
3	2256	User	76	80	70

Gambar 4.10 Proses

4.8.5 Tampilan Hasil

HASIL

No	Nisn	Nama	MP 1	MP 2	MP 3	Hasil
1	2289	User	80	90	80	Berprestasi
2	2278	User	88	76	86	Tidak Berprestasi
3	2256	User	76	80	70	Berprestasi

Gambar 4.11 Tampilan Hasil

4.9 Database

Data base adalah kumpulan data yang terorganisir yang umumnya disimpan dan diakses secara elektronik dari suatu sistem komputer.

Table Name	Fields and Data Types
klasifikasi_smp_1.siswa	id_mhs : int(3) nisn : int(8) nama : varchar(40) id_login : int(11) tahun_masuk : int(4)
klasifikasi_smp_1.guru	id : int(11) nipn : int(11) nama : varchar(30) id_login : int(11)
klasifikasi_smp_1.mata_pelajaran	id_mp : int(2) kode_mp : varchar(5) nama_mp : varchar(30)
klasifikasi_smp_1.guru_mp	id_guru_mp : int(2) id_guru : int(3) id_mp : int(5) id_periode : int(4)
klasifikasi_smp_1.login	id_login : int(3) id_priv : int(3) username : varchar(200) password : varchar(200)
klasifikasi_smp_1.data_latih	id : int(11) id_nisn : int(11) id_nama : int(11) mp1 : double mp2 : double mp3 : double mp4 : double mp5 : double mp6 : double mp7 : double mp8 : double mp9 : double mp10 : double
klasifikasi_smp_1.priv	id_priv : int(3) n_priv : varchar(10)
klasifikasi_smp_1.hasil	id : int(11) id_nisn : int(11) id_nama : int(11) mp1 : varchar(200) mp2 : varchar(200) mp3 : varchar(200) mp4 : varchar(200) mp5 : varchar(200) mp6 : varchar(200) mp7 : varchar(200) mp8 : varchar(200) mp9 : varchar(200) mp10 : varchar(200) mp11 : varchar(200) mp12 : varchar(200) mp13 : varchar(200) mp14 : varchar(200) mp15 : varchar(200) hasil : varchar(200)

Gambar 4.12 Data Base

4.10 Desain Struktur Data

4.10.1 Desain Struktur Private

Tabel 4.9 Struktur Private

Nama Arus Data: tbl private				
Type : Transaksi				
Primary Key : Username				
Media : Harddisk				
Fungsi : Merupakan Data Pengguna Aplikasi				
No	Field Name	Type	Size	Ket
1	Username	Varchar	30	Username
2	Password	Varchar	30	Password
3	Kepala Sekolah	Varchar	30	Kepala Sekolah
4	Wali Kelas	Varchar	30	Wali Kelas
5	No_Telpon	Integer	30	No_Telpon

4.10.2 Struktur Data Latih

Tabel 4.10 Data Latih

Nama : tbl data latih Primary Key : data_latih Media : Harddisk Fungsi : Merupakan Data Objek Yang Diteliti				
No	Field Name	Type	Size	Ket
1	Nisn	Int	5	Nomor Induk Siswa
2	Nama_Siswa	Varchar	30	Nama
3	Nilai mp 1	Int	5	Nilai mp 1
4	Nilai mp 2	Int	5	Nilai mp 2
5	Nilai mp 3	Int	5	Nilai mp 3
6	Nilai mp 4	Int	5	Nilai mp 4
7	Nilai mp 5	Int	5	Nilai mp 5
8	Nilai mp 6	Int	5	Nilai mp 6
9	Nilai mp 7	Int	5	Nilai mp 7
10	Nilai mp 8	Int	5	Nilai mp 8
11	Nilai mp 9	Int	5	Nilai mp 9
12	Nilai mp 10	Int	5	Nilai mp 10

4.10.3 Struktur Data Hasil

Tabel 4.11 Struktur Hasil

Nama : Hasil Type : Transaksi Primery Key : Hasil Fungsi : Merupakan Hasil Dari Data Latih				
No	Field Name	Type	Size	Keterangan
1	Nisn	Int	5	Nomor Induk Siswa
2	Nama_Siswa	Varchar	30	Nama
3	Nilai_mp_1	Int	5	Nilai mp_1
4	Nilai_mp_2	Int	5	Nilai mp_2
5	Nilai_mp_3	Int	5	Nilai mp_3
6	Nilai_mp_4	Int	5	Nilai mp_4
7	Nilai_mp_5	Int	5	Nilai mp_5
8	Nilai_mp_6	Int	5	Nilai mp_6
9	Nilai_mp_7	Int	5	Nilai mp_7
10	Nilai_mp_8	Int	5	Nilai mp_8
11	Nilai_mp_9	Int	5	Nilai mp_9
12	Nilai_mp_10	Int	5	Nilai mp_10
13	Hasil	Varchar	30	Hasil

4.11 Arsitektur Sistem

Sistem penerapan Algoritma K-Nearest Neighbor (KNN). Untuk data siswa berprestasi menggunakan Klasifikasi. Sedangkan spesifikasi Hardware dan Software yang direkomendasikan:

1. Processor : Intel Core i3
2. RAM : 4 GB
3. Hardisk : 500 MB
4. OS : Windows 10
5. Tools : Notepad+

4.12 Pengujian White Box

Berikut adalah kode program untuk pengujian pengujian white box sistem klasifikasi siswa berprestasi sebagai berikut:

```
function proses_mining($db_object, $k=5) {  
  
    //hapus table temporary jarak  
  
    $db_object->db_query("TRUNCATE jarak");  
  
    $db_object->db_query("DELETE FROM hasil");  
  
    $sql = "SELECT  
  
        period.`semester`,period.`tahun`,  
  
        mhs.`nisn`, siswa.`nama`,  
  
        latih.*  
  
    FROM  
  
        data_latih latih,  
  
        siswa mhs,
```

```

periode period

WHERE latih.`id_nisn` = siswa.`id_sis`

AND period.`id_periode` = latih.`id_nama`;

$query = $db_object->db_query($sql);

$jarak = array();

$a = 0;

//hitung jarak

echo "Jarak:echo "<table border=1>";

$temp_data_latih = array();

while ($row = $db_object->db_fetch_array($query)) {

    $temp_data_latih[] = $row;

    echo "<tr>";

    $sql1 = "SELECT

        period.`semester`,period.`tahun`,

        mhs.`nisn`, mhs.`nama`,

        latih.*

    FROM

        data_latih latih,

        siswa mhs,

        periode period

    WHERE latih.`id_nisn` = mhs.`id_mhs`

```

```

        AND period.`id_periode` = latih.`id_nama`;

$query1 = $db_object->db_query($sql1);

$b = 0;

while ($row1 = $db_object->db_fetch_array($query1)) {

    $jarak[$a][$b] = absolute($row['mp1'] - $row1['mp1']) +

        absolute($row['mp2'] - $row1['mp2']) +

        absolute($row['mp3'] - $row1['mp3']) +

        absolute($row['mp4'] - $row1['mp4']) +

        absolute($row['mp5'] - $row1['mp5']) +

echo "<table border=1>";

$temp_data_latih = array();

while ($row = $db_object->db_fetch_array($query)) {

    $temp_data_latih[] = $row;

    echo "<tr>";

    $sql1 = "SELECT

        period.`semester`,period.`tahun`,

        mhs.`nispn`, mhs.`nama`,

        latih.*

    FROM

        data_latih latih,

        siswa mhs,
```

```

periode period

WHERE latih.`id_nisn` = mhs.`id_mhs`

AND period.`id_periode` = latih.`id_nama`;

$query1 = $db_object->db_query($sql1);

$b = 0;

while ($row1 = $db_object->db_fetch_array($query1)) {

    $jarak[$a][$b] = absolute($row['mp1'] - $row1['mp1']) +

        absolute($row['mp2'] - $row1['mp2']) +

        absolute($row['mp3'] - $row1['mp3']) +

        absolute($row['mp4'] - $row1['mp4']) +

        absolute($row['mp5'] - $row1['mp5']) + absolute($row['mp6'] -

$row1['mp6']) +

        absolute($row['mp7'] - $row1['mp7']) +

        absolute($row['mp8'] - $row1['mp8']) +

        absolute($row['mp9'] - $row1['mp9']) +

        absolute($row['mp10'] - $row1['mp10']);

    echo "<td>" . $jarak[$a][$b] . "</td>";

    $b++;

}

echo "</tr>";

$a++;}

```



```

        echo "</table>";

//ambil sebanyak k terkecil dari jarak

$a=0;

$jarak_k_terkecil = array();

while ($a < count($jarak)) {

    $jarak_k_terkecil[$a] = get_jarak_terkecil_sebanyak_k($jarak[$a], $k);

    $a++;

}

//hitung average tiap jarak data

$average_tiap_jarak = array();

$a=0;

while ($a < count($jarak_k_terkecil)) {

$average_tiap_jarak[$a] = average($jarak_k_terkecil[$a]);

    $a++;

}

//average from all average...

$average_all_avg_jarak = average($average_tiap_jarak);

//standard deviasi

$std_deviasi1 = array();//pengurangan pangkat 2

$a=0;

while($a < count($average_tiap_jarak)){

```

```

        $std_deviasi1[$a] = pow($saverage_tiap_jarak[$a] -
$average_all_avg_jarak,2);

        $a++;

    }

    $std_deviasi2 = array_sum($std_deviasi1);

    $std_deviasi3 = $std_deviasi2/(count($std_deviasi1)-1);

    $stdDeviasi = sqrt($std_deviasi3);

    $treshold = $average_all_avg_jarak+($stdDeviasi*3);//3 itu apa???

    echo "Treshold = ".price_format($treshold);

    //

    //hitung nilai rata-rata nilai siswa permata pelajaran//

    $sql = "SELECT "

        . "AVG(mp1) AS mp1, "

        . "AVG(mp2) AS mp2, "

        . "AVG(mp3) AS mp3, "

    . "AVG(mp4) AS mp4, "

        . "AVG(mp5) AS mp5, "

        . "AVG(mp6) AS mp6, "

        . "AVG(mp7) AS mp7, "

        . "AVG(mp8) AS mp8, "

        . "AVG(mp9) AS mp9, "

```

```

"AVG(mp10) AS mp10 "

. " FROM data_latih ";

$query = $db_object->db_query($sql);

br();

echo "Rata-rata permata pelajaran:";

echo "<table class='table table-bordered table-striped table-hover'>";

echo "<tr>";

echo
"<td>MP_1</td><td>MP_2</td><td>MP_3</td><td>MP_4</td><td>MP_5</td>
>
<td>MP_6</td><td>MP_7</td><td>MP_8</td><td>MP_9</td><td>MP_10</td>
>";

echo "</tr>";

$row = $db_object->db_fetch_array($query);

$var_average_matkul = $row;

echo "<tr>";

echo "<td>".price_format($row['mp1'])."</td>";

echo "<td>".price_format($row['mp2'])."</td>";

echo "<td>".price_format($row['mp3'])."</td>";

echo "<td>".price_format($row['mp4'])."</td>";

($hasil_mp1=="KURANG")?$kurang++:$lebih++;

($hasil_mp2=="KURANG")?$kurang++:$lebih++;

```

```

hasil_mp3=="KURANG")?$kurang++:$lebih++;

($hasil_mp4=="KURANG")?$kurang++:$lebih++;

($hasil_mp5=="KURANG")?$kurang++:$lebih++;

($hasil_mp6=="KURANG")?$kurang++:$lebih++;

($hasil_mp7=="KURANG")?$kurang++:$lebih++;

($hasil_mp8=="KURANG")?$kurang++:$lebih++;

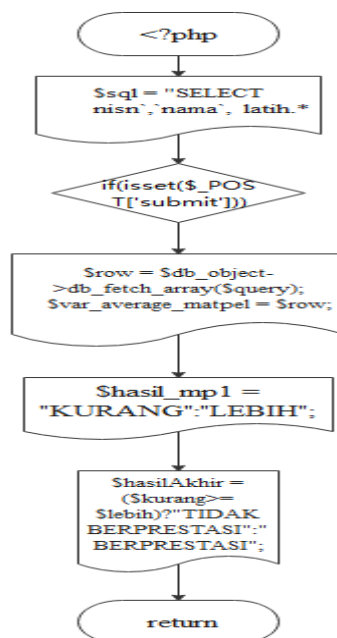
($hasil_mp9=="KURANG")?$kurang++:$lebih++;

($hasil_mp10=="KURANG")?$kurang++:$lebih++;

$hasilAkhir = ($kurang>=$lebih)?"TIDAK
BERPRESTASI":"BERPRESTASI";

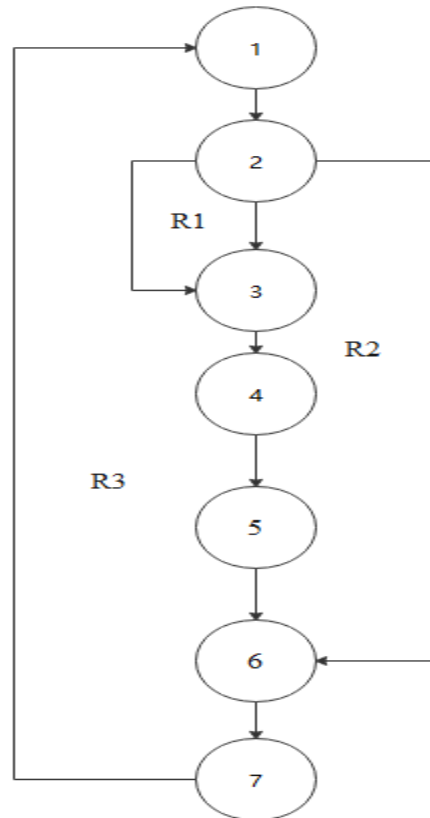
```

4.13 Flowchart Program Untuk Pengujian



Gambar 4.13 Flowchart Pengujian

4.14 Flowgraph Program Untuk Pengujian White Box



Gambar 4.14 Flowgrap Pengujian

4.15 Perhitungan CC Paada Pengujian White Box

Berikut adalah perhitungan hasil dari perhitungan flowgrap sebagai berikut:

Diketahui

Region (R) = 3

Node (N) = 7

Edge (E) = 9

Rumus : $V(G) = (E - N) + 2$

Penyelesaian :

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 9 - 7 + 2 \\ &= 4 \end{aligned}$$

$$\begin{aligned} V(G) &= P + 1 \\ &= 3 + 1 \\ &= 4 \end{aligned}$$

CC = R1, R2, R3

4.16 Path Pada Pengujian White Box

Tabel 4.12 Path Pengujian White Box

No	Path	Keterangan
1	1-2-3.....7	OK
2	1234...7	OK
3	12345..7	OK

4.17 Pengujian Black Box

Pengujian Black Box dilakukan untuk memastikan suatu even atau masukan untuk menjalankan proses yang tepat dan menghasilkan *output* sesuai dengan rancangan.

Tabel 4.13 Pengujian Black box

Input/Even	Fungsi	Hasil Yang Diharapkan	Hasil Uji
Login	Login dengan menginputkan username dan password.	Jika password salah, maka ulangi masukan username dan password.	Sesuai
Menu Home	Menampilkan Halaman Data Sekolah, Input Nilai dan Proses.	Halaman Data Sekolah, Input Nilai dan Proses.	Seusai

Menu Data Sekolah	Menampilkan Halaman Siswa, Guru Mata Pelajaran, dan Pembelajaran.	Menampilkan Halaman Siswa, Guru Mata Pelajaran, dan Pembelajaran.	Sesuai
Menu Input Nilai	Menampilkan halaman tabel data siswa.	Halaman tabel data siswa	Sesuai
Menu Proses	Menampilkan halaman data nilai siswa	Halaman data nilai siswa	Sesuai
Hasil	Menampilkan Halaman Hasil	Halaman Hasil	Sesuai
Logout	Menampilkan Halaman Login	Halaman Login	Sesuai

BAB V

PEMBAHASAN

5.1 Pembahasan Model

5.2 Pembahasan Sistem

Berikut ini adalah hasil pembahasan sistem klasifikasi Siswa Berprestasi Dengan Menggunakan Metode K-nearest Naeghbor di Smp Negeri 1 Kwandang.

5.2.1 Hasil Tampilan Window Sistem



Gambar 5.1 Tampilan Window Sistem

5.2.2. Tampilan Login

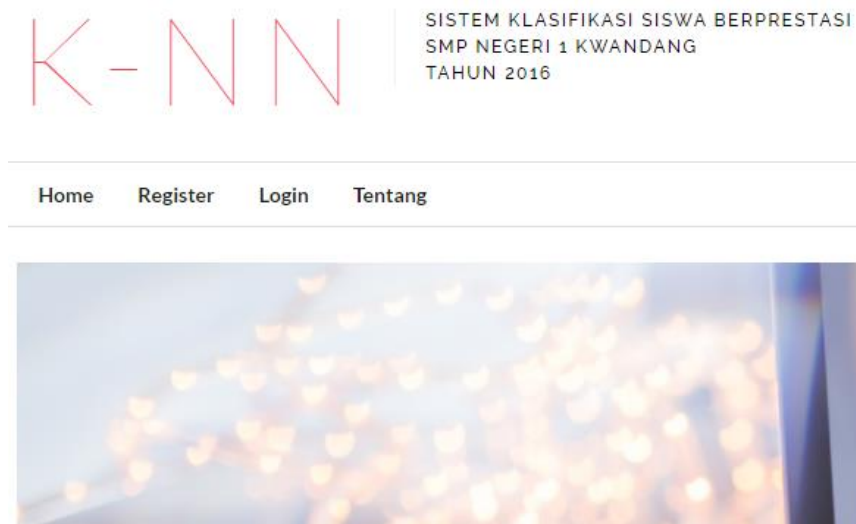
Berikut adalah tampilan login dari system klasifikasi siswa berprestasi sebagai berikut:

Login

Gambar 5.2 Tampilan Login

5.2.3 Tampilan Home

Berikut adalah tampilan home dalam system klasifikasi siswa berprestasi sebagai berikut:



Gambar 5.3 Tampilan Home

5.2.4 Hasil Tampilan Window Input Nilai

Berikut ini adalah tampilan dari input nilai system klasifikasi siswa berprestasi sebagai berikut:

Home Data Sekolah > Input Nilai Proses Tentang Logout

Input Nilai

Periode
-

Import data from excel
 No file chosen

Jumlah data: 85

No	Nisn	Nama	MP_1	MP_2	MP_3	MP_4	MP_5	MP_6	MP_7	MP_8	MP_9	MP_10
1	2888	RAHMAN ISHAK	90	75	75	85	75	80	75	75	86	90
2	2808	AZIZAH NOR KHOMAIRAH	75	87	75	75	95	75	90	75	80	90
3	2878	NUR AYIN VITRIA RAHMAN	75	80	75	76	90	75	85	75	87	75
4	2837	IRENE GABRIELA ROMBE	75	76	70	90	77	78	95	75	78	85

Gambar 5.4 Tampilan Input Nilai

5.2.5 Tampilan Proses

Berikut adalah tampilan dari proses system klasifikasi siswa berprestasi sebagai berikut:

Home Data Sekolah > Input Nilai Proses Tentang Logout

Proses

Jumlah data: 85

No	Nisn	Nama	MP_1	MP_2	MP_3	MP_4	MP_5	MP_6	MP_7	MP_8	MP_9	MP_10
1	2888	RAHMAN ISHAK	90	75	75	85	75	80	75	75	86	90
2	2808	AZIZAH NOR KHOMAIRAH	75	87	75	75	95	75	90	75	80	90
3	2878	NUR AYIN VITRIA RAHMAN	75	80	75	76	90	75	85	75	87	75
4	2837	IRENE GABRIELA ROMBE	75	76	70	90	77	78	95	75	78	85
5	2958	RAFIA OL'I	80	78	75	80	75	85	90	85	80	95

Gambar 5.5 Tampilan Proses

5.2.6 Hasil Tampilan Windows Hasil

Berikut adalah tampilan dari window system dari system klasifikasi siswa berprestasi sebagai berikut:

Hasil:

No	Nisn	Nama	MP_1	MP_2	MP_3	MP_4	MP_5	MP_6	MP_7	MP_8	MP_9	MP_10	Hasil
1	2888	RAHMAN ISHAK	LEBIH	LEBIH	LEBIH	KURANG	LEBIH	LEBIH	LEBIH	LEBIH	LEBIH	LEBIH	BERPRESTASI
2	2808	AZIZAH NOR KHOMAIRAH	LEBIH	LEBIH	KURANG	LEBIH	KURANG	KURANG	KURANG	LEBIH	KURANG	LEBIH	TIDAK BERPRESTASI
3	2878	NUR AYIN VITRIA RAHMAN	KURANG	LEBIH	LEBIH	LEBIH	KURANG	KURANG	KURANG	KURANG	KURANG	KURANG	TIDAK BERPRESTASI
4	2837	IRENE GABRIELA ROMBE	KURANG	KURANG	LEBIH	KURANG	KURANG	KURANG	LEBIH	KURANG	LEBIH	KURANG	TIDAK BERPRESTASI
5	2958	RAFIA OL'I	KURANG	LEBIH	KURANG	KURANG	KURANG	KURANG	KURANG	KURANG	KURANG	KURANG	TIDAK BERPRESTASI
6	2916	SRI WULIS ISHAK	KURANG	KURANG	KURANG	KURANG	KURANG	KURANG	KURANG	KURANG	KURANG	KURANG	TIDAK BERPRESTASI
7	2850	LUTFIAH RAMADHAN KABANGUNAN	KURANG	KURANG	KURANG	KURANG	KURANG	KURANG	KURANG	KURANG	KURANG	KURANG	TIDAK BERPRESTASI

Gambar 5.6 Hasil Tampilan Window Hasil

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Adapun kesimpulan dari hasil penelitian diatas adalah sebagai berikut :

1. Bahwa pada rancangan sistem klasifikasi siswa berprestasi di Smp Negeri 1 Kwandang dengan menggunakan metode *K-Nearest Neighbor* menghasilkan klasifikasi siswa berprestasi. Untuk mempermudah siswa dan guru untuk mengetahui siswa yang berprestasi.
2. Dalam penelitian ini telah diperoleh kinerja dan efektifitas system klasifikasi siswa berprestasi dengan menggunakan metode k-nearest nighbor.

6.2 Saran

Setelah melakukan penelitian ini tentang system klasifikasi siswa berprestasi di smp negeri 1 kwandang dengan menggunakan metode k-nearest nighbor telah ditemukan hal-hal sebagai berikut :

1. Untuk penggunaan variable dari klasifikasi siswa berprestasi perlu diperhentikan nilai variable prioritas seperti Nama, Nisn dan Nilai Matapelajaran untuk diklasifikasikan untuk mengetahui siswa yang memiliki prestasi.
2. Dalam perancangan d ata base diperlukan teknik perekaman data pengkodean merupakan data untuk kemudahan dalam koneksi data base kedalam aplikasi.
3. Penggunaan dataset ini perlu juga diuji coba dengan algoritma yang lain.

DAFTAR PUSTAKA

- Andreas Andoyo Dan Riko Rianto, dengan judul “Pengambilan Keputusan Siswa Berprestasi Pada Smk Mhammadia Pringsewu 2017.”
- Ashar Johar T Delvi Yansoma Dan Kurnia Anggriani dengan judul “Implementasi Metode Knn Dan Simple Additive Weighting (Saw) Dalam Pengambilan Keputusan Seleksi Penerimaan Anggota Paskibrakak 2016”
- Dewi Rosmala Dan Muhammad Ichwan, dengan Judul Komparasi Framework Mvc (Codeigniter, Dan Cake Php Pada Aplikasi Berbasis Wab 2011
- Jodi Irjaya Kartika, dengan judul “Penentuan Siswa Berprestasi Menggunakan Metode K-Nearest Neighbor Dan Weighted Product 2017”
- Jogianto, HM, 2005 dengan judul “Analisis Dan Desain Sisyem Informasi : Pendekatan Terstruktur Teori Dan Praktek Praktek Aplikasi Bisnis, Yogyakarta Andi”
- Moenawar Kholil, 2018 dengan judul “Penerapan Metode K-Knearest Neighbor Dalam Proses Seleksi Penerima Beasiswa Menggunakan Metode Knn”
- Rizaldi, dengan judul “Penerapan Waterfall Dalam Membangun Sistem Informasi Pengolahan Data Pelaksanaan Konstruksi Pembangunan Jalan 2017”
- Rivanda putra pratama dan ira puspitasari, dengan judul “sistem pendukung keputusan siswa berprestasi disekolah menengah pertama, 2017.”
- Program Study S1 Sistem Informasi, Universitas Air Langga*
- Rony Setiawan, dengan judul “Penerapan Data Mining Menggunakan K-Mean Clustering Untuk Menentuksn Strategi Promosi Mahasiswa Baru 2016
- Raymundus Nandi Irawan, Wawan Laksito YS, Sri Siswanti, dengan judul “Sistem Pendukung Keputusan Untuk Menentukan Status Prestasi Siswa Menggunakan Metode Knn 2013”
- Sinta Maria Dan Istiatul Muawanah, dengan judul “Perancangan Sistem Informasi Pengolahan Data 2015
- Ni Luh Gede Pivin Suwirmayanti “Penerapan K-Nearest Naeghbor Untuk Sistem Rekomendasi Pemilihan Mobi”¹ 2017 *Program Studi Sistem Komputer STMK STIKOM Bali.*

Mustakim, Giantika Oktaviani “ Algoritma *K-Nearest Naeghbor Clasifikascion* Sebagai Sistem Prediksi Predikat Prestasi Mahasiswa. 2017.

Universitas Islam Negeri Sultan Syarif Kasim Riau.

Sumarlin “Implementasi Algoritma K-Nearest Naeghbor Sebagai Pendukung Keputusan Klasifikasi Penerima Beasiswa PPA dan BBM. 2015

Arie Yandi Saputra, Yogi Primadasa “Penerapan Teknik Klasifikasi Untuk Prediksi Kelulusan Mahasiswa Menggunakan Algoritma K-Nearest Naeghbor 2018

```

<?php
session_start(); // harus ada di bagian paling atas kode
$path_to_root = "";
include $path_to_root . 'database.php';

//object database class
$db = new database();

$user = strip_tags(trim($_POST['username'])); #echo $user;
$pass = strip_tags(trim($_POST['password'])); #echo $pass;

$sql = get_sql_login_admin_page($user, $pass);

$result = $db->db_query($sql);
$num_rows = $db->db_num_rows($result);

if ($num_rows > 0) {
    $rows = $db->db_fetch_array($result);

    unset($_POST); // hapus post form
    $_SESSION['knn_mhs_prestasi_id'] = $rows['id_login']; // mengisi session
    $_SESSION['knn_mhs_prestasi_username'] = $rows['username'];
    $_SESSION['knn_mhs_prestasi_level'] = $rows['id_priv'];

    $sql_level = get_sql_level($rows['id_priv']);
    $result_1 = $db->db_query($sql_level);
    $row_1 = $db->db_fetch_array($result_1);
    $_SESSION['knn_mhs_prestasi_level_name'] = $row_1['n_priv'];

    $_SESSION['knn_mhs_prestasi_key'] = sha1(date("Y-m-d H:i:s")) .
    $rows['id_login']);

```

```

        $_SESSION['knn_mhs_prestasi_last_login'] = date("d-m-Y H:i:s");
        header("location:index.php");
    } else {
        header("location:login.php?login=1");
    }
}

/**
 * query get login
 * @param string $user username
 * @param string $pass password
 * @return string
 */
function get_sql_login_admin_page($user, $pass){
    $sql = "SELECT * FROM login"
        . " WHERE username = '" . $user . "' AND password = ('" . $pass . "')";
    return $sql;
}

function get_sql_level($id){
    $sql = "SELECT * FROM priv"
        . " WHERE id_priv = '" . $id . "'";
    return $sql;
}
?>

```

```

<?php
/*
 * Class Configuration Database
 */
class database {
    private $servername;
    private $user_db;
    private $password_db;
    private $database;
    private $koneksi;

    /**
     * fungsi konstruktor awal dan koneksi ke database;
     * set variable class $this->conn;
     */
    function database() {
        $this->load_conf_db();
        return $this->connect_db();
    }
    function load_conf_db() {
        $path = dirname(__FILE__) . '/koneksi.php';
        if (file_exists($path)) {
            $conf = include $path;
            $this->servername = @$conf['host'];
            $this->database = @$conf['dbname'];
            $this->user_db = @$conf['username'];
            $this->password_db = @$conf['password'];
        }
    }
}

```

```

}

/**
 * connection database
 * @return type
 */
public function connect_db() {
    $this->koneksi = mysqli_connect($this->servername, $this->user_db, $this-
>password_db, $this->database);
    return $this->koneksi;
}

/**
 * execute query mysql_query
 * @param string $sql
 * @return type
 */
function db_query($sql) {
    $conn = $this->koneksi;
    return mysqli_query($conn, $sql);
}

/**
 * mysql_error
 * @return type
 */
function db_error($result) {
    $conn = $this->koneksi;
    return mysqli_error($conn);
}

/**
 * mysql_fetch_array

```

```

* @param type $result
* @return type
*/
function db_fetch_array($result) {
    return mysqli_fetch_array($result);
}
/**
* mysql_num_rows
* @param type $result
* @return type
*/
function db_num_rows($result) {
    return mysqli_num_rows($result);
}
/**
* mysql_insert_id
* @return type
*/
function db_insert_id() {
    $conn = $this->koneksi;
    return mysqli_insert_id($conn);
}

//=====================================================
=====

/**
* Insert Data within table by accepting TableName and Table column => Data
as associative array
* contoh $val_cols = array("Name"=>'ido',"Age"=>23);
* @param string $table
* @param array $val_cols
* @return type hasil query (db_query)

```

```

*/
function insert_record($table, array $val_cols) {
    $field = implode("`", `", array_keys($val_cols));
    $i = 0;
    foreach ($val_cols as $key => $value) {
        $StValue[$i] = "" . $value . "";
        $i++;
    }
    $StValues = implode(" ", $StValue);
    $sql = "INSERT INTO $table (`$field`) VALUES ($StValues)";
    $result = $this->db_query($sql);
    return $result;
}
/**

```

* Delete data form table; Accepting Table Name and Keys=>Values as associative array

```

* contoh $val_cols = array("id"=>4,"Name"=>'ido');
* @param string $table
* @param array $val_cols where condition implode by AND
* @return type hasil query (db_query)
*/

```

```

function delete_record($table, array $val_cols) {
    //Append each element of val_cols associative array
    $i = 0;
    foreach ($val_cols as $key => $value) {
        $exp[$i] = $key . " = " . $value . "";
        $i++;
    }
    $Stexp = implode(" AND ", $exp);
    $sql = "DELETE FROM $table WHERE $Stexp ";
    return $this->db_query($sql);
}

```

```
}
```

```
/**
```

```
 * Update data within table; Accepting Table Name and Keys=>Values as  
associative array
```

```
 * contoh $set_val_cols = array("Name"=>'Ido',"Nilai"=>98.9);
```

```
 * contoh $cod_val_cols = array("id"=>'23');
```

```
 * @param type $table
```

```
 * @param array $set_val_cols set update
```

```
 * @param array $cod_val_cols where condition implode by AND
```

```
 * @return type
```

```
 */
```

```
function update_record($table, array $set_val_cols, array $cod_val_cols) {
```

```
    //append set_val_cols associative array elements
```

```
    $i = 0;
```

```
    foreach ($set_val_cols as $key => $value) {
```

```
        $set[$i] = $key . " = " . $value . "";
```

```
        $i++;
```

```
    }
```

```
    $Stset = implode(", ", $set);
```

```
    //append cod_val_cols associative array elements
```

```
    $i = 0;
```

```
    foreach ($cod_val_cols as $key => $value) {
```

```
        $cod[$i] = $key . " = " . $value . "";
```

```
        $i++;
```

```
    }
```

```
    $Stcod = implode(" AND ", $cod);
```

```
    $sql = "UPDATE $table SET $Stset WHERE $Stcod";
```

```
    return $this->db_query($sql);
```

```
}
```

```
//-----
/**
 * menghitung jumlah data (count)
 * @param type $table
 * @param type $field
 * @param type $where default=null, diisi dengan kolom dan isi nya contoh
warna='blue'
 * @return int hasil count (db_fetch_array)
 */
function count_data($table, $field, $where = null) {
    $sql = "SELECT COUNT($field) FROM $table ";
    if ($where != null || $where != "") {
        $sql .= " WHERE $where ";
    }
    $result = $this->db_query($sql);
    return $row = $this->db_fetch_array($result);
}

/**
 * ambil data table semua kolom return bisa db_query/db_fetch_array
 * bisa di limit
 * @param type $table
 * @param type $where default=null
 * @param type $fetch default=false
 * @param type $limit default=false
 * @param type $limit_posisi default=0
 * @param type $limit_batas default=0
 * @param type $sort default=", order by..."
 * @return fetch/db_query

```

```
*/
```

```
function display_table_all_column($table, $where = null, $fetch = false, $limit  
= false, $limit_posisi = 0, $limit_batas = 0, $sort = "") {
```

```
    $sql = "SELECT * FROM $table ";
```

```
    if ($where != null || $where != "") {
```

```
        $sql .= " WHERE $where";
```

```
    }
```

```
    if (!empty($sort)) {
```

```
        $sql .= " ORDER BY " . $sort;
```

```
    }
```

```
    if ($limit) {
```

```
        $sql .= " LIMIT $limit_posisi , $limit_batas ";
```

```
    }
```

```
    if ($fetch) {
```

```
        $result = $this->db_query($sql);
```

```
        return $row = $this->db_fetch_array($result);
```

```
    } else {
```

```
        return $result = $this->db_query($sql);
```

```
    }
```

```
}
```

```
/**
```

```
 * mencari sesuatu dalam sebuah table
```

```
 *
```

```
 * @param string $table
```

```
 * @param array string $find_column default = array()
```

```
 * @param string $where default=""
```

```
 * @return array db_fetch_array
```

```
 */
```

```
function find_in_table($table, $find_column = array(), $where = "") {
```

```
    $sql = "SELECT ";
```

```

        $column = "";
        if (!is_array($find_column)) {
            $column = $find_column;
        } else {
            $column = implode(",", $find_column);
        }
        $sql .= $column . " FROM " . $table . " " . $where;
        $result = $this->db_query($sql);
        $rows = $this->db_fetch_array($result);
        return $rows;
    }
}

/**
 * cek value data pada suatu table
 * @param string $table
 * @param string $field
 * @param string $value
 * @return boolean true jika ada , false jika tidak ada
 */

function cek_data_is_in_table($table, $field, $value) {
    $sql = "SELECT COUNT(" . $field . ") FROM " . $table . " WHERE " .
$field . " = '" . $value . "'";
    $result = $this->db_query($sql);
    $num = $this->db_fetch_array($result);
    if ($num[0] > 0) {
        return true;
    } else {
        return false;
    }
}

/**
 * ambil data login user
 * @param type $id_login

```

```
* @return type
*/
function get_login_by_id($id_login){
    $sql = "SELECT * FROM login WHERE id_login = ".$id_login;
    $result = $this->db_query($sql);
    $row = $this->db_fetch_array($result);
    return $row;
}
}
?>
```

```

<?php
//session_start();
if (!isset($_SESSION['knn_mhs_prestasi_id'])) {
    header("location:index.php?menu=forbidden");
}
include_once "database.php";
include_once "fungsi.php";
include_once "import/excel_reader2.php";
?>

<section class="page_head">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 col-md-12 col-sm-12">
                <div class="page_title">
                    <h2>Input Nilai</h2>
                </div>
            </div>
        </div>
    </div>
</section>

<?php
//object database class
$db_object = new database();

$pesan_error = $pesan_success = "";
if(isset($_GET['pesan_error'])){
    $pesan_error = $_GET['pesan_error'];
}

```

```

if(isset($_GET['pesan_success'])){
    $pesan_success = $_GET['pesan_success'];
}
if(isset($_POST['submit'])){
    // if(!$input_error){
    $data = new Spreadsheet_Excel_Reader($_FILES['userfile']['tmp_name']);
    $baris = $data->rowcount($sheet_index=0);
    $column = $data->colcount($sheet_index=0);
    //import data excel dari baris kedua, karena baris pertama adalah nama kolom
    for ($i=2; $i<=$baris; $i++) {
        for($c=1; $c<=$column; $c++){
            $value[$c] = $data->val($i, $c);
        }
        $table = "data_latih";
        $field_value = array("id_nisn"=>$value[1],
            "id_nama"=>$_POST['periode'],
            "mp1"=>$value[2],
            "mp2"=>$value[3],
            "mp3"=>$value[4],
            "mp4"=>$value[5],
            "mp5"=>$value[6],
            "mp6"=>$value[7],
            "mp7"=>$value[8],
            "mp8"=>$value[9],
            "mp9"=>$value[10],
            "mp10"=>$value[11]);
        $query = $db_object->insert_record($table, $field_value);
    }
    ?>
    <script> location.replace("?menu=input_nilai&pesan_success=Data berhasil
    disimpan"); </script>

```

```

        <?php
    }

    if(isset($_GET['delete'])){
        $id_delete=$_GET['delete'];
        $input_error = 0;
        //CEK EXISTING DATA IN TABLE
        $cek_exist = $db_object->count_data("peserta_mp",
            'id_dosen_mk',
            "id_dosen_mk='".$_id_delete.'"");
        if ($cek_exist[0]>0) {
            $input_error = 1;
            ?>
            <script> location.replace("?menu=pembelajaran&pesan_error=Data masih
digunakan"); </script>
            <?php
        }
        if(!$input_error){
            //delete
            $sql_del = "DELETE FROM guru_mp WHERE id_guru_mp = '$id_delete'";
            $tr=$db_object->db_query($sql_del);
            ?>
            <script> location.replace("?menu=pembelajaran&pesan_success=berhasil
delete data"); </script>
            <?php
        }
    }

    $sql = "SELECT
        period.`semester`,period.`tahun`,
        mhs.`nisd`, mhs.`nama`,
        latih.*

```

```

FROM
    data_latih latihan,
    siswa mhs,
    periode period
WHERE latihan.`id_nisn` = mhs.`id_mhs`
AND period.`id_periode` = latihan.`id_nama`;
$query=$db_object->db_query($sql);
$jumlah=$db_object->db_num_rows($query);
?>
<div class="super_sub_content">
    <div class="container">
        <div class="row">
            <!--UPLOAD EXCEL FORM-->
            <form method="post" enctype="multipart/form-data" action="">
                <div class="form-group">
                    <div class="input-group">
                        <label>Periode</label>
                        <?php
                            list_periode("periode", "", true, true);
                        ?>
                    </div>
                </div>
                </div>
                <div class="form-group">
                    <div class="input-group">
                        <label>Import data from excel</label>
                        <input name="userfile" type="file" class="form-control">
                    </div>
                </div>
                <div class="form-group">
                    <input name="submit" type="submit" value="Create" class="btn btn-
success">

```

```

        </div>
    </form>

    <?php
    if (!empty($pesan_error)) {
        display_error($pesan_error);
    }
    if (!empty($pesan_success)) {
        display_success($pesan_success);
    }
    echo "Jumlah data: ".$jumlah."<br>";
    if($jumlah==0){
        echo "Data kosong...";
    }
    else{
    ?>
    <table class='table table-bordered table-striped table-hover'>
        <tr>
            <th>No</th>
            <th>Nisn</th>
            <th>Nama</th>
            <th>MP_1</th>
            <th>MP_2</th>
            <th>MP_3</th>
            <th>MP_4</th>
            <th>MP_5</th>
            <th>MP_6</th>
            <th>MP_7</th>
            <th>MP_8</th>
            <th>MP_9</th>
            <th>MP_10</th>

```

```

        </tr>
    <?php
        $no=1;
        while($row=$db_object->db_fetch_array($query)){
            echo "<tr>";
                echo "<td>".$no."</td>";
                echo "<td>".$row['nisi']."</td>";
                echo "<td>".$row['nama']."</td>";
                echo "<td>".$row['mp1']."</td>";
                echo "<td>".$row['mp2']."</td>";
                echo "<td>".$row['mp3']."</td>";
                echo "<td>".$row['mp4']."</td>";
                echo "<td>".$row['mp5']."</td>";
                echo "<td>".$row['mp6']."</td>";
                echo "<td>".$row['mp7']."</td>";
                echo "<td>".$row['mp8']."</td>";
                echo "<td>".$row['mp9']."</td>";
                echo "<td>".$row['mp10']."</td>";
            echo "</tr>";
            $no++;
        }
    ?>
</table>
<?php
}
?>
</div>
</div>
</div>

```

```
<?php
```

```
function proses_mining($db_object, $k=5) {  
    //hapus table temporary jarak  
    $db_object->db_query("TRUNCATE jarak");  
    $db_object->db_query("DELETE FROM hasil");  
    $sql = "SELECT  
        period.`semester`,period.`tahun`,  
        mhs.`nisd`, mhs.`nama`,  
        latih.*  
    FROM  
        data_latih latih,  
        siswa mhs,  
        periode period  
    WHERE latih.`id_nisd` = mhs.`id_mhs`  
        AND period.`id_periode` = latih.`id_nama`";  
    $query = $db_object->db_query($sql);  
    $jarak = array();  
    $a = 0;  
    //hitung jarak  
    echo "Jarak: Mata Pelajaran";  
    echo "<table border=1>";  
    $temp_data_latih = array();  
    while ($row = $db_object->db_fetch_array($query)) {  
        $temp_data_latih[] = $row;  
        echo "<tr>";  
        $sql1 = "SELECT  
            period.`semester`,period.`tahun`,
```

```

mhs.`nisn`, mhs.`nama`,
    latih.*

FROM
    data_latih latih,
    siswa mhs,
    periode period
WHERE latih.`id_nisn` = mhs.`id_mhs`
    AND period.`id_periode` = latih.`id_nama`;
$query1 = $db_object->db_query($sql1);
$b = 0;
while ($row1 = $db_object->db_fetch_array($query1)) {
    $jarak[$a][$b] = absolute($row['mp1'] - $row1['mp1']) +
        absolute($row['mp2'] - $row1['mp2']) +
        absolute($row['mp3'] - $row1['mp3']) +
        absolute($row['mp4'] - $row1['mp4']) +
        absolute($row['mp5'] - $row1['mp5']) +
        absolute($row['mp6'] - $row1['mp6']) +
        absolute($row['mp7'] - $row1['mp7']) +
        absolute($row['mp8'] - $row1['mp8']) +
        absolute($row['mp9'] - $row1['mp9']) +
        absolute($row['mp10'] - $row1['mp10']);
    echo "<td>" . $jarak[$a][$b] . "</td>";
    $b++;
}
echo "</tr>";
$a++;
}
echo "</table>";

//ambil sebanyak k terkecil dari jarak

```

```

$a=0;
$jarak_k_terkecil = array();
while ($a < count($jarak)) {
    $jarak_k_terkecil[$a] = get_jarak_terkecil_sebanyak_k($jarak[$a], $k);
    $a++;
}
//hitung average tiap jarak data
$average_tiap_jarak = array();
$a=0;
while ($a < count($jarak_k_terkecil)) {
    $average_tiap_jarak[$a] = average($jarak_k_terkecil[$a]);
    $a++;
}
//average from all average...
$average_all_avg_jarak = average($average_tiap_jarak);
//standard deviasi
$std_deviasi1 = array();//pengurangan pangkat 2
$a=0;
while($a < count($average_tiap_jarak)){
    $std_deviasi1[$a] = pow($average_tiap_jarak[$a] -
$average_all_avg_jarak,2);
    $a++;
}
$std_deviasi2 = array_sum($std_deviasi1);
$std_deviasi3 = $std_deviasi2/(count($std_deviasi1)-1);
$stdDeviasi = sqrt($std_deviasi3);
$treshold = $average_all_avg_jarak+($stdDeviasi*3);//3 itu apa???
echo "Treshold = ".price_format($treshold);
//
//hitung nilai rata-rata nilai siswa permata pelajaran//
$sql = "SELECT "

```

```

        . "AVG(mp1) AS mp1, "
        . "AVG(mp2) AS mp2, "
        . "AVG(mp3) AS mp3, "
        . "AVG(mp4) AS mp4, "
        . "AVG(mp5) AS mp5, "
        . "AVG(mp6) AS mp6, "
        . "AVG(mp7) AS mp7, "
        . "AVG(mp8) AS mp8, "
        . "AVG(mp9) AS mp9, "
        . "AVG(mp10) AS mp10 "
        . " FROM data_latih ";

$query = $db_object->db_query($sql);
br();
echo "Rata-rata permata pelajaran:";
echo "<table class='table table-bordered table-striped table-hover'>";
echo "<tr>";
    echo
    "<td>MP_1</td><td>MP_2</td><td>MP_3</td><td>MP_4</td><td>MP_5</td>
    <td>MP_6</td><td>MP_7</td><td>MP_8</td><td>MP_9</td><td>MP_10</td>
    >";
    echo "</tr>";
    $row = $db_object->db_fetch_array($query);
    $var_average_matkul = $row;
    echo "<tr>";
        echo "<td>".price_format($row['mp1'])."</td>";
        echo "<td>".price_format($row['mp2'])."</td>";
        echo "<td>".price_format($row['mp3'])."</td>";
        echo "<td>".price_format($row['mp4'])."</td>";
        echo "<td>".price_format($row['mp5'])."</td>";
        echo "<td>".price_format($row['mp6'])."</td>";

```

```

        echo "<td>".price_format($row['mp7'])."</td>";
        echo "<td>".price_format($row['mp8'])."</td>";
        echo "<td>".price_format($row['mp9'])."</td>";
        echo "<td>".price_format($row['mp10'])."</td>";
    echo "</tr>";
echo "</table>";
foreach ($temp_data_latih as $key => $value) {
    $kurang = $lebih = 0;
    $hasil_mp1 =
($value['mp1']<=$var_average_matkul['mp1'])?"KURANG":"LEBIH";
    $hasil_mp2 =
($value['mp2']<=$var_average_matkul['mp2'])?"KURANG":"LEBIH";
    $hasil_mp3 =
($value['mp3']<=$var_average_matkul['mp3'])?"KURANG":"LEBIH";
    $hasil_mp4 =
($value['mp4']<=$var_average_matkul['mp4'])?"KURANG":"LEBIH";
    $hasil_mp5 =
($value['mp5']<=$var_average_matkul['mp5'])?"KURANG":"LEBIH";
    $hasil_mp6 =
($value['mp6']<=$var_average_matkul['mp6'])?"KURANG":"LEBIH";
    $hasil_mp7 =
($value['mp7']<=$var_average_matkul['mp7'])?"KURANG":"LEBIH";
    $hasil_mp8 =
($value['mp8']<=$var_average_matkul['mp8'])?"KURANG":"LEBIH";
    $hasil_mp9 =
($value['mp9']<=$var_average_matkul['mp9'])?"KURANG":"LEBIH";
    $hasil_mp10 =
($value['mp10']<=$var_average_matkul['mp10'])?"KURANG":"LEBIH";
    ($hasil_mp1=="KURANG")?$kurang++:$lebih++;
    ($hasil_mp2=="KURANG")?$kurang++:$lebih++;
    ($hasil_mp3=="KURANG")?$kurang++:$lebih++;

```

```

($hasil_mp4=="KURANG")?$kurang++:$lebih++;
($hasil_mp5=="KURANG")?$kurang++:$lebih++;
($hasil_mp6=="KURANG")?$kurang++:$lebih++;
($hasil_mp7=="KURANG")?$kurang++:$lebih++;
($hasil_mp8=="KURANG")?$kurang++:$lebih++;
($hasil_mp9=="KURANG")?$kurang++:$lebih++;
($hasil_mp10=="KURANG")?$kurang++:$lebih++;
$hasilAkhir = ($kurang>=$lebih)?"TIDAK

```

```
BERPRESTASI":"BERPRESTASI";
```

```
$table = "hasil";
```

```
$field_value = array("id_nisn"=>$value['id_nisn'],
```

```
    "id_nama"=>$value['id_nama'],
```

```
    "mp1"=>$hasil_mp1,
```

```
    "mp2"=>$hasil_mp2,
```

```
    "mp3"=>$hasil_mp3,
```

```
    "mp4"=>$hasil_mp4,
```

```
    "mp5"=>$hasil_mp5,
```

```
    "mp6"=>$hasil_mp6,
```

```
    "mp7"=>$hasil_mp7,
```

```
    "mp8"=>$hasil_mp8,
```

```
    "mp9"=>$hasil_mp9,
```

```
    "mp10"=>$hasil_mp10,
```

```
    "hasil"=>$hasilAkhir
```

```
);
```

```
$query = $db_object->insert_record($table, $field_value);
```

```
}
```

```
$sql = "SELECT
```

```
    period.`semester`,period.`tahun`,
```

```
    mhs.`nisn`, mhs.`nama`,
```

```
    hasil.*
```

```
FROM
```

```

        hasil hasil,
        siswa mhs,
        periode period
    WHERE hasil.`id_nisn` = mhs.`id_mhs`
    AND period.`id_periode` = hasil.`id_nama`;
$query = $db_object->db_query($sql);
br();
echo "Hasil:";
echo "<table class='table table-bordered table-striped table-hover'>";
echo "<tr>";
    echo "<th>No</th><th>Nisn</th><th>Nama</th>";
    <th>MP_1</th><th>MP_2</th><th>MP_3</th><th>MP_4</th><th>MP_5</th>
    <th>MP_6</th><th>MP_7</th><th>MP_8</th><th>MP_9</th><th>MP_10</th>
    <th>Hasil</th>";
echo "</tr>";
$no=1;
while ($row = $db_object->db_fetch_array($query)) {
    echo "<tr>";
        echo "<td>$no</td>";
        echo "<td>".$row['nisn'].</td>";
        echo "<td>".$row['nama'].</td>";
        echo "<td>".$row['mp1'].</td>";
        echo "<td>".$row['mp2'].</td>";
        echo "<td>".$row['mp3'].</td>";
        echo "<td>".$row['mp4'].</td>";
        echo "<td>".$row['mp5'].</td>";
        echo "<td>".$row['mp6'].</td>";
        echo "<td>".$row['mp7'].</td>";
        echo "<td>".$row['mp8'].</td>";
        echo "<td>".$row['mp9'].</td>";

```

```

        echo "<td>".$row['mp10']."</td>";
        echo "<td>".$row['hasil']."</td>";
    echo "</tr>";
    $no++;
}
echo "</table>";
}

function absolute($nilai) {
    return ($nilai < 0) ? ($nilai * -1) : ($nilai);
}

function get_jarak_terkecil_sebanyak_k($data, $k){
    $return = array();
    $x=0;
    while($x<$k){
        $min = min($data);
        if($min>0){
            $return[] = $min;
            $x++;
        }
        //$data = array_diff($data, [$min]);
        if(($key = array_search($min, $data)) !== false) {
            unset($data[$key]);
        }
    }
    return $return;
}

function average($nilai){
    $a = array_sum($nilai);
    $b = count($nilai);
    return $a/$b;
}

```



DAFTAR RIWAYAT HIDUP

- | | |
|--------------------------|--|
| 1. Nama | : Muhamad Reynaldi |
| 2. Nim | : T3116310 |
| 3. Tempat, Tanggal Lahir | : Parigi, 07 November 1997 |
| 4. Agama | : Islam |
| 5. Email | : naldyspring@gmail.com |

Riwayat pendidikan :

1. Tahun 2010, Menyelesaikan Pendidikan di Sekolah Dasar Negeri 2 Kwandang, Kecamatan Kwandang, Kabupaten Gorontalo Utara, Provinsi Gorontalo.
2. Tahun 2013, Menyelesaikan Pendidikan Di Sekolah Menengah Pertama Negeri 1 Kwandang, Kecamatan Kwandang, Kabupaten Gorontalo Utara, Provinsi Gorontalo.
3. Tahun 2016, Menyelesaikan Pendidikan Di Sekolah Menengah Atas 1 Gorontalo Utara, Kecamatan Kwandang, Kabupaten Gorontalo Utara, Provinsi Gorontalo.
4. Tahun 2016, Telah di terima menjadi Mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo.



PEMERINTAH KABUPATEN GORONTALO UTARA
DINAS PENDIDIKAN
SMP NEGERI 1 KWANDANG

Jln. Siswa No. 575 Desa Cisadane Kec. Kwandang Kab. Gorontalo (0442) 3110534 Kp. 96252



SURAT KETERANGAN

No. 421 / SMPN.1 / KWD / 208 / XI / 2020

Yang bertanda tangan di bawah ini :

Nama : SAHARA SARAH ZEES, S. Pd, M. Pd
NIP : 196306071985022001
Pangkat / Gol : Pembina Tk. I / IV.b
Jabatan : Kepala Sekolah

Dengan ini menerangkan bahwa :

Nama : MUHAMAD REYNALDI
NIM : T3116310
Program Pendidikan : Sarjana (S1)
Program Studi : Teknik Informatika

Bahwa yang bersangkutan telah selesai melaksanakan kegiatan penelitian tentang **"KLASIFIKASI SISWA BERPRESTASI DENGAN METODE K-NEAREST NEIGHBOR (KNN)"**, yang dilaksanakan dari tanggal 16 November 2020 s/d 18 November 2020 di SMP Negeri 1 Kwandang.

Demikian Surat Keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Gorontalo Utara, 18 November 2020

Kepala Sekolah


SAHARA SARAH ZEES / S. Pd, M. Pd
NIP. 196306071985022001



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS ICHSAN
(UNISAN) GORONTALO

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmed Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT REKOMENDASI BEBAS PLAGIASI

No. 0700/UNISAN-G/S-BP/XII/2020

Yang bertanda tangan di bawah ini:

Nama : Sunarto Taliki, M.Kom
NIDN : 0906058301
Unit Kerja : Postikom, Universitas Ichsan Gorontalo

Dengan ini Menyatakan bahwa :

Nama Mahasiswa : MUHAMAD REYNALDI
NIM : T31 f6310
Program Studi : Teknik Informatika (S1)
Fakultas : Fakultas Ilmu Komputer
Judul Skripsi : KLASIFIKASI SISWA BERPRESTASI DENGAN
METODE K-NEAREST NEIGHBOR (KNN)

Sesuai dengan hasil pengecekan tingkat kemiripan skripsi melalui aplikasi Turnitin untuk judul skripsi di atas diperoleh hasil Similarity sebesar 31%, berdasarkan SK Rektor No. 237/UNISAN-G/SK/X/2019 tentang Panduan Pencegahan dan Penanggulangan Plagiarisme, bahwa batas kemiripan skripsi maksimal 35% dan sesuai dengan Surat Pernyataan dari kedua Pembimbing yang bersangkutan menyatakan bahwa isi softcopy skripsi yang diolah di Turnitin SAMA ISINYA dengan Skripsi Aslinya serta format penulisannya sudah sesuai dengan Buku Panduan Penulisan Skripsi, untuk itu skripsi tersebut di atas dinyatakan BEBAS PLAGIASI dan layak untuk diujikan.

Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Gorontalo, 05 Desember 2020

Tim Verifikasi,



Sunarto Taliki, M.Kom
NIDN. 0906058301

Tembusan :

1. Dekan
2. Ketua Program Studi
3. Pembimbing I dan Pembimbing II
4. Yang bersangkutan
5. Arsip