

**KLASIFIKASI PENDERITA PENYAKIT DIABETES BERBASIS DATA
REKAM MEDIS MENGGUNAKAN *SUPPORT VECTOR MACHINE* (SVM)**

SKRIPSI

**Oleh:
ARYA ABIMANYU
NIM. 17650059**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

**KLASIFIKASI PENDERITA PENYAKIT DIABETES BERBASIS DATA
REKAM MEDIS MENGGUNAKAN *SUPPORT VECTOR MACHINE* (SVM)**

SKRIPSI

**Oleh :
ARYA ABIMANYU
NIM. 17650059**

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

HALAMAN PERSETUJUAN

**KLASIFIKASI PENDERITA PENYAKIT DIABETES BERBASIS DATA
REKAM MEDIS MENGGUNAKAN *SUPPORT VECTOR MACHINE* (SVM)**

SKRIPSI

Oleh:
ARYA ABIMANYU
NIM. 17650059

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal: 6 Juni 2023

Dosen Pembimbing I

Dosen Pembimbing II


Dr. Irwan Budi Santoso, M.Kom
NIP. 19770103 201101 1 004


Fatchurrochman, M.Kom
NIP. 19700731 200501 1 002

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Fatchri Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

KLASIFIKASI PENDERITA PENYAKIT DIABETES BERBASIS DATA
REKAM MEDIS MENGGUNAKAN *SUPPORT VECTOR MACHINE* (SVM)

SKRIPSI

Oleh :
ARYA ABIMANYU
NIM. 17650059

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 12 Juni 2023

Susunan Dewan Penguji

Ketua Penguji	: <u>Dr. M. Amin Hariyadi, M.T</u> NIP. 19670018 200501 1 001	()
Anggota Penguji I	: <u>A'la Syaqui, M.Kom</u> NIP. 19771201 200801 1 007	()
Anggota Penguji II	: <u>Dr. Irwan Budi Santoso, M.Kom</u> NIP. 19770103 201101 1 004	()
Anggota Penguji III	: <u>Fatchurrochman, M.Kom</u> NIP. 19700731 200501 1 002	()

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Fatchurrochman, M.MT, IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Arya Abimanyu

NIM : 17650059

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : KLASIFIKASI PENDERITA PENYAKIT DIABETES
BERBASIS DATA REKAM MEDIS MENGGUNAKAN
SUPPORT VECTOR MACHINE (SVM)

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 26 Juni 2023

Yang membuat pernyataan,



Arya Abimanyu
NIM.17650059

MOTTO

“God has perfect timing, never early, never late. It takes a little patience and it takes a lot of faith, but it’s worth the wait”

HALAMAN PERSEMBAHAN

Puji syukur kehadiran Allah SWT, shalawat dan salam bagi Rasul-Nya

Penulis persembahkan sebuah karya ini kepada:

Keluarga penulis yang sangat dicintai, Bapak Siswaya, Ibu Nur Sri Solikhah dan Adik-adik penulis Ratih Puspitasari, Gendhis Ayu Cahyaningrum yang selalu memberikan motivasi, dukungan serta doa kepada penulis dan menjadi alasan utama penulis untuk menyelesaikan skripsi ini

Dosen pembimbing penulis, Bapak Dr. Irwan Budi Santoso, M.Kom dan Bapak Fatchurrochman, M.Kom yang senantiasa sabar dalam memberikan bimbingan dan masukan dalam skripsi ini.

Seluruh dosen Teknik Informatika UIN Maulana Malik Ibrahim Malang, serta seluruh guru-guru penulis yang telah membimbing dan memberikan ilmu yang amat bermanfaat bagi penulis sampai sekarang.

Orang-orang yang penulis sayangi, yang tak bisa penulis sebut satu per satu yang selalu memberikan semangat dan bantuan dalam menyelesaikan skripsi ini.

KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Segala puji bagi Allah SWT Tuhan Yang Maha Esa atas Rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan sanggup menyelesaikan skripsi dengan judul “Klasifikasi Penderita Diabetes Berbasis Data Rekam Medis Menggunakan *Support Vector Machine* (SVM)” sebagai salah satu syarat kelulusan guna mendapat gelar sarjana pada Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Skripsi ini tidak dapat terwujud tanpa adanya doa,restu, dan motivasi dari berbagai pihak serta niat dari penulis sendiri. Oleh karena itu, penulis ingin mengucapkan terima kasih sedalam-dalamnya kepada:

1. Prof. Dr. M. Zainudddin, MA., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang
2. Dr. Sri Hariani, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang
3. Dr. Fachrul Kurniawan M.MT.,IPM selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Irwan Budi Santoso, M.Kom selaku Dosen Pembimbing I yang senantiasa membimbing dan memberikan arahan bagi penulis untuk menyelesaikan skripsi ini.
5. Fatchurrochman,M.Kom selaku Dosen Pembimbing II yang juga telah membimbing penulis dalam menyelesaikan skripsi ini.
6. Dr. M. Amin Hariyadi, M.T sebagai Dosen Penguji I penulis dan A'la Syauqi, M.Kom selaku Dosen Penguji II penulis yang dengan sabar telah memberi arahan dan saran dalam proses penyelesaian skripsi ini.
7. Seluruh Dosen dan Jajaran Staf Program Studi Teknik Informatika, terima kasih atas segenap ilmu dan bimbingannya.

8. Sahabat penulis yang telah menemani penulis selama masa perkuliahan dan memberikan banyak motivasi serta bantuan dalam penyelesaian skripsi ini.
9. Seluruh keluarga besar Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang yang banyak memberikan pengalaman berharga.
10. Semua pihak yang telah membantu secara langsung atau tidak langsung dalam proses penyelesaian skripsi ini.
11. Diri penulis sendiri yang akhirnya mampu menyelesaikan karya tulis ini.

Penulis menyadari bahwa dalam penulisan skripsi ini masih jauh dari kata sempurna. Maka dari itu penulis membuka kesempatan sebesar-besarnya untuk tiap kritik dan saran yang membangun. Terlepas dari itu, penulis berharap semoga skripsi ini bisa memberikan manfaat kepada para pembaca.

Wassalamu'alaikum Wr. Wb.

DAFTAR ISI

HALAMAN JUDUL	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
البحث مستخلص	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah	5
1.3 Tujuan Penelitian	5
1.4 Batasan Masalah.....	5
1.5 Manfaat Penelitian	5
BAB II STUDI LITERATUR	7
2.1 Penyakit Diabetes.....	7
2.2 <i>Min Max Scaler</i>	7
2.3 <i>K-Fold Cross Validation</i>	8
2.4 <i>Support Vector Machine (SVM)</i>	9
2.5 Fungsi Kernel	12
2.6 Penelitian Terkait	12
BAB III METODE PENELITIAN	15
3.1 Pengumpulan Data	15
3.2 Tahapan Penelitian	16
3.3 Desain Sistem.....	19
3.3.1 <i>Split Data</i>	20
3.3.2 <i>Praproses</i>	21
3.3.3 <i>K-Fold Cross Validation</i>	24
3.3.4 <i>Implementasi SVM</i>	27
3.4 Skenario Pengujian.....	41
BAB IV UJI COBA DAN PEMBAHASAN	47
4.1 Langkah-Langkah Uji Coba.....	47
4.2 Hasil Training.....	48
4.3 Hasil Uji Coba.....	48
4.4 Implementasi Antarmuka	55
4.5 Pembahasan.....	57
BAB V KESIMPULAN DAN SARAN	61

5.1 Kesimpulan	61
5.2 Saran.....	62
DAFTAR PUSTAKA	

DAFTAR GAMBAR

Gambar 2. 1 Transformasi data ruang input ke ruang fitur	12
Gambar 3. 1 Tahapan Penelitian.....	16
Gambar 3. 2 Desain Sistem.....	20
Gambar 3. 3 Missing Value Pada Data	22
Gambar 3. 4 Representasi dari metode Cross Validation	25
Gambar 3. 5 Flowchart K-Fold Cross Validation	26
Gambar 3. 6 Proses Support Vector Machine	28
Gambar 3. 7 Flowchart Proses Training	29
Gambar 3. 8 Flowchart perhitungan matriks Hessian	31
Gambar 3. 9 Flowchart perhitungan nilai delta alfa.....	33
Gambar 3. 10 Flowchart perhitungan nilai alfa i	34
Gambar 3. 11 Flowchart Proses Testing	38
Gambar 3. 12 Confusion Matrix	45
Gambar 4. 1 Halaman Utama.....	55
Gambar 4. 2 Tampilan Klasifikasi Diabetes	56
Gambar 4. 3 Tampilan Klasifikasi Non-Diabetes	56
Gambar 4. 4 Grafik uji coba.....	58

DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait	14
Tabel 3. 1 Atribut dan outcome.....	16
Tabel 3. 2 Nilai fitur glucose sebelum normalisasi.....	23
Tabel 3. 3 Hasil normalisasi fitur glucose.....	23
Tabel 3. 4 Contoh Data Latih.....	30
Tabel 3. 5 Perhitungan Matriks Hessian	31
Tabel 3. 6 Perhitungan Nilai E_i pada Iterasi ke 1	32
Tabel 3. 7 Perhitungan nilai $I \delta\alpha_1$ pada Iterasi ke 1	34
Tabel 3. 8 Perhitungan nilai α_i pada iterasi ke 1	35
Tabel 3. 9 Perhitungan nilai E_i pada iterasi ke 2.....	35
Tabel 3. 10 Perhitungan nilai $\delta\alpha_i$ pada iterasi ke 2	36
Tabel 3. 11 Perhitungan nilai α_i pada iterasi ke 2	36
Tabel 3. 12 Data Uji	37
Tabel 3. 13 Nilai Matriks Hessian.....	39
Tabel 3. 14 Nilai X_+ dan X_-	39
Tabel 3. 15 Perhitungan Nilai $K(x_i, x_{test1})$ dan nilai $K(x_i, x_{test2})$	40
Tabel 3. 16 Hasil Klasifikasi.....	40
Tabel 4. 1 Tabel Hasil Training	48
Tabel 4. 2 Performa sistem menggunakan 8 fitur	49
Tabel 4. 3 Performa sistem menggunakan 7 fitur	50
Tabel 4. 4 Performa sistem menggunakan 6 fitur	50
Tabel 4. 5 Performa sistem menggunakan 5 fitur	51
Tabel 4. 6 Performa sistem menggunakan 4 fitur	52
Tabel 4. 7 Performa sistem menggunakan 3 fitur	52
Tabel 4. 8 Performa sistem menggunakan 2 fitur	53
Tabel 4. 9 Performa sistem menggunakan 1 fitur	54
Tabel 4. 10 Confusion Matrix Uji Coba 7 Fitur Kernel Linear	54
Tabel 4. 11 Performa Sistem (7 Fitur Kernel Linear)	55

ABSTRAK

Abimanyu, Arya. 2023. **Klasifikasi Penderita Penyakit Diabetes Berbasis Data Rekam Medis Menggunakan Support Vector Machine (SVM)**. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Irwan Budi Santoso (II) Fatchurrohman, M.Kom.

Kata kunci: *Support Vector Machine, Diabetes, Structural Risk Minimization*

Banyak penderita diabetes secara global yang terdiagnosa diabetes setelah mengalami komplikasi. Padahal, jika diagnosis dilakukan secara dini, penanganan diabetes dapat dilakukan lebih cepat dan dapat menghindari komplikasi penyakit lain yang berbahaya. Hal tersebut dapat didukung dengan penggunaan teknik *data mining* dalam menggali informasi berharga dari kumpulan data penyakit diabetes. *Support Vector Machine* (SVM) merupakan salah satu algoritma pada metode klasifikasi. Metode ini merupakan metode mesin pembelajaran dengan tujuan menemukan *hyperplane* (fungsi pemisah). Tujuan dari penelitian ini mengukur performa sistem klasifikasi penyakit diabetes menggunakan algoritma *Support Vector Machine*. Hasil dari penelitian ini percobaan yang memiliki hasil paling baik adalah uji coba menggunakan kernel *Linear 7* fitur. Dengan nilai akurasi sebesar 79.65%, presisi sebesar 66.26% serta *recall* dengan nilai 74.32%. Sehingga dapat disimpulkan bahwa dalam klasifikasi menggunakan metode *Support Vector Machine*, dengan input jumlah fitur yang berbeda maka hasil klasifikasi yang dihasilkan maka berbeda pula. Penggunaan jumlah fitur yang banyak bukan jaminan hasil klasifikasi akan baik juga, karna berpotensi terjadi *noise* pada proses klasifikasi. Namun dengan penggunaan jumlah fitur yang sedikit juga dapat menghilangkan fitur penting yang dapat membuat performa klasifikasi menjadi lebih baik.

ABSTRACT

Abimanyu, Arya. 2023. **The Classification of Diabetic Patients Based on the Medical Record Data Using Support Vector Machine (SVM)**. Undergraduate Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Dr. Irwan Budi Santoso (II) Fatchurrochman, M.Kom.

Keywords: Support Vector Machine, Diabetes, Structural Risk Minimization

Diabetes are globally diagnosed in several diabetic patients after they experience complications. In fact, if the diagnosis is made early, diabetes treatment can be carried out more quickly and complications of other dangerous diseases can be avoided. This can be supported by the use of data mining techniques in obtaining valuable information from diabetes data sets. Support Vector Machine (SVM) is one of the algorithms in the classification method. This method is a machine learning method aiming to find a hyperplane (separator function). This study aims to measure the performance of the diabetes classification system using the Support Vector Machine algorithm. The results of this study indicate that the experiment having the best results was the trial using the 7 features of Linear kernel. The accuracy value is 79.65%, the precision is 66.26% and the recall value is 74.32%. Thus, it can be concluded that within the classification using the Support Vector Machine method, with a different input of number of features, the classification results are different. In addition, the use of a large number of features cannot guarantee that the classification results will be good since noise has the potential to occur during the classification process. However, using a small number of features can also eliminate important features, which can make classification performance better.

Translator,	Date	the Director of Language Center,
Prima Purbasari, M.Hum NIDT 19861103201608012099	June 23, 2023	Prof. Dr. H. M. Abdul Hamid, MA NIP. 19732011998031007

البحث مستخلص

أبيمانيو، أريا. ٢٠٢٢. تصنيف مرضى السكري بناء على بيانات السجلات الطبية باستخدام آلة المتجه الداعمة (SVM). البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: د. إيروان بودي سانتوسو. المشرف الثاني: فتح الرحمن، الماجستير.

الكلمات الرئيسية: آلة المتجه الداعمة، مرض السكري، تقليل المخاطر الهيكلية.

يتم تشخيص العديد من مرضى السكري على مستوى العالم بمرض السكري بعد تعرضهم لمضاعفات. في الواقع، إذا تم التشخيص مبكرا، يمكن إجراء علاج مرض السكري بشكل أسرع ويمكن تجنب مضاعفات الأمراض الخطيرة الأخرى. يمكن دعم ذلك من خلال استخدام تقنيات استخراج البيانات في استخراج معلومات قيمة من مجموعات بيانات مرض السكري. آلة المتجه الداعمة (SVM) هي إحدى الخوارزميات في طريقة التصنيف. هذه الطريقة هي طريقة تعلم آلي بهدف العثور على المستوى الفائق (وظيفة الفاصل). الهدف من هذا البحث هو قياس أداء نظام تصنيف مرض السكري باستخدام خوارزمية آلة المتجه الداعمة. كانت نتيجة هذا البحث التي حققت أفضل النتائج هي تجربة باستخدام الجبر الخطي بسبع ميزات. مع قيمة الدقة ٧٩.٦٥% والضبط ٦٦.٢٦% والاستدعاء ٧٤.٣٢%. لذلك يمكن الاستنتاج منها أن في التصنيف باستخدام آلة المتجه الداعمة، مع إدخال عدد من الميزات المختلفة، تختلف نتائج التصنيف الناتجة. إن استخدام عدد كبير من الميزات لا يضمن أن نتائج التصنيف ستكون جيدة أيضا، لأن هناك إمكانية للوضاء في عملية التصنيف. ومع ذلك، فإن استخدام عدد صغير من الميزات يمكن أن يؤدي أيضا إلى التخلص من الميزات المهمة التي يمكن أن تجعل أداء التصنيف أفضل.

Penerjemah, M.Mubasysyir Munir, MA NIDT:19860513201802011215	Tanggal 23-6-2023	Validasi Kepala PPB, Prof. Dr. H. M. Abdul Hamid, MA NIP: 19730201 1998031007
---	-----------------------------	--

BAB I

PENDAHULUAN

1.1 Latar Belakang

Setiap tahun sejak 1991, *International Diabetes Federation* (IDF) serta organisasi Kesehatan Dunia (WHO) di tanggal 14 November ditetapkan sebagai Hari Diabetes. Pada tahun 2007, hari diabetes ditetapkan menjadi agenda *internasional* dalam agenda PBB. Hal ini merupakan sebuah indikator jika Diabetes perlu penanganan serius. Diabetes adalah suatu kondisi yang timbul karena gangguan metabolik yang disebabkan oleh kadar glukosa yang tinggi dalam darah yang berlangsung dengan jangka waktu yang lama. Kadar Glukosa dalam darah dapat bervariasi sepanjang hari, jika setelah makan kadar glukosa akan meningkat dan akan kembali normal setelah dua jam (Putra, et al., 2015). Diabetes dapat terjadi ketika insulin yang dihasilkan tubuh kurang untuk bisa mempertahankan kadar glukosa dalam darah agar normal atau dapat terjadi juga ketika respon yang diberikan sel tidak tepat terhadap insulin. Jika segera mendapat penanganan awal bahaya diabetes dapat memicu komplikasi.

Menurut data Infodatin Diabetes Melitus 2020 yang dirilis oleh Pusat Data dan Informasi KEMENKES Republik Indonesia, *International Diabetes Federation* (IDF) juga menyatakan sedikitnya ada 463 juta jiwa pada usia 20-79 tahun mengidap diabetes di tahun 2019. Jumlah tersebut dinilai akan terus meningkat hingga 578 juta jiwa di 2030 dan 700 juta

jiwa pada tahun 2045 (Pangribowo, 2020). Indonesia termasuk kedalam sepuluh besar negara yang warganya terjangkit diabetes dengan jumlah 10,7 juta jiwa yang menempatkan Indonesia pada urutan ke-7. Prediksi menyatakan bahwa jumlah tersebut diperkirakan akan terus meningkat, mencapai 13,7 juta orang di 2030, dan 16,6 juta orang pada tahun 2045. (Anon., 2019)

Banyak individu yang menderita diabet baru mendapatkan diagnosis setelah terjadi komplikasi. Namun, hal tersebut dapat dihindari melalui diagnosis yang lebih awal, sehingga pertolongan medis diabet dapat dilaksanakan secara cepat dan bisa menghindari risiko komplikasi penyakit lain. Penggunaan teknik data mining dapat mendukung upaya ini dengan mendapat informasi penting dari data penyakit diabet (Isbandiyo, 2016). Salah satu Teknik *data mining* dalam adalah metode pembelajaran berbimbing, yang juga dikenal sebagai *supervised learning* (Anon., n.d.). Salah satu jenis metode tersebut ialah klasifikasi. Klasifikasi dapat dimanfaatkan untuk pengambilan keputusan berdasarkan pola baru yang ditemukan dari data sebelumnya menggunakan algoritma perhitungan. (Indrayanti, et al., 2017).

Beberapa algoritma dapat digunakan untuk melakukan tugas klasifikasi (Han & Kamber, 2006). Klasifikasi bisa dilakukan dengan salah satu algoritma yaitu *Support Vector Machine* (SVM). SVM merupakan *algorithm* perhitungan yang baik di dalam mendapatkan hasil klasifikasi dengan tingkat akurasi tinggi (Navin & Singh, 2015). Beberapa penelitian yang menerapkan *Support Vector Machine* sudah banyak diteliti seperti yang dilakukan oleh Muis dan Affandes (2015) yang melakukan klasifikasi *tweet* dengan menerapkan *support vector*

machine dan menghasilkan akurasi tertinggi hingga 97,54%. Dalam penelitian lain yang dilakukan oleh Kaaffah dkk (2020) yang menerapkan *support vector machine* dalam klasifikasi ukuran baju menghasilkan akurasi sebesar 92%, presisi 93.8% dan *recall* senilai 88.5%.

SVM bekerja dengan asas *Structural Risk Minimization* (SRM), metode ini merupakan metode mesin pembelajaran dengan tujuan menemukan *hyperlane* (fungsi pemisah). *Hyperlane* yaitu suatu garis yang memisahkan data antar kelas/kategori (Irmanda & Astriratma, 2020) . SVM pada dasarnya didesain untuk klasifikasi 2-kelas (Kaaffah, et al., 2020). SVM memiliki poin *plus* diantaranya ialah ketika menentukan jarak menggunakan *support vector* sehingga proses komputasi menjadi cepat (Cortes & Vapnik, 1995). SVM berada dalam satu kelas dengan ANN dalam hal fungsi dan kondisi permasalahan yang bisa diselesaikan.

Klasifikasi diabetes sudah banyak dilakukan dengan berbagai metode klasifikasi, seperti yang dilakukan oleh Hairani dkk (2020) yang menggunakan metode *Naïve Bayes*, *SVM* dan *C4.5* yang menghasilkan kesimpulan bahwa metode klasifikasi *SVM* menghasilkan akurasi dan sensitivitas sebesar 82% dan 77%. Lalu pada penelitian yang dilakukan oleh Purbolaksono dkk (2021) yang membandingkan *Support Vector Machine* dan *Modified Balanced Random Forest* menghasilkan akurasi dari *SVM* mencapai 91.48%. Selain itu Apriyani dan Kurniati (2020) yang melakukan penelitian dengan membandingkan metode *Support Vector Machine* dan *Naïve Bayes* untuk klasifikasi penyakit diabetes menghasilkan kesimpulan *SVM* memiliki akurasi lebih tinggi dengan nilai 96.2704% dibanding *Naïve Bayes* dengan tingkat akurasi 92.0746%. Dari latar

belakang di tersebut, penelitian ini akan mengimplementasikan algoritma *Support Vector Machine* (SVM). Objek riset yang hendak digunakan dalam riset merupakan berasal dari *National Institute of Diabetes and Digestive and Kidney Diseases* yang bisa diakses di *UCI Machine Learning Repository : Pima Indians Diabetes Database*. *Dataset* ini tersusun dari informasi 768 pasien perempuan dengan 8 fitur diagnosa kedokteran yang berbeda serta 1 fitur label.

Penelitian ini diharapkan bisa menjadi pengingat bagi penulis dan kepada masyarakat luas untuk senantiasa menjaga kesehatan dan menjauhi perbuatan-perbuatan yang dapat memberikan efek bagi kesehatan. Seperti dalam salah satu hadits Nabi Muhammad shallallahu ‘alaihi wa sallam yang melarang umatnya untuk makan secara berlebihan dan terburu-buru. Hadits dari Ibnu Ka’ab bin Malik, dari ayahnya ia berkata:

يَمْسَحُهَا أَنْ قَبْلَ يَدِهِ وَيَلْعَقُ أَصَابِعَ بَثَلَاثَ يَأْكُلُ وَسَلَّمَ عَلَيْهِ اللَّهُ صَلَّى اللَّهُ رَسُولَ كَانَ

“*Rasulullah shallallahu ‘alaihi wa sallam makan dengan tiga jari, dan beliau menjilatinya sebelum mencuci tangannya.*” (HR. Muslim: 2032)

Di riwayat tersebut Rasulullah shallallahu ‘alaihi wa sallam memberi tuntunan kepada pengikutnya untuk mempergunakan tangan saat makan. Menurut Situs healthierwayoflife.com pada hari Selasa (13/10/2015) salah satu makan menggunakan tangan dibanding alat bantu seperti sendok dan sebagainya saat makan dari kaca mata medis ialah dapat mengurangi resiko terjangkit diabetes tipe 2 karena orang-orang yang makan dengan cepat beresiko lebih tinggi terkena diabet tipe 2. Dasarnya ketika seseorang dapat makan lebih cepat ketika menggunakan alat bantu seperti sendok, garpu dan sebagainya. Karena itu

dianjurkan bagi kita untuk makan menggunakan tangan langsung dibanding menggunakan alat bantu.

1.2 Pernyataan Masalah

Berapa nilai performa sistem dalam mengklasifikasikan penyakit diabetes menggunakan metode *Support Vector Machine*?

1.3 Tujuan Penelitian

Menghitung nilai performa meliputi akurasi, presisi, dan *recall* dari metode *Support Vector Machine* untuk mengklasifikasikan penderita diabetes dengan data rekam medis.

1.4 Batasan Masalah

1. Data yang digunakan adalah data sekunder yang dapat diakses oleh public melalui *UCI Machine Learning Repository : Pima Indians Diabetes Database*
2. Performa yang menjadi variabel pengukuran adalah presisi, akurasi, dan *recall*

1.5 Manfaat Penelitian

Hasil dari riset ini adalah *output* performa dari metode *Support Vector Machine* untuk mengklasifikasi diabetes. Kelompok-kelompok yang bisa mendapat manfaat dari hasil penelitian ini antara lain:

- a. Peneliti *Data Mining* untuk tujuan rujukan penelitian;

- b. Komunitas *Data Science* dengan tujuan musyawarah; dan pihak-pihak terkait.

BAB II

STUDI LITERATUR

2.1 Penyakit Diabetes

Diabetes merupakan penyakit yang diderita dalam waktu lama yang diakibatkan insulin yang dihasilkan pancreas tidak memenuhi kebutuhan atau ketika insulin yang dihasilkan tidak dapat digunakan secara efisien oleh badan. Insulin sendiri mempunyai guna selaku pengubah glukosa menjadi glikogen serta trigliserida dalam tubuh. Diabet yang tidak terkontrol dapat mengakibatkan kerusakan sungguh-sungguh pada banyak organ dalam dan saraf. Diabet pula merupakan penyebab serangan jantung, kebutaan dan gagal ginjal sampai kematian.

Pada tahun 2019 menurut (IDF) Organisasi International Diabetes Federation di seluruh dunia penderita diabet terdapat 463 juta, jumlah tersebut diperkirakan dapat meningkat menjadi 578 juta jiwa di tahun 2030. Penentuan jenis penyakit diabetes membutuhkan kemampuan ahli atau pakar (Riadi, 2017).

2.2 *Min Max Scaler*

Normalisasi merupakan proses penskalaan nilai atribut dari informasi sehingga dapat terletak pada rentang tertentu. *Min-max Scaler* ialah salah satu tata cara *normalization* dengan cara linier yang mentransformasikan data dari satu *space* nilai ke *space* nilai yang baru, sehingga nilai perbandingan antar data menjadi seimbang. Metode *Min-max Scaler* merupakan salah satu metode

yang mengubah data yang kompleks dengan tidak menghilangkan isi, sehingga lebih mudah diolah (Wimmer, 2018) . Penelitian oleh (Mucholladin et al., 2021) yang menggunakan *Pima Indians Dataset* sebagai objek penelitian menerapkan *Min-Max Scaller* karena distribusi yang ada sudah mendekati distribusi normal sehingga hanya perlu penskalaan ulang menggunakan *Min-Max Scaller*. Cara kerja dari *Min max scaler* ,setiap *value* pada fitur tersebut dikurangi dengan *value* minimum yang ada pada fitur tersebut, kemudian dibagi dengan rentang nilai dari fitur tersebut. Persamaan 2.3 merupakan rumus *Min-max scaler* :

$$z = \frac{X - \min(x)}{[\max(x) - \min(x)]} \quad (2.3)$$

Keterangan :

Z = Hasil normalisasi

X = nilai x (asli)

Min(x) = nilai minimal ntuk variabel x

Max(x) = nilai maksimal untuk variabel x.

2.3 *K-Fold Cross Validation*

Validasi dapat diartikan sebagai sebuah proses untuk menguji performa dari sebuah *algorithm*. Pada umumnya *validation* dilaksanakan dengan metode mengulangi proses perhitungan sampai dengan beberapa kali. Dalam riset ini cara yang digunakan untuk validasi adalah *cross validation*. *Cross validation* merupakan metode berfungsi guna mendapatkan hasil *accuracy* yang maskimal. *K-Fold Cross Validation* merupakan jenis percobaan *cross validation* yang bermanfaat guna mengukur performa sebuah algoritma melalui cara memisah

sample data menjadi *test data* dan *validation data* secara acak dan mengelompokan data tersebut sebanyak nilai K . Berdasarkan Marcot dan Hanea (2021) beberapa penelitian yang mengimplementasikan $k = 10$ dalam *k-fold cross validation*, namun dengan menggunakan $k = 5$ akan lebih menghemat waktu dan komputasi. Alur dari *k-fold validation* adalah sebagai berikut:

1. Keseluruhan data dibagi menjadi k bagian
2. *Fold* pertama merupakan ketika bagian pertama menjadi *validation data* dan sisanya menjadi *testing data*. Kemudian menghitung nilai akurasi atau seberapa dekat hasil pengukuran dengan data sebenarnya. Cara kalkulasi akurasi menggunakan persamaan berikut.

$$akurasi = \frac{\sum \text{data uji benar klasifikasi}}{\sum \text{total data uji}} \times 100 \quad (2.4)$$

3. *Fold* kedua ialah ketika data kedua jadi *test data* kemudian sisanya *validation data*. Kemudian kalkulasi akurasi berdasarkan porsi data tersebut.
4. Demikian seterusnya sampai *fold* ke- k . Rata-rata akurasi ddari k dihitung. Rata-rata akurasi ini menjadi akurasi final.

2.4 Support Vector Machine (SVM)

Awalnya *Support Vector Machine* (SVM) dipublikasikan pada 1992 Vapnik bersama Bernhard Boser, Isabelle Guyon. *Support vector machine* ialah *algorithm* yang berjalan berdasarkan pemetaan nonlinier untuk mengalihkan *testing data* asli ke ukuran yang lebih luas. SVM adalah salah satu *algorithm machine learning* yang berkerja atas prinsip *structural risk minimazation* yang

bertujuan menciptakan *hyperlane* yang membagi 2 buah kelas pada *input space* (Darsyah, 2014). Margin adalah ruang maksimal yang dapat digunakan untuk membedakan dua kelas. Hal yang paling penting dalam menemukan *support vector* adalah *dot product* yang memiliki jarak terdekat dengan margin. SVM juga memiliki tujuan guna meminimalkan batas tertinggi dari *general error*. Kelebihan lainnya dari *algorithm* ini yaitu penggunaan metode SVM dapat dikaji secara teoritis dengan menerapkan konsep teori pembelajaran komputasi.

Konsep *Support Vector Machine* (SVM) yaitu mencari *hyperplane* yang memilah himpunan data ke dalam dua kelas secara *linier*. Dalam perihal ini ukuran baru, akan mendapatkan *hyperplane* optimal guna mengklasifikasikan secara *linier* dan dengan pemetaan *nonlinier* yang tepat ke dimensi lebih tinggi, data dari 2 kelas senantiasa sanggup dipisah menggunakan *hyperplane* tersebut. Konsep bawah SVM sesungguhnya merupakan campuran yang sesuai dari teori komputasi yang telah ada selama sebagian dekade sebelumnya seperti margin *hyperlane*.(Duda & Hart 1973). SVM berusaha menemukan *hyperlane* yang paling optimal, *hyperplane* yang paling optimal merupakan *hyperplane* yang ada di tengah antara dua set objek dari dua kelas (Ritonga & Purwaningsih, 2018). Ada dua langkah yang dilakukan SVM untuk mendapatkan *hyperlane*, yaitu menemukan data – data terluar pada kedua kelas yang berada di perbatasan lalu dengan memperhitungkan data-data terluar tersebut dan tanpa memperhitungkan data lainnya ditentukanlah *hyperlane* paling optimum (Suyanto, 2018).

Support Vector Machine awalnya hanya bisa diaplikasikan untuk klasifikasi informasi yang hanya memiliki 2 kelas. Klasifikasi ke dalam 2 kelas biasa disebut

binary classifier. Tetapi bersama dengan perkembangannya, SVM dapat mengklasifikasikan data yang memiliki jumlah kelas lebih dari dua atau biasa disebut *multiclass*. Prosedur klasifikasi menggunakan SVM dapat dirumuskan sebagai berikut :

$$f(x) = \sum_{i=1}^m \alpha_i y_i K(x_l, x) + b \quad (2.5)$$

Keterangan :

α_i . = alfa nilai ke-i.

$K.(x_l, x_j)$ = kernel nilai yang dipakai.

y_i . = data nilai dari kelas ke-i.

b . = nilai *bias*

m . = data jumlah dari SV.

Dibawah ini merupakan rumus untuk mencari nilai *bias*.

Persamaan untuk mencari nilai bias adalah sebagai berikut.

$$b = \frac{1}{2} \left[\sum_{i=1}^m \alpha_i y_i K(x_l, x_+) + \sum_{i=1}^m \alpha_i y_i K(x_l, x_-) \right] \quad (2.6)$$

Keterangan :

α_i . = alfa nilai ke-i.

y_i . = data nilai dari kelas ke-i.

m . = data jumlah dari SV.

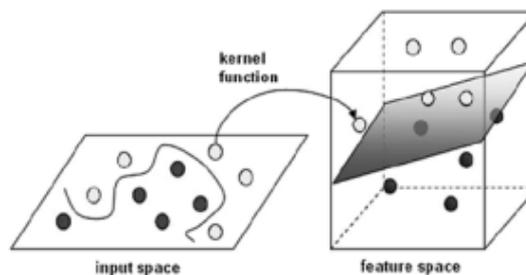
$K.(x_l, x_-), K.(x_l, x_+) =$ kernel nilai yang dipakai.

2.5 Fungsi Kernel

Dalam klasifikasi yang menggunakan banyak sampel data, terdapat data yang tidak bisa terbagi secara linier oleh karena itu jika menggunakan SVM model *linear output* yang dihasilkan kurang optimal dan menghasilkan hasil klasifikasi yang tidak baik. Karena itu *algorithm Support Vector Machine* (SVM) dimutasi memasukan *kernel function* kedalam *non linear Support Vector Machine*. *Kerenel function* merupakan *function k* dimana semua vector masukan x serta z akan melengkapi kondisi :

$$k(x, z) = \varphi(x)^t \varphi(z) \quad (2.7)$$

Gambar 2.1 menunjukkan ilustrasi transformasi data ruang *input* ke ruang fitur. Di algoritma ini, *kernel function trick* saat model pembelajaran sangat berguna untuk permasalahan *feature space*.



Gambar 2. 1 Transformasi data ruang input ke ruang fitur\

2.6 Penelitian Terkait

Riset oleh (Chang, et al., 2022) *dataset* PID dipilih untuk diklasifikasikan menggunakan metode *Naïve Bayes Classifier*, *Random Forest Classifier*, dan *J48 decision tree*. Penelitian ini menerapkan sebuah sistem e-diagnosis yang diimplementasikan pada *Internet of Medical Things* (IoMT). Dalam penelitian

yang dilakukan sistem yang dibangun menggunakan bahasa pemrograman R. Kesimpulan dari penelitian ini algoritma *Naïve Bayes* bekerja lebih baik dengan pilihan fitur yang lebih halus untuk klasifikasi biner, sementara algoritma *Random forest* menghasilkan *output* lebih baik dengan lebih banyak fitur

Dataset PID digunakan pada riset oleh (Abedini, et al., 2020) pada riset tersebut dilakukan pengklasifikasian menggunakan dataset PID sebagai objek penelitian. Langkah pertama yang dilakukan dalam penelitian tersebut adalah melatih *Decision Tree* dan *Logistic Regression*, kemudian selanjutnya langkah kedua setelah mendapat *output* dari dua algoritma tersebut *output* dijadikan *input* untuk *Neural Network*. Sebelumnya *Neural Network* juga dilatih untuk menggabungkan dua *output* dari pengklasifikasian sebelumnya untuk mendapat akurasi keseluruhan yang baik. Hasil dari penelitian tersebut, model yang dikembangkan mendapat hasil akurasi diatas 83%.

Penelitian berjudul “*Classification of Diabetes using Deep Learning*” dilakukan oleh (Kumar, et al., 2020). Pada riset ini PID digunakan sebagai objek penelitian untuk klasifikasi diabetes menggunakan metode *Multi-Layer Feed Forward Neural Network* (MLFNN). Pada penelitian tersebut juga menerapkan *Min-Max Scaller* untuk menskala ulang data kedalam *range* tertentu untuk mengurangi bias dalam *Neural Network*. Terakhir dari penelitian ini dibandingkan dengan *algorithm Naïve Bayes* dan *Random Forest*. Hasilnya MLFNN mendapat akurasi sebesar 84,17%.

Kemudian penelitian yang dilakukan oleh (Sisodia & Sisodia, 2018) juga menggunakan PID sebagai objek penelitian dalam penelitian berjudul “*Prediction*

of Diabetes using Classification Algorithms”. Dalam penelitian yang dilakukan mengimplementasikan 3 algoritma untuk mengklasifikasikan diabetes yaitu, *Decision Tree*, *Support Vector Machine*, dan *Naïve Bayes*. Kinerja ketiga algoritma diulas dengan berbagai faktor *Precision*, *Accuracy*, *F-Measure*, dan *Recall*. Penelitian ini menyimpulkan *Naïve Bayes* mendapat hasil akurasi paling tinggi sebesar 76,30%. Hasil ini divalidasi dengan kurva *Receiver Operating Characteristic* (ROC) secara tepat dan sistematis.

Tabel 2. 1 Penelitian Terkait

No	Peneliti (Tahun)	Metode dan Studi Kasus
1	Chang dkk (2022)	-Metode yang digunakan <i>Naïve Bayes Classifier</i> , <i>Random Forest Classifier</i> , dan <i>J48 decision tree</i> -Penerapan sistem kedalam Internet of Medical Things (IoMT) -Sistem yang dibangun menggunakan bahasa R
2	Abedini dkk (2020)	-Penelitian menggunakan <i>Decision Tree</i> dan <i>Logistic Regression</i> - Output dari 2 metode diproses menggunakan <i>Neural Network</i> - Belum menggunakan <i>K-Fold Cross Validation</i> untuk split data
3	Kumar dkk (2020)	-Menggunakan metode <i>Multi-Layer Feed Forward Neural Network (MLFNN)</i> -Menerapkan <i>Min Max Scaller</i> untuk mengurani nilai bias - Hasil akhir penelitian dibandingkan dengan hasil dari klasifikasi <i>Naïve Bayes</i> dan <i>Random Forest</i> -Tidak menggunakan <i>K-Fold Cross Validation</i>
4	Sisodia dan Sisodia 2018	-Membandingkan 3 metode yaitu, <i>Decision Tree</i> , <i>Support Vector Machine</i> , dan <i>Naïve Bayes</i> -Sistem yang dibangun diterapkan menggunakan WEKA tool -Menggunakan <i>10-fold cross validation</i>
5	Penelitian ini (2022)	-Klasifikasi menggunakan <i>Support Vector Machine</i> -Normalisasi menggunakan <i>Min-Max Scaller</i> -Pembagian data menggunakan <i>K-fold cross validation</i>

BAB III

METODE PENELITIAN

3.1 Pengumpulan Data

Penelitian ini memanfaatkan datasekunder, data sekunder yang dipakai di penelitian diambil dari web *Kaggle*. Data sekunder sendiri ialah data yang dihimpun buakn oleh peneneliti langsung melainkan pihak lain dan peneliti gunakan sebagai data penelitian dan dimanfaatkan untuk *testing data* dalam melakukan tahapan klasifikasi sehingga mendapat *output* prediksi penelitian. Dalam penelitian ini *dataset* yang dimanfaatkan sebagai *training data* dan *test data* berasal dari web *Kaggle*.

Dikarenakan riset ini memakai data sekunder yang notabene data bukan bersumber dari peneliti langsung diperlukan pengecekan untuk mengetahui ada atau tidaknya *missing value* dari dataset *Pima Indians Diabetes Database*. *Missing value* adalah ketika tidak ada nilai yang disimpan untuk variabel tertentu, *missing value* yang tidak dibenahi dapat berpengaruh terhadap hasil analisis sehingga menimbulkan hasil analisis yang tidak sesuai dengan harapan. Untuk melakukan pengecekan terhadap *missing value* digunakan *package Panda*

Data yang dipakai dalam penelitian merupakan *public dataset* yang bersumber dari *Kaggle* yaitu dataset *Pima Indians Diabetes Database*. Dataset ini berasal dari hasil diagnosa diabet pasien perempuan yang memiliki usia paling muda 21 tahun. Datase *Pima Indians Diabetes Database* terdiri

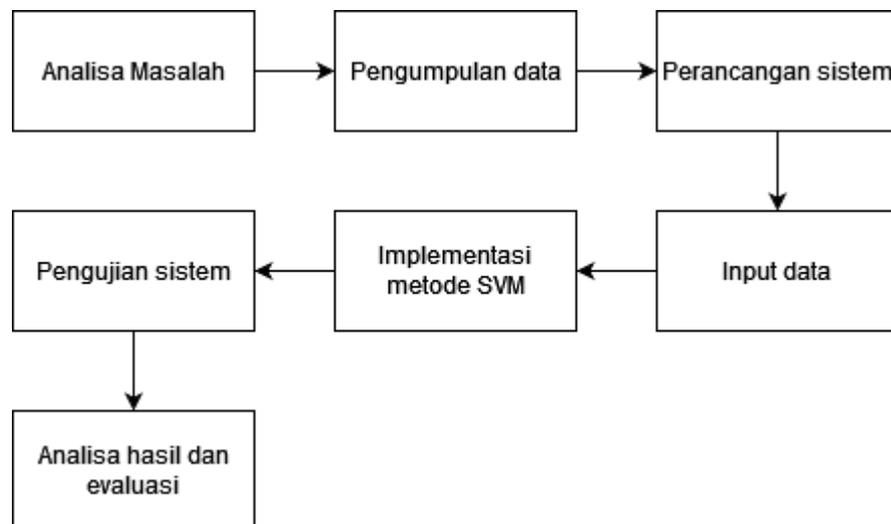
berdasarkan 768 informasi yang terbagi menjadi 8 atribut seperti pada Tabel 3.1 dan dua kelas dengan masing-masing kelas 1 (268) dan kelas 0 (500).

Tabel 3. 1 Atribut dan outcome

No	Attribute Name	Description
1	Pregnancies	Jumlah Kehamilan
2	Glucose	Kadar glukosa 2 jam setelah konsumsi larutan glukosa
3	BloodPressure	Tekanan dara <i>diastolic</i> (mmHg)
4	SkinThickness	Ketebalan lipatan kulit <i>triceps</i> (mm)
5	Insulin	Kadar insulin dalam darah, 2 jam setelah makan (muU/ml)
6	BMI	Berat masa tubuh ($\text{weight in kg}/(\text{height in m})^2$)
7	DiabetesPedifreeFunction	Riwayat penyakit keluarga
8	Age	Umur
9	Outcome	Variable (0 or 1)

3.2 Tahapan Penelitian

Dalam sub-bab ini akan dipaparkan mengenai alur penelitian yang akan dilaksanakan. Sebuah penelitian memerlukan prosedur penelitian yang jelas agar dapat berjalan sesuai dengan rencana. Prosedur penelitian yang akan dilakukan ditunjukkan pada Gambar 3.2.



Gambar 3. 1 Tahapan Penelitian

1. Analisa Masalah

Tahapan ini dilakukan analisa masalah yang ada dan dipertimbangkan untuk dijadikan judul penelitian serta mencari metode yang akan diterapkan untuk menyelesaikan permasalahan yang ditemukan. Pada tahap ini dihasilkan bahwa peneliti akan mengangkat judul penelitian mengenai klasifikasi penderita diabetes berdasarkan data rekam medis pasien. Metode yang diterapkan dan menyelesaikan permasalahan tersebut adalah *Support Vector Machine (SVM)*. Pengambilan keputusan tersebut didasarkan pada penjabaran yang ada pada bagian Latar Belakang di Bab 1.

2. Pengumpulan Data

Kemudian setelah mendapatkan masalah yang akan dijadikan judul penelitian tahap selanjutnya ialah mencari data yang diperlukan guna menjadi *dataset* yang dipakai dalam riset ini. Dalam riset ini peneliti menggunakan *dataset public* yang berasal dari web Kaggle. *Dataset* yang dipakai dalam riset ini ialah *Pima Indians Diabetes Database*.

3. Perancangan Sistem

Tahap selanjutnya setelah mendapatkan data yang akan dipakai dalam riset ini didapat ialah merancang sistem yang akan dibangun guna menyelesaikan permasalahan yang telah dianalisa sebelumnya. Tahapan perancangan sistem meliputi membuat alur dari sistem yang akan dibangun mulai *input data* sampai menghasilkan *output* sesuai dengan diinginkan serta mencari penelitian serupa untuk menjadi *benchmarking* dalam membangun sistem dalam penelitian ini

4. Input Data

Setelah rancangan sistem sudah selesai selanjutnya ialah memasukan data kedalam sistem yang sudah dirancang sebelumnya. Proses input data ini juga termasuk melakukan praproses pada data yang digunakan dalam penelitian guna memudahkan pengolahan data oleh sistem serta mengoptimalkan kinerja sistem yang nantinya akan memberi efek pada hasil yang dihasilkan oleh sistem juga akan lebih optimal dibanding data mentah yang langsung diolah oleh sistem. Praproses data yang dilakukan pada penelitian ini meliputi mencari *missing value* pada data dan menormalisasi data menggunakan *min-max scaler*.

5. Implementasi Metode SVM

Selanjutnya setelah *dataset* yang akan dipakai di riset ini sudah di *input* dan melalui tahapan praproses maka tahap berikutnya ialah mengimplementasikan *algorithm* yang akan dipakai dalam riset ini. Dalam riset ini *algorithm* yang diimplementasikan untuk memproses data agar menghasilkan *output* yang diinginkan adalah *Support Vector Machine*. Pemilihan penggunaan metode ini dalam penelitian ini dijelaskan pada Latar belakang penelitian yang ada di Bab 1. SVM yang diimplementasikan dalam riset ilmiah ini meliputi *Support Vector Machine* dengan beberapa kernel yaitu, Linear, RBF serta Polynomial.

6. Pengujian Sistem

Tahap selanjutnya setelah metode yang digunakan sudah diimplementasikan kedalam sistem adalah melakukan pengujian sistem yang telah dibangun. Pengujian dalam riset ini menerapkan metode *K-fold cross validation* dengan *fold=5*. Uji coba sistem meliputi pemisahan *dataset* menjadi *training data* serta

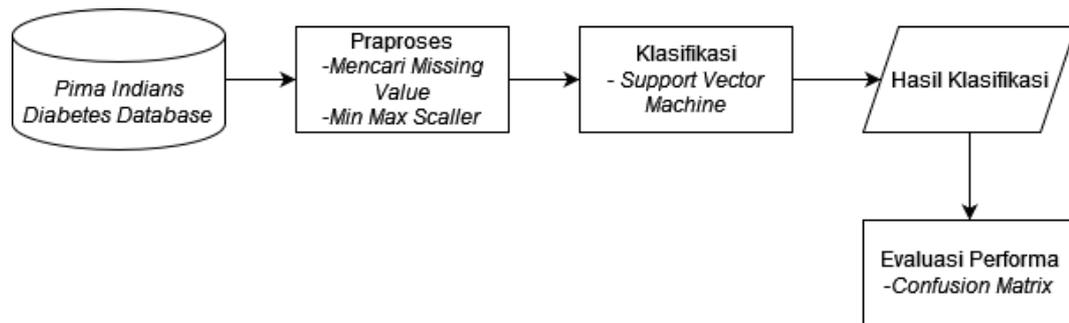
testing data, pengolahan dan pengklasifikasian data yang menggunakan metode (SVM) dengan kernel Linear, RBF, dan Polynomial. Pengujian dilakukan hingga sistem menghasilkan *output* berupa hasil akurasi, presisi, dan *recall*.

7. Analisa Hasil dan Evaluasi

Setelah sistem menghasilkan *output* berupa hasil akurasi, presisi, dan *recall* dari tiap pengujian yang dilakukan maka langkah terakhir adalah analisa dan *evaluation* dari *output* dihasilkan sistem. Analisa dilakukan dengan mencari nilai *accuracy*, presisi, dan *recall* yang optimal dari tiap pengujian menggunakan kernel Linear, RBF, dan Polynomial dimana tiap kernel diuji menggunakan 8 fitur sampai 1 fitur. Hal tersebut dilakukan dengan tujuan mencari penggunaan fitur dengan nilai paling optimal untuk digunakan dalam mengklasifikasikan data yang digunakan menggunakan *Support Vector Machine*. Kemudian disimpulkan untuk mendapatkan performa yang optimal dalam penelitian ini menggunakan berapa fitur serta kernel jenis apa yang cocok untuk mengklasifikasikan data yang digunakan.

3.3 Desain Sistem

Dalam riset ini akan diwujudkan sebuah *system* yang mampu mengklasifikasikan penderita diabetes berdasarkan gaya hidup. Untuk membuat sistem tersebut maka diperlukan adanya desain sistem. Tahap desain sistem ditunjukkan pada Gambar 3.2.



Gambar 3. 2 Desain Sistem

Algorithm yang diimplementasikan dalam penelitian ini adalah *Support Vector Machine* (SVM) oleh sebab itu desain sistem dalam riset ini akan meliputi prosedur latih dan prosedur uji. Prosedur latih diperlukan untuk melatih algoritma supaya bisa memahami pola informasi yang dijadikan sebagai input. Proses testing adalah proses untuk mengetahui performa dari metode SVM yang sudah dilatih sebelumnya. Dalam Gambar 3.2 merupakan desain *system* dari riset ini.

3.3.1 Split Data

Langkah pertama dalam penelitian ini ialah memilah data penelitian menjadi *training data* dan *testing data*. Pemisahan *training data* serta *data testing data* menggunakan perbandingan 70:30. Dari total 768 data yang diplot menjadi *training data* sejumlah 537 data lalu data yang digunakan sebagai *testing data* berjumlah 231 data. Proses pembagian data menggunakan *library python sklearn*. Data yang sudah dipisah menjadi data uji akan disimpan untuk nantinya digunakan dalam proses pengujian, sedangkan data yang menjadi data latih akan masuk proses *K-fold cross validation* dan proses pelatihan. Berikut adalah implementasi pembagian *dataset* menjadi data uji dan data latih.

```

Features=['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickn
ness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']
X = df.loc[:, features]
y = df.loc[:, ['Outcome']]
X_train, X_test, y_train, y_test = train_test_split(X, y,
train size=0.7, test size=0.3, random state=0)

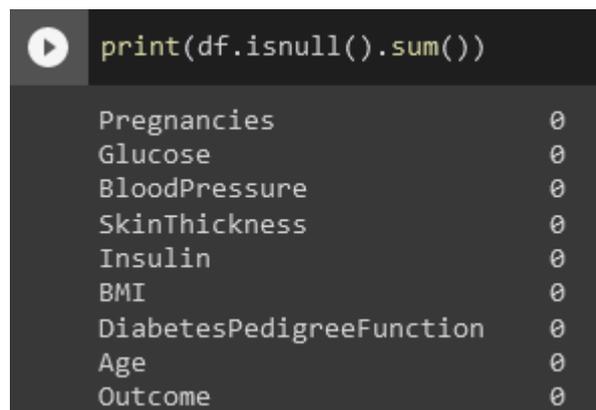
```

3.3.2 Praproses

Tahapan praproses dilakukan guna mengolah data agar siap dipakai untuk penelitian ini. Praproses dalam riset ini meliputi beberapa tahapan yaitu pencarian missing value pada *dataset*, kemudian tahapan praproses selanjutnya ialah melakukan normalisasi dengan tujuan untuk menskala ulang data menggunakan skala umum tanpa mengubah *value* dari data tersebut.

1. Mencari *missing value*

Pencarian *missing value* pada data bertujuan untuk mendeteksi ada tidaknya data yang tidak memiliki *value* pada fitur tertentu. Karena dalam penelitian menggunakan data sekunder yang notabene *dataset public* dimana data dikumpulkan oleh pihak lain dan tersedia secara *public* yang bisa digunakan untuk menguji suatu metode penelitian maka pencarian *Missing value* diperlukan untuk menghindari adanya data ber-*value* kosong dan memastikan data yang digunakan dalam penelitian mempunyai struktur yang baik. *Missing value* memiliki beberapa tipe diantaranya hilang keseluruhan secara acak, hilang secara *random* dan hilang tidak secara *random*. Sedangkan cara untuk menangani *missing value* yang biasa digunakan adalah menghapus secara baris yang terdapat *missing value* dan menghitung nilai pengganti untuk *missing value* tersebut. Hasil dari pencarian *missing value* pada *dataset* yang akan digunakan disajikan pada Gambar 3.3.



```

print(df.isnull().sum())
Pregnancies      0
Glucose           0
BloodPressure    0
SkinThickness    0
Insulin          0
BMI              0
DiabetesPedigreeFunction  0
Age              0
Outcome          0

```

Gambar 3. 3 *Missing Value* Pada Data

Dari hasil pencarian *missing value* dalam penelitian ini terhadap data yang digunakan hasilnya tidak ditemukan *missing value* pada tiap fitur. Karena tidak adanya *missing value* pada data maka tidak diperlukan penanganan untuk menangani *missing value*.

2. *Min-Max Scaller*

Pada tahap praproses ini data dinormalisasikan menggunakan *Min-max scaler*. *Dataset* yang dimanfaatkan dalam riset ini mempunyai *value* yang timpang antara satu fitur dengan fitur lainnya, maupun di dalam satu fitur terdapat nilai yang timpang antar data. Karena itu diperlukan normalisasi data terlebih dahulu sebelum data diolah di dalam sistem. Fungsi penggunaan metode normalisasi dalam penelitian ini ialah untuk mengubah kolom numerik menjadi skala umum tanpa mengubah *value* dari data itu sendiri. Proses normalisasi dapat dikalkulasi menggunakan persamaan *Min-max scaler*. Persamaan *Min-Max scaller* sesuai dengan Persamaan 2.3. Sebagai gambaran berikut pada Tabel 3.2 merupakan contoh *value* yang akan dinormalisasikan menggunakan *Min-max Scaller* :

Tabel 3. 2 Nilai fitur *glucose* sebelum normalisasi

NO	Glucose
1	148
2	85
3	183
4	89
5	137
6	116
7	78
8	115
9	197
10	125

Hitung menggunakan Persamaan 2.3, Maka didapatkan nilai baru dari fitur *glucose* yang sudah dinormalisasikan menggunakan *Min-max scaller* dapat dilihat pada Tabel 3.3. Implementasi dari *Min-Max Scaller* ditampilkan berikut.

Tabel 3. 3 Hasil normalisasi fitur *glucose*

NO	Glucose
1	0,3529411765
2	0,0588235294
3	0,9195979899
4	0,4472361809
5	0,6884422111
6	0,5829145729
7	0,391959799
8	0,5778894472
9	0,9899497487
10	0,6281407035

```

trans_test=test_data[['Pregnancies','Glucose','BloodPressure','SkinThi
ckness','Insulin','BMI','DiabetesPedigreeFunction','Age']]
label = test_data['Outcome']
scaler = MinMaxScaler()
abis_scaler = scaler.fit_transform(trans_test)
AbisTransform = pd.DataFrame(abis_scaler, columns
=['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','B
MI','DiabetesPedigreeFunction','Age'])
result_scaler = label.to_frame().reset_index(drop=True)
test_transform = pd.concat([AbisTransform,result_scaler], axis=1)
test_transform

```

3.3.3 K-Fold Cross Validation

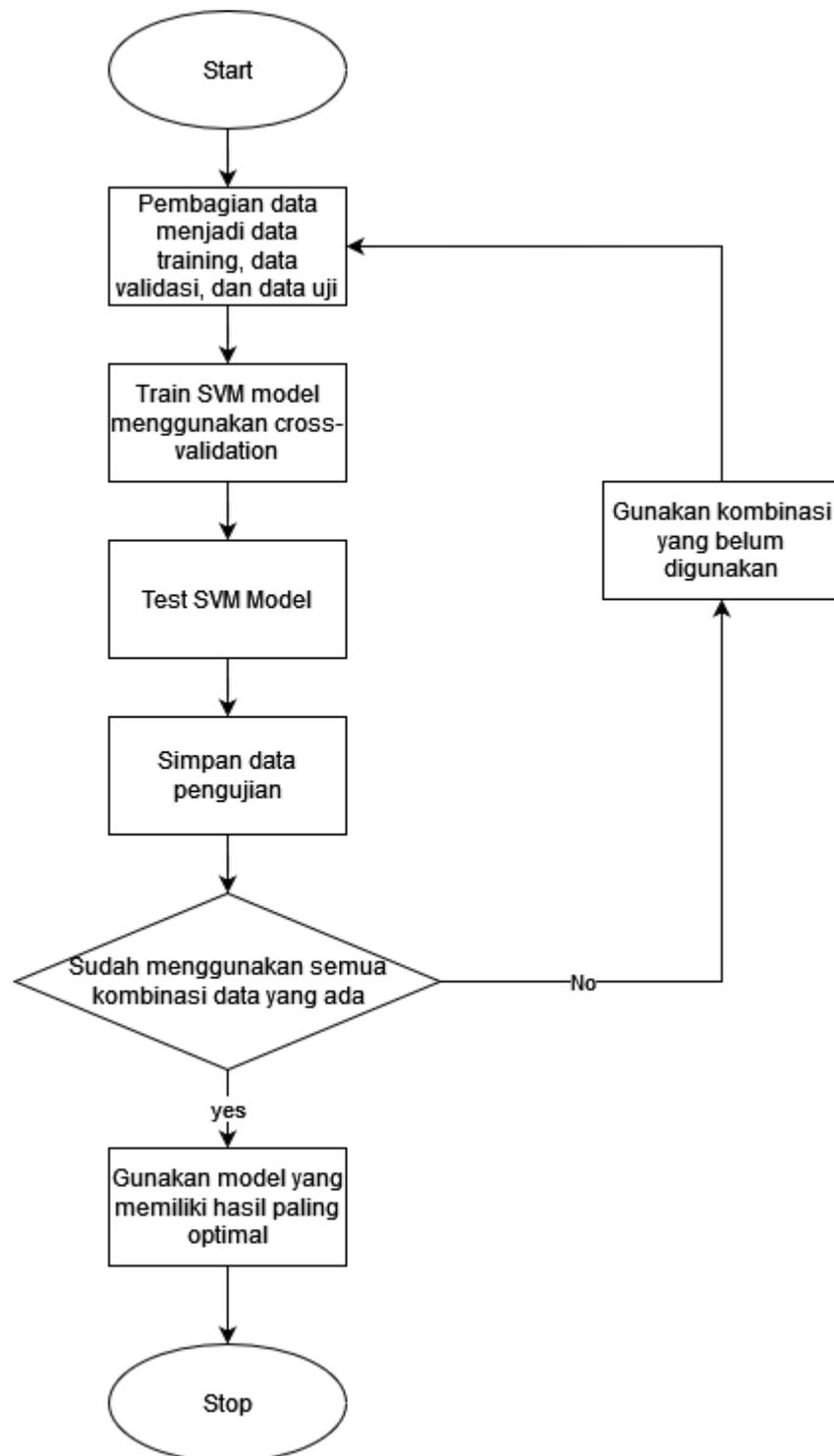
Pada tahapan ini data latih sejumlah 537 akan di *split* menjadi 2 jenis data latih dan data validasi sejumlah 5 *fold*. Kemudian dengan menggunakan data tersebut akan diuji untuk tiap pasang data untuk menemukan data pada *fold* berapa yang memiliki nilai performa paling optimal yang nantinya data latih pada *fold* tersebut akan menjadi *input testing data* pada tahap pengujian menggunakan data uji yang sudah dipisah sebelumnya. Untuk mendapatkan besaran data yang digunakan untuk data validasi dapat dihitung menggunakan persamaan 3.1 dan mendapatkan jumlah data yang digunakan sebagai data validasi adalah 108 dan data latih sejumlah 429 data. Implementasi dari *K-Fold Cross Validation* ditampilkan sebagai berikut. Representasi dan *flowchart* dari penerapan *K-fold cross validation* ditunjukkan pada Gambar 3.5 dan Gambar 3.6.

$$Jumlah\ data\ uji = \frac{1}{jumlah\ fold} \times jumlah\ dataset \quad (3.1)$$

$$\text{Jumlah data uji} = \frac{1}{5} \times 537 = 107.49$$

Train	Test	Test	Test	Test
Test	Train	Test	Test	Test
Test	Test	Train	Test	Test
Test	Test	Test	Train	Test
Test	Test	Test	Test	Train

Gambar 3. 4 Representasi dari metode Cross Validation



Gambar 3. 5 Flowchart K-Fold Cross Validation

```

fold = 1
for train_index, val_index in kf.split(p):
    print("lipatan ke " + str(fold))
    print("TRAIN:", train_index, "VAL:", val_index)
    p_train,p_val = p[train_index], p[val_index]
    o_train, o_val = o[train_index], o[val_index]

    tr = pd.DataFrame(p_train, o_train)
    tr.reset_index(drop=False, inplace=True)
    tr.columns =
['Outcome', 'Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'In
sulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']
    print("Data train")
    print(tr)
    tr.to_csv("/content/drive/MyDrive/Colab
Notebooks/NewSplitData/train_" + str(fold) + ".csv", index=True)

    ts = pd.DataFrame(p_val, o_val)
    ts.reset_index(drop=False, inplace=True)
    ts.columns =
['Outcome', 'Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'In
sulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']
    # ts = o_train
    print("Data validation")
    print(ts)
    ts.to_csv("/content/drive/MyDrive/Colab
Notebooks/NewSplitData/val_" + str(fold) + ".csv", index=True)

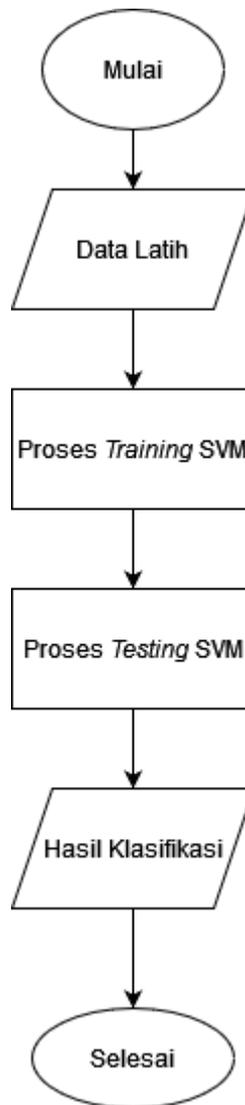
    fold += 1

```

3.3.4 Implementasi SVM

Implementasi SVM pada penelitian ini dilakukan menggunakan Bahasa pemrograman Python versi 3.7.10. Proses *testing* dan *training* dilakukan pada aplikasi Google Colaboratory. Dalam penerapan proses SVM yang pertama dilakukan yaitu memisahkan *dataset* menjadi *training data* serta *testing data* dengan perbandingan 70:30. *Testing data* digunakan untuk membangun model *Support Vector Machine*, lalu data *training* digunakan untuk menguji model yang sudah dibangun sebelumnya. Lalu selanjutnya melakukan tahapan *training*

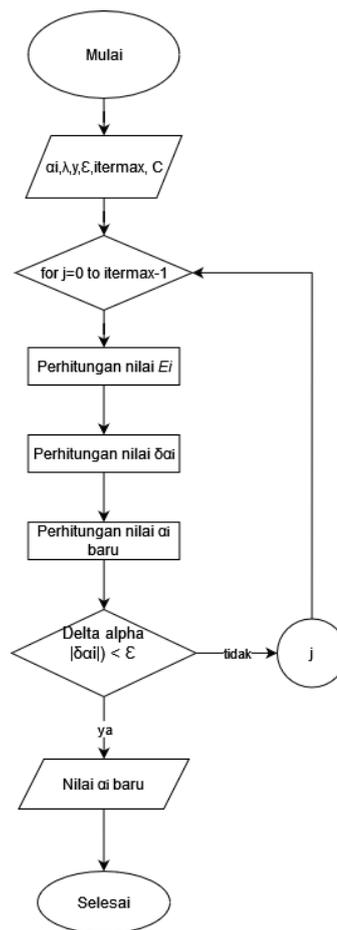
dengan menggunakan proses *Sequential Minimal Optimization*. Kemudian setelah model telah dibangun maka tahapan selanjutnya adalah pengujian menggunakan data uji yang sudah ada. Tahapan implementasi *Support Vector Machine* lebih jelasnya akan dijelaskan sebagai berikut.



Gambar 3. 6 Proses Support Vector Machine

Prosedur latih *Support Vector Machine* dilakukan guna mengklasifikasikan data diantara 2 *object* dari 2 *class* yang mempunyai *hyperplane*. *Support*

Vector Machine memiliki banyak proses latih antara lain, *Sequential Training*, *Sequential Minimal*, serta *Optimization Quadratic Programming*. Di *Quadratic Programming* waktu yang dibutuhkan untuk menyelesaikan proses terbilang lama serta *algorithm* yang *complex*. Peningkatan dari *Quadratic Programming* adalah *Sequential Minimal Optimization*, algoritma dalam metode ini juga cukup *complex* tetapi fokus pada masalah optimasi yang kecil.



Gambar 3. 7 *Flowchart* Proses *Training*

Contoh data *training* yang digunakan terdapat pada Tabel 3.4, yaitu data ke-1 sampai dengan data ke-5. Dalam data ke-1 dan ke-2 adalah data pasien yang tidak memiliki kemungkinan diabetes dengan *class* 1. Sedangkan

data ke-3 sampai ke-5 adalah data dari pasien yang memiliki kemungkinan diabetes berdasarkan gaya hidup dengan *class* -1.

Tabel 3. 4 Contoh Data Latih

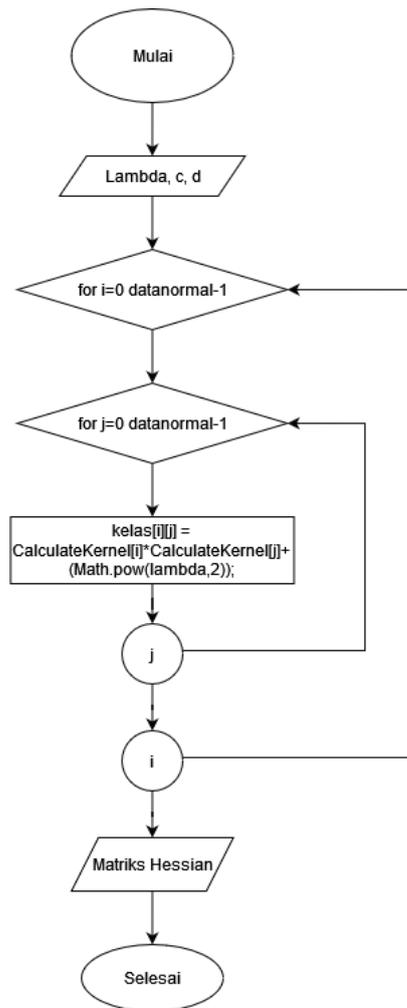
No.	Fitur a	Fitur b	Fitur c	Fitur d	Fitur e	Fitur f	Fitur g	Fitur h	Kelas	Outcome
1	0,353	0,353	0,353	0,353	0,353	0,353	0,353	0,353	Non Diabetes	1
2	0,059	0,059	0,059	0,059	0,059	0,059	0,059	0,059	Non Diabetes	-1
3	0,471	0,920	0,525	0,000	0,000	0,347	0,674	0,183	Diabetes	1
4	0,059	0,447	0,541	0,232	0,111	0,419	0,167	0,000	Diabetes	-1
5	0,000	0,688	0,328	0,354	0,199	0,642	0,002	0,200	Non Diabetes	1

Langkah perhitungan training pada metode SVM adalah sebagai berikut :

1. Inisialisasi $a_i = 0$, $\gamma = 0,000001$, $C = 1$, $\lambda = 0,5$
2. Mencari nilai matrik *Hessian* menggunakan persamaan

$$D_{ij} = y_i y_j (K(x_i x_j) + \lambda^2) \quad (3.2)$$

Menggunakan Persamaan 3.2 dilakukan perhitungan untuk mendapatkan nilai matriks *Hessian*.



Gambar 3. 8 *Flowchart* perhitungan matriks *Hessian*

Pada Tabel 3.5 merupakan nilai Matriks *Hessian* yang sudah dihitung.

Tabel 3. 5 Perhitungan Matriks *Hessian*

<i>Dij</i>	1	2	3	4	5
1	1,121972318	-0,41609	1,350944	-0,94765	1,101629
2	0,416089965	0,277682	-0,43349	0,366275	-0,03548
3	1,350944239	-0,43349	2,200759	-1,23101	1,316276
4	-0,94764986	0,366275	-1,23101	1,0158675	-1,10883
5	1,101628589	-0,39194	1,316276	-1,108826	1,448459

3. Setelah mendapat hasil matriks *Hessian* langkah berikutnya adalah mengkalkulasi nilai *error*.

a. Mencari nilai E_i menggunakan Persamaan 3.3.

$$E_i = \sum_{j=1}^n \alpha_i D_{ij} \quad (3.3)$$

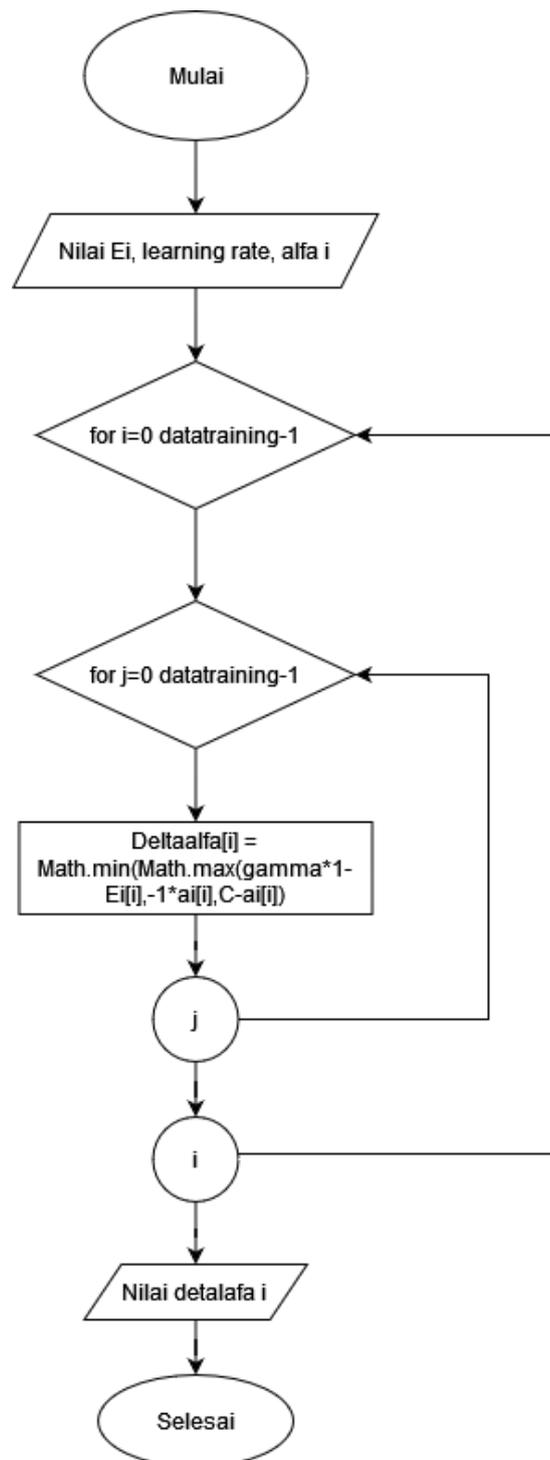
Penghitungan nilai E_i dijalankan untuk semua *training data*, *output* dari perhitungan tersebut disajikan pada Tabel 3.6.

Tabel 3. 6 Perhitungan Nilai E_i pada Iterasi ke 1

I	E_i (Nilai Error)
1	0,00
2	0,00
3	0,00
4	0,00
5	0,00

b. Mengkalkulasi nilai $\delta\alpha_i$ menggunakan Persamaan 3.4.

$$\delta\alpha_i = \min(\max[\gamma(1 - E_i), \alpha_i], C - \alpha_i) \quad (3.4)$$



Gambar 3. 9 Flowchart perhitungan nilai delta alfa

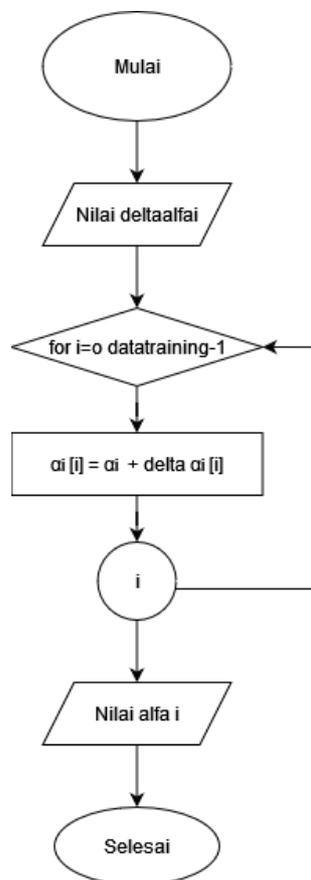
Hasil dari perhitungan $\delta\alpha_i$ dari contoh data latih yang digunakan dapat dilihat pada Tabel 3.7.

Tabel 3. 7 Perhitungan nilai $\delta\alpha_1$ pada Iterasi ke 1

I	$\delta\alpha_1$
1	0,1
2	0,1
3	0,1
4	0,1
5	0,1

c. Langkah selanjutnya adalah memperbarui nilai α_i , dapat dikalkulasi dengan persamaan

$$\alpha_i = \alpha_i + \delta\alpha_i \quad (3.4)$$



Gambar 3. 10 Flowchart perhitungan nilai alfa i

Hasil perhitungan nilai α_i yang baru dapat dilihat pada Tabel 3.8.

Tabel 3. 8 Perhitungan nilai α_i pada iterasi ke 1

I	α_i
1	0,1
2	0,1
3	0,1
4	0,1
5	0,1

4. Selanjutnya menghitung literasi tiap $i = 1, 2, 3, \dots, n$. Prosedur kalkulasi dilakukan literasi (*itermax*) sebanyak dua kali. Langkah-langkah perulangan dimulai dengan mengkalkulasi nilai *error*.

Masuk Iterasi ke-2

- a. Mengkalkulasi nilai E_i menggunakan Persamaan 3.3 juga inisialisasi α_i baru yaitu pada Tabel 3.8. *Output* perhitungan seluruh data pada iterasi ke-2 disajikan pada Tabel 3.9.

Tabel 3. 9 Perhitungan nilai E_i pada Iterasi ke 2

I	E_i
1	0,2
2	0,0
3	0,3
4	-0,2
5	0,2

- b. Selanjutnya menghitung nilai $\delta\alpha_i$ untuk iterasi ke dua menggunakan Persamaan 3.4. *Output* seluruh perhitungan data ditampilkan di Tabel 3.10.

Tabel 3. 10 Perhitungan nilai $\delta\alpha_i$ pada iterasi ke 2

I	$\delta\alpha_i$
1	-0,099982211
2	-0,09997976
3	-0,099983204
4	-0,099978096
5	-0,099982366

- c. Kemudian mengganti nilai α_i yang baru menggunakan persamaan 3.4 dimana nilai α_i diganti dengan angkat baru yaitu pada Tabel 3.8. Hasil dari penghitungan nilai α_i yang baru dapat dilihat pada tabel 3.11 dibawah ini.

Tabel 3. 11 Perhitungan nilai α_i pada iterasi ke 2

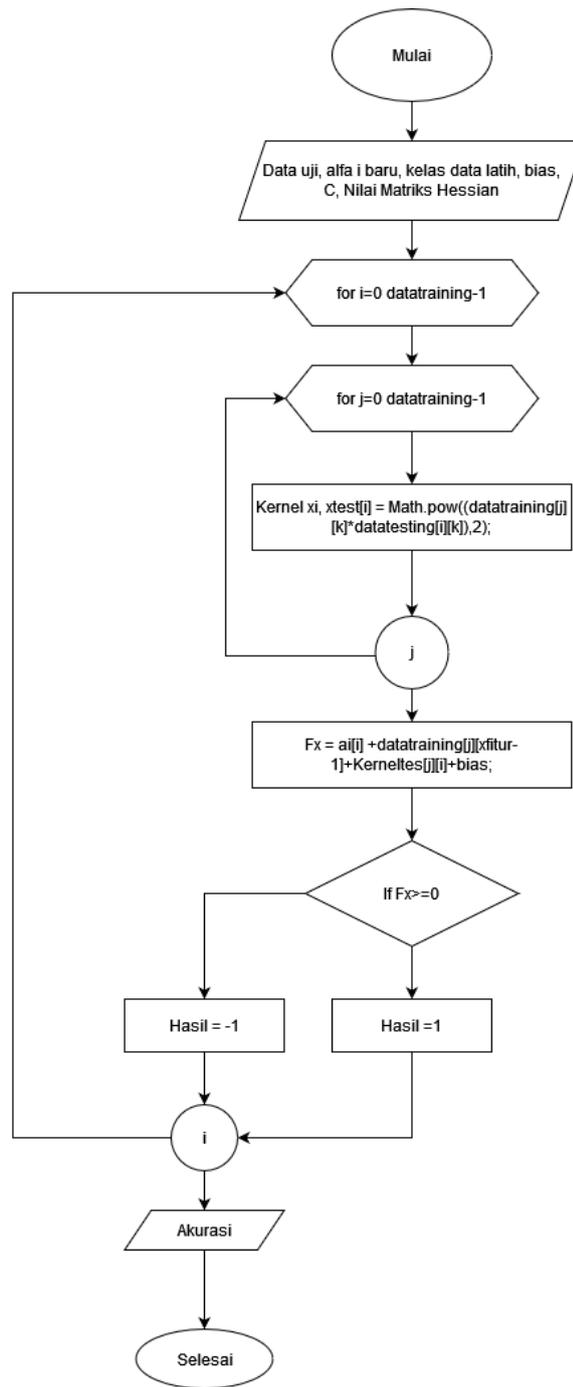
I	α_i
1	0,121
2	-0,124
3	0,220
4	-0,290
5	0,137

Setelah selesai melakukan proses *training* tahap selanjutnya adalah menjalankan proses uji untuk data *testing*. Tahap ini dijalankan untuk menghasilkan nilai $f(x)$ menggunakan Persamaan 2.6. *Testing data* yang digunakan dalam tahapan perhitungan ditampilkan di Tabel 3.12.

Tabel 3. 12 Data Uji

No.	Fitur a	Fitur b	Fitur c	Fitur d	Fitur e	Fitur f	Fitur g	Fitur h	Kelas	Outcome
6	0,294	0,583	0,607	0,000	0,000	0,382	0,202	0,150	Diabetes	-1
7	2	10	10	21	5	10	1	78	Non Diabetes	1

Sebelum menemukan nilai $f(x)$ dibutuhkan mencari nilai *bias*.. Nilai (x_i, x_+) ialah nilai *support vector* kelas yang paling besar, sedangkan nilai (x_i, x_-) ialah nilai *support vector* kelas paling kecil.



Gambar 3. 11 *Flowchart Proses Testing*

Contoh perhitungan untuk (x_i, x_+) didapatkan berdasarkan nilai a_i terbesar dari kelas positif, sedangkan nilai (x_i, x_-) didapatkan dari nilai a_i yang memiliki

nilai terbesar dari kelas negatif. Nilai tersebut ditentukan berdasarkan hasil matriks *Hessian* kolom 3 dan kolom 4, yang disajikan pada tabel berikut.

Tabel 3. 13 Nilai Matriks *Hessian*

<i>Dij</i>	3	4
1	1,350944	-0,94765
2	-0,43349	0,366275
3	2,200759	-1,23101
4	-1,23101	1,0158675
5	1,316276	-1,108826

Langkah selanjutnya yaitu menghitung nilai (x_i, x_+) dan (x_i, x_-) dengan menggunakan Persamaan 3.5.

$$K(x_i, x^+) = \sum_{i=1}^m y_i y_j K(x_i, x) \quad (3.5)$$

Setelah dihitung berikut merupakan nilai dari (x_i, x_+) dan nilai (x_i, x_-) , yang dapat dilihat pada Tabel 3.14.

Tabel 3. 14 Nilai X_+ dan X_-

x_+	0,39143
x_-	-0,32284

Setelah nilai (x_i, x_+) dan nilai (x_i, x_-) didapatkan, nilai *bias* dapat dihitung menggunakan persamaan 2.6.

Kemudian setelah diperoleh nilai *bias* maka dapat dihitung nilai $f(x)$ menggunakan persamaan 2.5. Hasil nilai (x_i, x_{test1}) dan nilai (x_i, x_{test2}) untuk seluruh contoh data uji dapat disajikan pada Tabel 3.15 berikut.

Tabel 3. 15 Perhitungan Nilai (x_i, x_{test1}) dan nilai (x_i, x_{test2})

$K(x_i, x_{testj})$	x_{test1}	x_{test2}
x1	0,782	48,353
x2	0,130	8,059
x3	1,288	33,829
x4	0,800	19,789
x5	0,876	40,606

Setelah nilai (x_i, x_{testj}) diperoleh, langkah selanjutnya yaitu mendapatkan nilai (x)

Setelah mendapat nilai $f(x)$ dari tiap kelas, maka hasil klasifikasi dapat dilihat dari nilai $f(x)$ tersebut. Jika $f(x) > 0$ artinya data tersebut kategori kelas 1, sedangkan $f(x) < 0$ berarti data tersebut terhitung kategori kelas -1. Hasil pengolahan data uji disajikan pada Tabel 3.16.

Tabel 3. 16 Hasil Klasifikasi

	Kelas Prediksi	Kelas Hasil
Fxtest1 0,50645	1	-1
Fxtest2 18,35665	1	1

```

#Model SVM kernel linear

from sklearn import metrics

for i in range (1,6):

    dataLatih_csv = "/content/drive/MyDrive/Colab
Notebooks/NewSplitData/train_" + str(i) + ".csv"

    latih = pd.read_csv(dataLatih_csv)

M_train=latih[['Pregnancies','Glucose','BloodPressure','SkinThickness','I
nsulin','BMI','DiabetesPedigreeFunction']].values

    w_train = latih['Outcome'].values

    dataTest_csv = "/content/drive/MyDrive/Dataset Skripsi/testMinMax.csv"

    test = pd.read_csv(dataTest_csv)

    M_test =
test[['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','
BMI','DiabetesPedigreeFunction']].values

    w_test = test['Outcome'].values

    clf = svm.SVC(kernel='linear', C = 1.0)

    clf.fit(M_train,w_train)

    Hasil_Klasifikasi_Linear = clf.predict(M_test)

    test['Prediction'] = Hasil_Klasifikasi_Linear

    test.to_csv('/content/drive/MyDrive/Colab
Notebooks/NewFinalResult/linear_result_test_new_'+str(i) + '.csv',
index=False)

```

3.4 Skenario Pengujian

Skenario pengujian akan memaparkan tahap-tahap yang akan dilakukan untuk pengujian dalam penelitian ini

1. *Input Dataset*
2. Data yang sudah *diinput* masuk ke tahap praproses
3. Pembagian *Dataset* menjadi data uji dan data latih

4. Pelatihan *Dataset*
5. Pengujian menggunakan kernel linear
 - Pengujian 8 fitur
 - Pengujian 7 fitur
 - Pengujian 6 fitur
 - Pengujian 5 fitur
 - Pengujian 4 fitur
 - Pengujian 3 fitur
 - Pengujian 2 fitur
 - Pengujian 1 fitur
6. Pengujian menggunakan kernel RBF
 - Pengujian 8 fitur
 - Pengujian 7 fitur
 - Pengujian 6 fitur
 - Pengujian 5 fitur
 - Pengujian 4 fitur
 - Pengujian 3 fitur
 - Pengujian 2 fitur
 - Pengujian 1 fitur
7. Pengujian menggunakan kernel Polynomial *deggre 3*
 - Pengujian 8 fitur
 - Pengujian 7 fitur
 - Pengujian 6 fitur

- Pengujian 5 fitur
 - Pengujian 4 fitur
 - Pengujian 3 fitur
 - Pengujian 2 fitur
 - Pengujian 1 fitur
8. Pengujian menggunakan kernel Polynomial *degre* 5
- Pengujian 8 fitur
 - Pengujian 7 fitur
 - Pengujian 6 fitur
 - Pengujian 5 fitur
 - Pengujian 4 fitur
 - Pengujian 3 fitur
 - Pengujian 2 fitur
 - Pengujian 1 fitur
9. Pengujian menggunakan kernel Polynomial *degre* 7
- Pengujian 8 fitur
 - Pengujian 7 fitur
 - Pengujian 6 fitur
 - Pengujian 5 fitur
 - Pengujian 4 fitur
 - Pengujian 3 fitur
 - Pengujian 2 fitur
 - Pengujian 1 fitur

10. Pengujian menggunakan kernel Polynomial *degre* 9
 - Pengujian 8 fitur
 - Pengujian 7 fitur
 - Pengujian 6 fitur
 - Pengujian 5 fitur
 - Pengujian 4 fitur
 - Pengujian 3 fitur
 - Pengujian 2 fitur
 - Pengujian 1 fitur

11. Pengujian menggunakan kernel Polynomial *degre* 11
 - Pengujian 8 fitur
 - Pengujian 7 fitur
 - Pengujian 6 fitur
 - Pengujian 5 fitur
 - Pengujian 4 fitur
 - Pengujian 3 fitur
 - Pengujian 2 fitur
 - Pengujian 1 fitur

Kemudian hasil dari proses pengujian tersebut akan dikalkulasi nilai akurasi, presisi, *recall* menggunakan *confusion matrix* seperti pada gambar berikut :

		PREDIKSI	
		TRUE	FALSE
AKTUAL	TRUE	TP <i>(True Positive)</i> <i>correct result</i>	FN <i>(False Negative)</i> <i>Missing result</i>
	FALSE	FP <i>(False Positive)</i> <i>Unexpected result</i>	TN <i>(True Negative)</i> <i>Correct absence of result</i>

Gambar 3. 12 Confusion Matrix

Penjelasan dari Tabel 3.12 adalah sebagai berikut.

1. *True Positive* artinya jumlah data yang bernilai positif dan diprediksi benar sebagai positif
2. *True Negative* artinya jumlah data yang bernilai negatif dan diprediksi sebagai negatif
3. *False Positive* artinya jumlah data yang bernilai negatif tetapi diprediksi sebagai positif
4. *False Negative* artinya jumlah data yang bernilai positif tetapi diprediksi sebagai negatif

Untuk memperoleh nilai akurasi, presisi, *recall* dan *f-measure* dilakukan perhitungan menggunakan rumus berikut:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.6)$$

Untuk memperoleh nilai *recall* dilakukan perhitungan dengan menggunakan rumus :

$$Recall = \frac{TP}{TP + FN} \quad (3.7)$$

Untuk mendapatkan nilai presisi menggunakan perhitungan dengan menggunakan rumus berikut:

$$Presisi = \frac{TP}{TP + FP} \quad (3.8)$$

BAB IV

UJI COBA DAN PEMBAHASAN

4.1 Langkah-Langkah Uji Coba

Tahapan evaluasi dalam *system* yang dibangun, dilakukan melalui serangkaian langkah uji coba guna menentukan tingkan kesalahan (*error*) pada model. Terdapat beberapa langkah atau tahapan yang dilakukan sebelum akhirnya diperoleh nilai kesalahan (*error*) dari model yang sedang di uji.

Dalam penelitian ini melakukan uji coba dalam mengklasifikasikan penderita diabetes berdasarkan data rekam medis dengan menerapkan *Support vector machine* (SVM). Untuk melakukan riset ini, langkah-langkah yang dilakukan terdiri dari tahapan-tahapan berikut:

1. Menyiapkan *dataset* yang berasal dari *dataset public* yang diambil dari *Kaggle*. Dalam riset ini, ada dua *class* yang digunakan, yakni kelas Diabetes dan kelas Non-Diabetes.
2. Kemudian data yang digunakan dipisahkan dua bagian yakni data latih serta data uji dengan jumlah data *training* sejumlah 537 dan data *testing* sebanyak 231. Lalu data *training* dibagi menjadi data *training* dan *validation* dari 537 data training diambil sebanyak 108 sebagai data *validation*.
3. Selanjutnya tahapan *training*, *validation*, dan *testing* dilakukan sebanyak 5 *fold*. Untuk tahapan ini menggunakan metode SVM dengan kernel *Linear*, *RBF*, *Polynomial 3 deegre*, *Polynomial 5 deegre*, *Polynomial 7 deegre*, *Polynomial 9 deegre*, dan *Polynomial 11 deegre*.

4. Dalam setiap *fold* dan tiap kernel akan diambil *fold* yang memiliki nilai akurasi paling baik dibanding *fold* lainnya.
5. Setelah semua *fold* selesai, hasil pengujian akan dikumpulkan kemudian mencari penggunaan kernel yang menghasilkan nilai paling optimal.

4.2 Hasil Training

Hasil *training* yang ditampilkan merupakan hasil dari *fold* yang memiliki nilai paling baik dibanding *fold* lainnya. Hasil ditampilkan yang ditampilkan merupakan hasil *training* menggunakan 8 fitur, 7 fitur, 6 fitur, 5 fitur, 4 fitur, 3 fitur, 2 fitur, dan 1 fitur. Hasil *training* disajikan dalam tabel berikut.

Tabel 4. 1 Tabel Hasil Training

Fitur	Akurasi Training
8	78.13%
7	78.13%
6	77.20%
5	75.75%
4	75.75%
3	75.99%
2	75.75%
1	65.26%

4.3 Hasil Uji Coba

Pada sub-bab ini, disajikan hasil uji coba yang telah dijalankan dengan menggunakan 8 hingga 1 fitur dengan menggunakan kernel Linear, RBF, Polynomial Derajat (*Polynomial degree*) 3, Polynomial Derajat (*Polynomial degree*) 5, Polynomial Derajat (*Polynomial degree*) 7, Polynomial Derajat (*Polynomial degree*) 9 dan Polynomial Derajat (*Polynomial degree*) 11. Uji coba dilakukan guna mendapatkan tingkat akurasi paling baik. Hasil yang ditampilkan merupakan hasil paling baik dari tiap *fold*.

1. Uji Coba 8 Fitur

Nilai performa sistem meliputi akurasi, presisi, *recall* ditunjukkan pada Tabel 4.2. Pada penggunaan 8 fitur nilai akurasi tertinggi 76.19% terdapat pada SVM yang menggunakan kernel RBF dan *Polynomial Degree 3*, nilai presisi paling tinggi 62.66% pada kernel RBF, dan untuk nilai *recall* paling tinggi 66.21 % SVM yang menggunakan kernel *Polynomial degree 3*.

Tabel 4. 2 Performa sistem menggunakan 8 fitur

Fitur	Kernel	Akurasi	Presisi	Recall
8	Linier	74.45%	58.42%	70.27%
	RBF	76.19%	62.66%	63.51%
	Polynomial Degree 3	76.19%	62.02%	66.21%
	Polynomial Degree 5	68.39%	50.64%	52.70%
	Polynomial Degree 7	58.44%	37.20%	43.24%
	Polynomial Degree 9	61.74%	42.42%	56.75%
	Polynomial Degree 11	60.17%	39.53%	45.94%

2. Uji Coba 7 Fitur

Nilai performa sistem meliputi akurasi, presisi, *recall* ditunjukkan pada Tabel 4.3. Pada penggunaan 7 fitur nilai akurasi, presisi, dan *recall* tertinggi yaitu 79.65%, 66.26% dan 74.32% terdapat pada SVM menggunakan kernel Linear.

Tabel 4. 3 Performa sistem menggunakan 7 fitur

Fitur	Kernel	Akurasi	Presisi	Recall
7	Linier	79.65%	66.26%	74.32%
	RBF	76.62%	63.15%	64.86%
	Polynomial Degree 3	76.62%	63.15%	64.86%
	Polynomial Degree 5	71.86%	56.33%	54.05%
	Polynomial Degree 7	69.26%	52.54%	41.89%
	Polynomial Degree 9	61.47%	41.75%	51.35%
	Polynomial Degree 11	60.17%	40.62%	52.70%

3. Uji coba 6 fitur

Nilai performa sistem meliputi akurasi, presisi, *recall* ditunjukkan pada Tabel 4.4. Pada penggunaan 6 fitur nilai akurasi tertinggi 76.62% terdapat pada SVM yang menggunakan kernel Linear, nilai presisi paling tinggi 63.51% pada kernel Linear, dan untuk nilai *recall* paling tinggi 67.56% SVM yang menggunakan kernel RBF.

Tabel 4. 4 Performa sistem menggunakan 6 fitur

Fitur	Kernel	Akurasi	Presisi	Recall
6	Linier	76.62%	63.51%	63.51%
	RBF	72.72%	56.27%	67.56%
	Polynomial Degree 3	74.45%	60.0%	60.81%
	Polynomial Degree 5	70.99%	54.21%	60.81%
	Polynomial Degree 7	64.93%	46.84%	70.27%
	Polynomial Degree 9	65.80%	46.75%	48.64%

	Polynomial Degree 11	62.77%	42.68%	42.29%
--	----------------------	--------	--------	--------

4. Uji coba 5 fitur

Nilai performa sistem meliputi akurasi, presisi, *recall* ditunjukkan oleh Tabel 4.5. Dalam penggunaan 5 fitur hasil akurasi tertinggi 75.32% terdapat pada SVM yang menggunakan kernel *Polynomial degree 3*, nilai presisi paling tinggi 63.49% pada kernel *Polynomial degree 3*, dan untuk nilai *recall* paling tinggi 58.10% SVM yang menggunakan kernel RBF.

Tabel 4. 5 Performa sistem menggunakan 5 fitur

Fitur	Kernel	Akurasi	Presisi	Recall
5	Linier	74.89%	62.90%	52.70%
	RBF	73.59%	58.90	58.10%
	Polynomial Degree 3	75.32%	63.49%	54.05%
	Polynomial Degree 5	68.83%	51.61%	42.24%
	Polynomial Degree 7	69.69%	53.70%	39.18%
	Polynomial Degree 9	70.56%	56.01%	37.83%
	Polynomial Degree 11	67.96%	50.0%	51.35%

5. Uji coba 4 fitur

Nilai performa sistem meliputi akurasi, presisi, *recall* ditunjukkan pada Tabel 4.6. Dalam penggunaan 4 fitur hasil akurasi tertinggi 74.89% terdapat pada SVM yang menggunakan kernel Linear, nilai presisi paling tinggi 63.04% pada kernel *Polynomial degree 7*, dan untuk nilai *recall* paling tinggi 54.05% SVM yang menggunakan kernel RBF.

Tabel 4. 6 Performa sistem menggunakan 4 fitur

Fitur	Kernel	Akurasi	Presisi	Recall
4	Linier	74.89%	62.90%	52.70%
	RBF	74.02%	60.60%	54.05%
	Polynomial Degree 3	73.59%	61.40%	47.29%
	Polynomial Degree 5	70.99%	57.7%	35.13%
	Polynomial Degree 7	73.16%	63.04%	39.18%
	Polynomial Degree 9	68.39%	51.06%	32.43%
	Polynomial Degree 11	66.66%	47.27%	35.13%

6. Uji coba 3 fitur

Nilai performa sistem meliputi akurasi, presisi, *recall* ditunjukkan pada Tabel 4.7. Dalam penggunaan 3 fitur akurasi tertinggi 74.89% terdapat pada SVM yang menggunakan kernel Linear, nilai presisi paling tinggi 69.38% pada kernel *Polynomial degree 7*, dan untuk nilai *recall* paling tinggi 60.81% SVM yang menggunakan kernel *Polynomial degree 9*.

Tabel 4. 7 Performa sistem menggunakan 3 fitur

Fitur	Kernel	Akurasi	Presisi	Recall
3	Linier	74.89%	62.90%	52.70%
	RBF	74.02%	62.06%	48.64%
	Polynomial Degree 3	76.62%	67.85%	51.35%
	Polynomial Degree 5	76.62%	69.23%	48.64%
	Polynomial Degree 7	76.19%	69.38%	45.94%
	Polynomial Degree 9	74.89%	60.81%	60.81%

	Polynomial Degree 11	62.33%	45.66%	37.27%
--	----------------------	--------	--------	--------

7. Uji coba 2 fitur

Nilai performa sistem meliputi akurasi, presisi, *recall* ditunjukkan pada Tabel 4.8. Dalam penggunaan 2 fitur hasil akurasi tertinggi 74.89% terdapat pada SVM yang menggunakan kernel Linear, nilai presisi paling tinggi 71.79% pada kernel *Polynomial degree 7*, dan untuk nilai *recall* paling tinggi 45.94% SVM yang menggunakan kernel RBF.

Tabel 4. 8 Performa sistem menggunakan 2 fitur

Fitur	Kernel	Akurasi	Presisi	Recall
2	Linier	74.89%	66.66%	43.24%
	RBF	74.45%	64.15%	45.94%
	Polynomial Degree 3	74.02%	63.46%	44.59%
	Polynomial Degree 5	75.75%	70.45%	41.89%
	Polynomial Degree 7	75.32%	71.79%	37.83%
	Polynomial Degree 9	71.86%	58.82%	40.54%
	Polynomial Degree 11	60.55%	40.32%	35.27%

8. Uji coba 1 fitur

Nilai performa sistem meliputi akurasi, presisi, *recall* ditunjukkan pada Tabel 4.9. Dalam penggunaan 2 fitur hasil akurasi tertinggi 69.69% terdapat pada SVM yang menggunakan kernel RBF, *Polynomial degree 3*, *Polynomial degree 7*, nilai presisi paling tinggi 55.26% pada kernel *Polynomial degree*

3, dan *Polynomial degree* 7, dan untuk nilai *recall* paling tinggi 52.70% SVM yang menggunakan kernel *Polynomial degree* 9.

Tabel 4. 9 Performa sistem menggunakan 1 fitur

Fitur	Kernel	Akurasi	Presisi	Recall
1	Linier	67.96%	0.0%	0.0%
	RBF	69.69%	55.26%	28.37%
	Polynomial Degree 3	69.69%	55.26%	28.37%
	Polynomial Degree 5	68.83%	54.16%	17.56%
	Polynomial Degree 7	69.69%	55.26%	28.73%
	Polynomial Degree 9	67.53%	49.36%	52.70%
	Polynomial Degree 11	55.77%	38.55%	30.22%

Dari uji coba yang sudah dilakukan dapat dilihat hasil uji coba yang memiliki nilai akurasi paling baik diantara uji coba yang lain adalah uji coba yang menggunakan kernel linear dan fitur sejumlah 7 yaitu *Pregnancies*, *Glucose*, *BloodPressure*, *SkinThickness*, *Insulin*, *BMI*, dan *DiabetesPedigreeFunction*. *Confusion Matrix* dari uji coba tersebut ditunjukkan oleh Tabel 4.10.

Tabel 4. 10 *Confusion Matrix* Uji Coba 7 Fitur Kernel Linear

		Prediksi	
		True	False
Aktual	True	55	19
	False	28	129

Kemudian dengan menggunakan *Confusion Matrix* yang sudah ada, dapat dihitung hasil akurasi sistem yang dibangun menggunakan Persamaan 3.6, nilai presisi dihitung menggunakan Persamaan 3.7, dan *recall* dengan menggunakan Persamaan 3.8. Hasil dari performa sistem menggunakan 7 fitur dan kernel *Linear* ditunjukkan oleh Tabel 4.11.

Tabel 4. 11 Performa Sistem (7 Fitur Kernel Linear)

Performa sistem (7 Fitur Kernel Linear)		
Akurasi	Presisi	Recall
79.65%	66.26%	74.32%

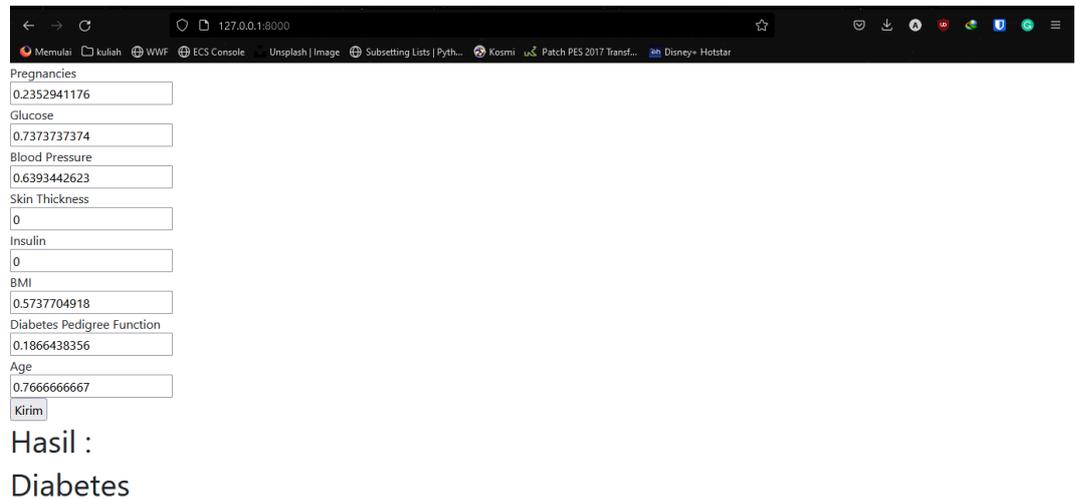
4.4 Implementasi Antarmuka

Sub-bab ini akan menampilkan halaman antarmuka pada sistem klasifikasi penderita diabetes, sistem yang dibangun menggunakan *web framework Django*. Tampilan halaman utama pada sistem yang dibangun ditunjukkan oleh Gambar 4.6.

Gambar 4. 1 Halaman Utama

Pada halaman utama terdapat *form* untuk *input* data berupa data rekam medis yang akan diklasifikasikan menggunakan model yang terdapat dalam

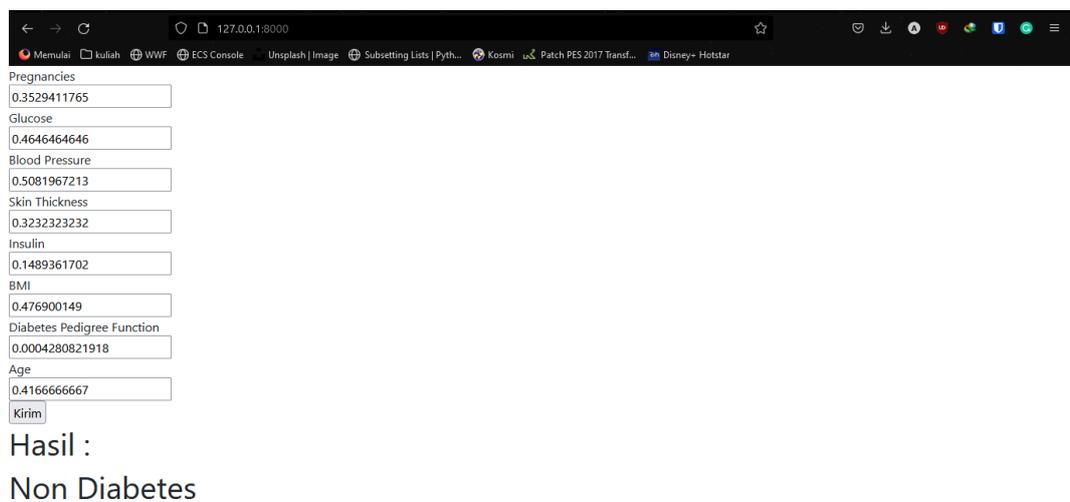
sistem. Setelah semua *form* terisi dan menekan *button* kirim sistem akan menghasilkan prediksi, jika hasilnya Diabetes maka hasil yang akan keluar ditunjukkan oleh pada Gambar 4.7 sedangkan jika hasilnya *Non-Diabetes* makak tampilan pada sistem ditunjukkan oleh Gambar 4.8.



Pregnancies
0.2352941176
Glucose
0.7373737374
Blood Pressure
0.6393442623
Skin Thickness
0
Insulin
0
BMI
0.5737704918
Diabetes Pedigree Function
0.1866438356
Age
0.7666666667
Kirim

Hasil :
Diabetes

Gambar 4. 2 Tampilan Klasifikasi Diabetes



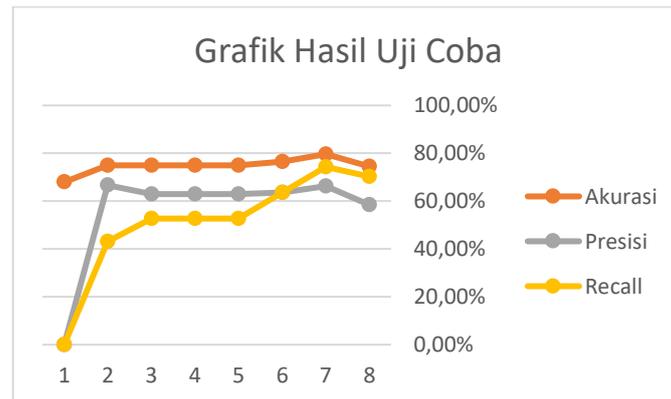
Pregnancies
0.3529411765
Glucose
0.4646464646
Blood Pressure
0.5081967213
Skin Thickness
0.3232323232
Insulin
0.1489361702
BMI
0.476900149
Diabetes Pedigree Function
0.0004280821918
Age
0.4166666667
Kirim

Hasil :
Non Diabetes

Gambar 4. 3 Tampilan Klasifikasi Non-Diabetes

4.5 Pembahasan

Berdasar hasil pengujian yang sudah dijalankan menggunakan kernel *Linear*, *RBF*, *Polynomial degree 3*, *Polynomial degree 5*, *Polynomial degree 7*, *Polynomial degree 9* dan *Polynomial degree 11* serta dari tiap kernel diuji menggunakan 8 fitur sampai 1 fitur guna mendapatkan fitur paling optimal untuk uji coba ini. Hasilnya percobaan yang memiliki hasil paling baik adalah uji coba menggunakan kernel *Linear* serta 7 fitur yaitu *Pregnancies*, *Glucose*, *BloodPressure*, *SkinThickness*, *Insulin*, *BMI*, dan *DiabetesPedigreeFunction*. Hasil dari percobaan tersebut seperti yang dicantumkan pada Tabel 4.4 adalah akurasi sebesar 79.65% nilai akurasi menunjukkan berapa persen pasien yang diprediksi diabetes dan tidak menderita diabetes dari keseluruhan data pasien, presisi sebesar 66.26% nilai dari presisi menunjukkan berapa persentase pasien yang didiagnosis menderita diabetes dari total pasien yang diprediksi menderita diabetes, sedangkan *recall* adalah nilai persentase pasien yang diprediksi menderita diabetes dibandingkan dengan total pasien yang sebenarnya menderita diabetes, untuk nilai *recall* sebesar 74.32%.



Gambar 4. 4 Grafik uji coba

Gambar 4.4 menunjukkan bagan hasil uji coba menggunakan jumlah fitur tertentu. Berdasarkan gambar tersebut dapat disimpulkan bahwa dalam klasifikasi yang menerapkan *algorithm Support Vector Machine*, hasil klasifikasi berbeda tergantung pada penggunaan jumlah fitur. Meskipun menggunakan jumlah fitur yang lebih banyak tidak menjamin hasil klasifikasi yang lebih baik, karena ada potensi terjadinya noise dalam proses klasifikasi. Di sisi lain, menggunakan jumlah fitur yang sedikit juga dapat menghilangkan *value* penting yang dapat meningkatkan kinerja klasifikasi.

Berdasarkan hasil uji coba telah dilakukan masih terdapat hasil yang tidak sesuai dengan data sebenarnya. Hal ini disebabkan oleh kurangnya tahap dalam *preproessing* data. Dalam data terdapat beberapa fitur yang memiliki variabel bernilai '0' yang harus ditangani lebih baik untuk penelitian selanjutnya. Beberapa fitur yang memiliki variabel bernilai '0' dikarenakan data yang digunakan merupakan data rekam medis sehingga jika pasien tersebut tidak memiliki gejala setelah dilakukan pengukuran maka nilai yang diantumkan adalah '0'.

Dari hasil penelitian yang sudah dilakukan diharapkan dapat menjadi sarana pengingat bagi masyarakat untuk selalu menjaga kesehatan. Kesehatan memegang peran yang penting dalam Islam dan merupakan anugerah yang harus dihargai oleh setiap individu. Terkait dengan pentingnya kesehatan, Nabi Muhammad SAW bersabda:

وَالْفَرَاغُ الصَّحَّةُ النَّاسِ مِنْ كَثِيرٍ فِيهِمَا مَعْبُونٌ نِعْمَتَانِ

“Dua kenikmatan yang sering dilupakan oleh kebanyakan manusia adalah kesehatan dan waktu luang.” (HR. Al-Bukhari: 6412, at-Tirmidzi: 2304, Ibnu Majah: 4170)

Ibnu Bathal menjelaskan bahwa inti dari hadits ini adalah seseorang tidak dianggap memiliki waktu luang sampai dia juga memiliki kondisi tubuh yang sehat. Jika seseorang memiliki kedua hal tersebut (waktu luang dan tubuh yang sehat), maka dia harus memiliki semangat agar tidak terperdaya dan melupakan rasa bersyukur atas segala nikmat yang diberikan Allah SWT. Salah satu bentuk syukur kepada Allah adalah dengan melaksanakan segala perintah-Nya dan menjauhi segala larangan-Nya. Orang yang tidak melakukan hal tersebut dianggap terperdaya (Hajar, et al., 2002);.

Dari dilakukannya penelitian Klasifikasi penderita diabetes berdasarkan data rekam medis menggunakan *Support Vector Machine* ini dan hasil yang sudah dihasilkan hendaknya bisa sebagai pengingat agar senantiasa bersyukur terhadap nikmat yang sudah diberikan Allah SWT. Salah satu nikmat besar yang diberikan oleh-Nya kepada umat manusia adalah nikmat sehat. Sehingga dengan masih

diberikannya nikmat tersebut. Sebagai seorang manusia, penting bagi kita untuk selalu bersyukur dengan cara menjalankan perintah-Nya dan senantiasa menghindari larangan-Nya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil yang didapat dari uji coba yang telah dilakukan sistem klasifikasi penderita diabetes berbasis data rekam medis menggunakan *Support Vector Machine* (SVM) menghasilkan nilai paling optimal dengan penggunaan 7 fitur dan kernel *Linear* fitur yang digunakan adalah *Pregnancies*, *Glucose*, *BloodPressure*, *SkinThickness*, *Insulin*, *BMI*, dan *DiabetesPedigreeFunction* dengan menghasilkan nilai akurasi sebesar 79.65%, presisi sebesar 66.26% serta *recall* 74.32%. Dari hasil uji coba yang sudah dilakukan menunjukkan bahwa kernel *linear* lebih efektif digunakan untuk data yang bisa dipisahkan secara linear dibanding kernel RBF dan *Polynomial*. Berdasarkan hasil identifikasi pada penelitian ini dengan penggunaan jumlah fitur yang berbeda maka *output* yang dihasilkan berbeda pula. Penggunaan jumlah fitur yang banyak bukan jaminan hasil klasifikasi akan baik juga, karna berpotensi terjadi *noise* pada proses klasifikasi. Namun dengan jumlah fitur yang minim juga dapat menghilangkan fitur kunci yang dapat membuat performa klasifikasi menjadi lebih baik.

Pemilihan data yang akan dimanfaatkan dalam riset juga berpengaruh terhadap nilai yang dihasilkan, hal ini dikarenakan terjadi ketimpangan antara data

yang memiliki nilai positif dan negatif sehingga sistem tidak bisa melakukan proses *training* secara optimal.

5.2 Saran

Berdasar pada hasil dari percobaan ini maka di penelitian selanjutnya diharapkan dapat mengoptimalkan hasil dari klasifikasi agar lebih akurat. Maka dari itu penulis memiliki saran untuk pengembangan penelitian dimasa mendatang, yaitu sebagai berikut :

1. Menggunakan *Dataset* yang lebih besar yang memiliki jumlah nilai positif dan negatif yang seimbang.
2. Melakukan percobaan dengan menggunakan metode lain seperti *Neural Network*, *Naïve Bayes* dan lain sebagainya
3. Menggunakan metode seleksi fitur untuk menemukan penggunaan fitur paling optimal.

DAFTAR PUSTAKA

- A. A., Hajar, I. & Ummah, G. A., 2002. *Fathul baari syarah shahih al bukhari*. Jakarta: Pustaka Azzam.
- Abedini, M., Bijari, A. & Banirostam, T., 2020. Classification of Pima Indian Diabetes Dataset using Ensemble of Decision Tree, Logistic Regression and Neural Network. *International Journal of Advanced Research in Computer and Communication Engineering*, 9(7).
- Alehegn, M., Joshi, R. R. & Mulay, P., 2019. Diabetes Analysis And Prediction Using Random Forest, KNN, Naïve Bayes, And J48: An Ensemble Approach. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, Volume 8.
- Anon., 2019. IDF Diabetes Atlas Ninth edition 2019.
- Apriyani, H. & Kurniati, 2020. *PERBANDINGAN METODE NAÏVE BAYES DAN SUPPORT VECTOR MACHINE DALAM KLASIFIKASI PENYAKIT DIABETES MELITUS (STUDI KASUS : RS. SITI KHADIJAH PALEMBANG)*. s.l., Bina Darma Conference on Computer Science .
- Ariani, A. & Samsuryadi, 2019. Klasifikasi Penyakit Ginjal Kronis menggunakan. *Prosiding Annual Research Seminar*, Volume V.
- Atthalla, I. N., Jovandy, A. & Habibie, H., 2018. Klasifikasi Penyakit Kanker Payudara. *Annual Research Seminar (ARS)*, Volume IV.
- Bramer, M., 2007. *Principles of Data Mining*. London: Springer.
- Chang, V., Bailey, J., Xu, Q. A. & Sun, Z., 2022. Pima Indians diabetes mellitus classification based on machine. *Neural Computing and Applications*, Volume IV.
- Cortes, C. & Vapnik, V., 1995. Support-Vector Networks. *Machine Learning*, Volume 20, pp. 273-297.
- Hairani, Saputro, K. E. & Fadli, S., 2020. K-means-SMOTE untuk menangani ketidakseimbangan kelas dalam klasifikasi penyakit diabetes dengan C4.5, SVM, dan naive Bayes. *Jurnal Teknologi dan Sistem Komputer*, 8(2), pp. 89-93.
- Han, J., 2000. *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann Publishers Inc.
- Han, J. & Kamber, M., 2006. *Data Mining: Concepts and Techniques Second Edition*. Elsevier.
- Indrayanti, Sugianti, D. & Al Karomi, M. A., 2017. Peningkatan Akurasi Algoritma KNN dengan Seleksi Fitur Gain Ratio. *IC-Tech*, XII(2).

- Irmanda, H. N. & Astriratma, R., 2020. Klasifikasi Jenis Pantun dengan Metode Support Vector Machines (SVM). *JURNAL RESTI*, 4(5), pp. 915-922.
- Isbandiyo, 2016. Penerapan Sequential Methods untuk Handling Missing Value pada Algoritma C 4.5 dan Naïve Bayes untuk Memprediksi Penyakit Diabetes Mellitus.
- Kaaffah, M. A. M., Hidayatno, A. & Soetrisno, Y. A. A., 2020. SISTEM KLASIFIKASI UKURAN BAJU DENGAN METODE SUPPORT VECTOR MACHINE (SVM). *TRANSIENT*, Volume 9.
- Kumar, S., Bhusan, . B., Singh, D. & Choubey, D. k., 2020. Classification of Diabetes using Deep Learning. *International Conference on Communication and Signal Processing*, pp. 652-655.
- Li, X., Ling, C. X. & Wang, H., 2015. The Convergence behavior of Naive Bayes on Large Sparse Datasets. *Journal ACM Transactions on Knowledge Discovery from Data (TKDD)*, Volume 11, pp. 853-858.
- Mucholladin, A. W., Bachtiar, F. A. & Furqon, M. T., 2021. Klasifikasi Penyakit Diabetes menggunakan Metode Support Vector Machine. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(2), pp. 622-633.
- Muhammad, H. et al., 2017. OPTIMASI NAÏVE BAYES CLASSIFIER DENGAN MENGGUNAKAN PARTICLE SWARM OPTIMIZATION PADA DATA IRIS. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, Volume 4, pp. 180-184.
- Muis, I. A. & Affandes, M., 2015. Penerapan Metode Support Vector Machine (SVM) Menggunakan Kernel Radial Basis Function (RBF) Pada Klasifikasi Tweet. *Jurnal Sains, Teknologi dan Industri*, Volume 12, pp. 189-197.
- Navin, P. & Singh, Y., 2015. Support vector machines for face recognition. *IRJET*, 02(08).
- Pangribowo, S., 2020. *Infodatin 2020 Diabetes*, s.l.: Pusdatin Kemkes.
- Purbolaksono, M. D., Tantowi, M. I., Hidayat, A. I. & Adiwijaya, 2021. Perbandingan Support Vector Machine dan Modified Balanced Random Forest dalam Deteksi Pasien Penyakit Diabetes. *JURNAL RESTI*, 5(2), pp. 393-399.
- Putra, A. L., Wowor, P. M. & Wungouw, H. I. S., 2015. GAMBARAN KADAR GULA DARAH SEWAKTU PADA MAHASISWA ANGKATAN 2015 FAKULTAS KEDOKTERAN UNIVERSITAS SAM RATULANGI MANADO. *Jurnal e-Biomedik (eBm)*, III(3).
- Riadi, A., 2017. Penerapan Metode Certainty Factor Untuk Sistem Pakar Diagnosa Penyakit Diabetes Melitus Pada Rsud Bumi Panua Kabupaten. *ILKOM Jurnal Ilmiah*, 9(3), pp. 309-316.

- Ritonga, A. S. & Purwaningsih, E. S., 2018. PENERAPAN METODE SUPPORT VECTOR MACHINE (SVM) DALAM KLASIFIKASI KUALITAS PENGELASAN SMAW (SHIELD METAL ARC WELDING). *Jurnal Ilmiah Edutic* , 5(1).
- Sisodia, D. & Sisodia, D. S., 2018. Prediction of Diabetes using Classification Algorithms. *Procedia Computer Science*, Volume 132.
- Wimmer, H., 2018. Effects of Normalization Techniques on Logistic Refression in Data Science. *Proceedings of the Conference on Informaation System Applied Research*, pp. 1-9.
- Wu, et al., 2007. Top 10 Algorithms in Data Mining.. Volume 14.