

BAB 2

TINJAUAN PUSTAKA

2.1 Komite Olahraga Nasional Indonesia (KONI)

Pada tahun 1938, Di tengah keresahan terhadap diskriminasi penggunaan fasilitas olahraga, para pemuda Indonesia mendirikan Ikatan Sport Indonesia (ISI). Berbentuk federasi, ISI beranggotakan PSSI (Perserikatan Sepak Bola Indonesia), Pelti (Perserikatan Lawn Tennis Indonesia), dan Perserikatan Bola Keranjang Seluruh Indonesia (PBKSI).

Tahun 1946, PORI (Persatuan Olahraga Republik Indonesia) sebagai badan olahraga bersifat nasional dan KORI (Komite Olimpiade Republik Indonesia) dibentuk oleh para Pemimpin olahraga eks pengurus GELORA . Tahun 1948, PONI diselenggarakan di Surakarta (9-9-1948). PORI dan KORI membentuk delegasi untuk menghadiri Olympic Games XIV, London, namun gagal karenasituasi politik. Tahun 1949, Kongres PORI III, Induk Organisasi mendapat hak otonomi, PORI sebagai badan koordinator. Tahun 1950, PORI diubah jadi Persatuan Olahraga Indonesia (PORI) KORI diubah jadi KOI (Komite Olimpiade Indonesia). Tahun 1951, Indonesia ikut serta dalam Asian Games I di New Delhi. Dalam persiapan tim ke Asian Games terjadi adanya tumpang tindih pelaksanaan tugas antara PORI dan KOI. Pada Kongres PORI – KOI bertepatan dengan PON II di Jakarta, dicapai kesepakatan bahwa demi efisiensi PORI melebur ke KOI. Ketua tetap Sri Sultan Hamengku Buwono IX. Tahun 1952, KOI mendapat pengakuan IOC dan untuk pertama kali Indonesia ikut serta pada Olympic Games XV Helsinki. Tahun 1959, DAGI (Dewan Asian Games Indonesia) dibentuk pemerintah. Tugas DAGI mempersiapkan penyelenggaraan Asian Games IV 1962. KOI sebagai badan pembantu DAGI dalam hubungan internasional. Tahun 1961, KOGOR (Komando Gerakan Olahraga) dibentuk pemerintah. Tugas KOGOR mempersiapkan pembentukan tim nasional Indonesia. Induk-induk organisasi olahraga sebagai pelaksana teknis cabang olahraga yang bersangkutan. KOGOR dibentuk ditiap daerah tingkat I, bertugas menggerakkan olahraga

membina bibit menunjang pembinaan olahraga nasional. Demokrasi dipimpin dengan pengerahan segenap “fund and forces” semaksimal mungkin.

Tahun 1962, DEPORA (Departemen Olahraga) dibentuk, dipimpin oleh Menteri Maladi. Asian Games IV diselenggarakan di Jakarta (24/8 - 4/9 - 1962).

Tugas DEPORA :

1. Mengelola pembibitan, pembinaan SDM (penataran pembina, pelatih, dsb)
2. Penelitian dan pengembangan.
3. Dukungan anggaran serta pembangunan dan pengembangan prasarana sarana olahraga.

Tahun 1963, Februari, KOI diskors oleh IOC karena tidak mengundang Israel dan Taiwan dalam Asian Games IV. Juni, skorsing dicabut oleh IOC. GANEFO I diselenggarakan di Jakarta (10 - 22/11-1963). Tahun 1964, DORI (Dewan Olahraga Republik Indonesia) dibentuk pemerintah. Semua organisasi, KOGOR, KOI, Induk organisasi olahraga dimasukkan dalam DORI. Tugas DORI:

1. Menetapkan kebijakan umum olahraga.
2. Membina dan mengawasi seluruh kegiatan olahraga.

Tahun 1965, Sekretariat Bersama Induk-induk Organisasi Cabang Olahraga dibentuk (25 Desember 1965), mengusulkan mengganti DORI menjadi Komite Olahraga Nasional Indonesia yang mandiri dan bebas dari pengaruh politik. Presidium Sekretariat Bersama (Sek Ber). Brig.Jen. Jonosewojo (PELTI). Kombes Pol. Tjoek Soejono (PABBSI). Drs. Ferry Sonnevile (PBSI). Kol. Saelan (PSSI) 1966- SK Presiden (Soekarno) No. 143 A dan 156 A tahun 1966 mengukuhkan dibentuknya KONI (“lama”) sebagai pengganti DORI. Badan baru ini tidak dapat berfungsi karena tidak didukung oleh Induk Organisasi Olahraga berkenaan situasi politik pada masa itu. Kabinet Ampera dibentuk oleh Presiden Soeharto. Depora dibubarkan dan dibentuk Direktorat Jenderal Olahraga dibawah Kementerian Pendidikan dan Kebudayaan. KONI (“baru”) dibentuk oleh Induk Organisasi Olahraga (31 Desember 1966) dengan Ketua Sri Sultan Hamengbuwono IX.

Tahun 1967, KONI dikukuhkan dengan SK Presiden (Soeharto) No. 57 Tahun 1967. Status KONI :

1. KONI adalah badan mandiri dan non pemerintah, artinya kegiatan olahraga kembali kepada masyarakat.
2. KONI sebagai mitra membantu pemerintah dibidang olahraga.
3. KONI tidak dikendalikan kelompok kekuasaan dan bebas dari kepentingan politik.

Tahun 1978, Dengan alasan efisiensi KONI – KOI menjadi satu, pengurusnya sama namun fungsinya berbeda. KONI melakukan pembinaan di dalam negeri, KOI melakukan kegiatan dalam hubungan luar negeri. Tahun 1981, MUSORNAS IV KONI mengesahkan AD/ART dengan menetapkan KONI ibarat sekeping mata uang dua sisi yang kedalam menjalankan tugasnya sebagai KONI dan keluar berstatus sebagai KOI. Kondisi tersebut berlangsung sampai keberadaan KONI saat ini. Tahun 2005, Pemerintah menerbitkan Undang-undang Nomor 3 Tahun 2005 Tentang Sistem Keolahragaan Nasional dan memecah KONI menjadi KON dan KOI. KON melakukan pembinaan dalam negeri dan penyelenggaraan Pekan Olahraga Nasional, KOI melakukan kegiatan pengiriman atlet keluar negeri dan penyelenggara pekan olahraga internasional di Indonesia. Tahun 2007, Pemerintah menerbitkan Peraturan Pemerintah Nomor 16, 17, dan 18 Tahun 2007 sebagai peraturan pelaksanaan UU No.3 Tahun 2005. KONI menyelenggarakan Musornas Luar Biasa (Musornaslub) antara lain mengesahkan anggaran dasar KONI dan KOI serta Rita Subowo sebagai Ketua Umum KONI dan KOI masa bakti 2007-2011.

Tahun 2010, Rakor di Surabaya, seluruh peserta KONI Provinsi merekomendasikan pembentukan Pokja Amandemen UU No. 3 Tahun 2005 tentang SKN dan penyatuan KONI dan KOI. Tahun 2011, Musornas KONI di NTB, Tono Suratman terpilih sebagai Ketua Umum KONI Pusat. Tahun 2012, RAT KONI memutuskan perlu adanya penyempurnaan AD/ART KONI. Dibentuk Pokja dari unsur KONI Pusat, KONI DKI Jaya, Kalbar, Sultra, PB Perbakin, dan PB IKASI. Pada Desember 2012, Rembuk Olahraga Nasional KONI di Balikpapan merekomendasikan perlunya pemerintah menetapkan aturan untuk KONI yang implementatif sebagai satu-satunya wadah organisasi dalam mencapai prestasi Nasional menuju prestasi Internasional.

2.2 Prediksi

Menurut kamus besar bahasa Indonesia (KBBI), prediksi adalah hasil dari kegiatan memprediksi atau meramal atau memperkirakan. Prediksi mempunyai makna sebagai sebuah informasi, pemberitahuan, peringatan, pengetahuan, ulasan tentang segala sesuatu yang akan terjadi berdasarkan fakta maupun tidak berdasarkan fakta

Dalam memprediksi kemenangan seorang atlet diperlukan informasi-informasi yang berkaitan dengan kondisi atlet saat itu. Pada penelitian ini, informasi-informasi yang digunakan dalam memprediksi kemenangan atlet, diantaranya adalah informasi mengenai kondisi kesehatan dan kondisi fisik atlet. Informasi mengenai kondisi kesehatan berupa data hasil *medical check up* dan informasi mengenai kondisi fisik berupa data hasil uji kekuatan fisik.

2.2.1 *Medical Check Up* (MCU)

Medical Check Up (MCU) merupakan pemeriksaan kesehatan secara keseluruhan yang dapat digunakan untuk mendeteksi gejala penyakit lebih dini. Jenis *medical check up* yang dilakukan atlet berupa pemeriksaan laboratorium yang meliputi pemeriksaan darah dan urine. Beberapa hasil pemeriksaan laboratorium diantaranya :

1. Hemoglobin (Hb)

Hemoglobin adalah suatu protein majemuk yang mengandung unsur non protein yaitu heme yang terdapat pada sel darah merah dan yang memberi warna merah pada darah[2]. Menurut ahli analis kesehatan Bapak Iwan Priyatno A.Md.A.K., hemoglobin berperan penting dalam tubuh yang berfungsi mensuplai oksigen ke seluruh tubuh. Dampak kekurangan Hb menyebabkan terjadinya anemia dan mengalami penyakit kronis. Dampak kelebihan Hb menyebabkan terjadinya kelainan pada darah.

2. Leukosit (Leu)

Leukosit adalah sel darah yang mengandung inti, disebut juga sel darah putih. Di dalam darah manusia, normal didapati jumlah leukosit rata-rata 5000-9000 sel/mm³ [3]. Menurut dr. Fajar Jaka Pratama sebagai salah satu dokter di RS. Palu Rotinsulu, kelebihan leukosit menandakan terjadinya infeksi (bakteri) di

dalam tubuh, sedangkan kekurangan leukosit menandakan terjadinya gangguan kekebalan di dalam tubuh.

3. Hematokrit (Ht)

Hematokrit adalah angka yang menunjukkan presentasi zat padat dalam darah terhadap cairan darah dan volume eritrosit yang dipisahkan dari plasma dengan dalam memutarinya di dalam tabung khusus yang nilainya dinyatakan dalam %. Hematokrit merupakan salah satu metode yang paling teliti dan simple dari Hb dan HE di dalam deteksi dan mengukur derajat anemia atau polycythemia[4]. Menurut dr. Fajar Jaka, kelebihan hematokrit menandakan seseorang mengalami kekentalan darah dan biasanya terjadi pada saat demam berdarah, sedangkan kekurangan hematokrit menandakan seseorang mengalami pengenceran darah dan biasa terjadi dalam keadaan seseorang terlalu banyak cairan di dalam tubuh.

4. Trombosit (Tr)

Trombosit adalah sel yang dibuat dalam sumsum tulang oleh sel-sel yang dinamakan megakariosit beserta tingkatan sel-sel yang lebih muda (pro megakariosit megakarioblast). Trombosit berfungsi :

1. untuk pembekuan darah dan retraksi bekuan.
2. Sebagai sumber pembentukan protrombin dan sebagai dari pembuatan trhomboastin.
3. Melindungi dinding-dinding pembuluh darah bagian dalam.
4. Menjaga daya tahan kapiler, kontraksi kapiler, dan sebagainya[4].

5. Basofil (Bs)

Basofil merupakan sitoplasma yang dipenuhi granula besar yang dengan pewarnaan dasar mengambil warna biru tua. Walaupun sel-sel ini berasal dari sumsum tulang seperti granulosit lainnya, mereka ini mempunyai banyak ciri yang sama dengan sel-sel tertentu dari jaringan penyambung yang disebut sel mast atau basofil jaringan[5]. Menurut dr. Fajar Jaka, kondisi basofil yang tinggi menyebabkan terjadinya alergi, sedangkan kondisi basofil yang rendah tidak memiliki pengaruh yang signifikan terhadap tubuh.

6. Eosinofil (Es)

Eosinofil adalah jenis granulosit lain yang dapat ditemukan dalam eksudat peradangan, walaupun biasanya dalam jumlah yang lebih sedikit. Eosinofil mempunyai inti yang tidak teratur sangat mirip dengan netrofil, tetapi granula sitoplasmanya berwarna merah cerah dengan zat warna eosin dan jauh lebih menyolok dari granula netrofil yang berwarna lembayung. Granula eosinofil sebenarnya merupakan paket-paket enzim sama seperti pada netrofil [5]. Menurut dr. Fajar Jaka Pratama, kondisi eosinofil yang tinggi menyebabkan alergi atau infeksi oleh cacing, sedangkan kondisi eosinofil yang rendah tidak memiliki pengaruh yang signifikan terhadap tubuh.

7. Batang (Bt)

Batang merupakan sel-sel pertama yang timbul dalam jumlah besar di dalam eksudat pada jam-jam pertama peradangan dan inti dari sel ini berlobus tidak teratur atau polimorf[5]. Menurut dr. Fajar Jaka Pratama, kondisi batang yang tinggi menunjukkan terjadinya gangguan pada pembentukan sel darah putih, sedangkan kondisi batang yang rendah menunjukkan pergeseran pembentukan jenis sel darah putih yang lain.

8. Segmen (Sg)

Segmen merupakan sel-sel pertama yang timbul dalam jumlah besar di dalam eksudat pada jam-jam pertama peradangan dan inti dari sel ini berlobus tidak teratur atau polimorf[5]. Menurut dr. Fajar Jaka Pratama, kondisi segmen yang tinggi menunjukkan terjadinya infeksi oleh bakteri, sedangkan kondisi segmen yang rendah menunjukkan terjadinya pergeseran pembentukan sel darah putih yang lainnya.

9. Limfosit (Lm)

Limfosit umumnya terdapat dalam eksudat dalam jumlah yang sangat kecil untuk waktu yang cukup lama, yaitu sampai reaksi peradangan menjadi kronik. Karena fungsi limfosit yang diketahui semuanya berada dalam bidang imunologis[5]. Menurut dr. Fajar Jaka Pratama, kondisi limfosit yang tinggi menunjukkan terjadinya infeksi oleh virus, sedangkan kondisi limfosit yang

rendah menunjukkan terjadinya pergeseran pembentukan sel darah putih yang lainnya.

10. Monosit (Mn)

Monosit merupakan bentuk leukosit yang berbeda dari granulosit karena susunan morfologi intinya dan sifat sitoplasmanya yang relatif agranular. Monosit berasal dari dalam susunan tulang seperti granulosit, tetapi umur sirkulasinya 3 sampai 4 kali lebih panjang daripada umur granulosit. Menurut dr. Fajar Jaka Pratama, kondisi limfosit yang tinggi menunjukkan terjadinya infeksi oleh virus, sedangkan kondisi limfosit yang rendah menunjukkan terjadinya pergeseran pembentukan sel darah putih yang lainnya.

11. LED

Laju Endap Darah (LED) adalah kecepatan mengendapnya eritrosit dari suatu monster atau sample darah yang diperiksa dalam suatu alat tertentu yang dinyatakan dalam mm per jam[4].

12. Glukosa Darah Puasa (GDP)

Glukosa adalah salah satu karbohidrat terpenting yang digunakan sebagai sumber tenaga utama dalam tubuh. Umumnya tingkat glukosa darah berada pada batas 70 – 150 mg/dl. Biasanya glukosa darah meningkat setelah makan dan umumnya pada pagi hari berada pada level terendah. Glukosa darah sewaktu adalah pemeriksaan kadar glukosa darah yang dilakukan setiap waktu sepanjang hari tanpa memperhatikan makanan terakhir yang dimakan dan kondisi tubuh orang tersebut. Glukosa darah puasa adalah pemeriksaan kadar glukosa darah yang dilakukan setelah pasien berpuasa selama 8 – 10 jam, sedangkan pemeriksaan glukosa 2 jam setelah makan adalah pemeriksaan yang dilakukan 2 jam diitung setelah pasien menyelesaikan makan[6]. Menurut dr. Fajar Jaka Pratama, kondisi GDP yang tinggi menyebabkan terjadinya gangguan pada metabolisme insulin dasar atau baseline, sedangkan kondisi GDP yang rendah menyebabkan ketidakseimbangan antara energi yang dikeluarkan dengan asupan makanan yang dikonsumsi.

13. Kolesterol Total (Chol)

Kolesterol adalah suatu zat lemak yang dibentuk oleh hati dan digunakan dalam pencernaan lemak. Selama pencernaan, kolesterol bergabung dengan garam empedu, fosfolipid, dan trigliserida menjadi suspensi kecil yang disebut misel. Kolesterol merupakan lemak yang penting namun jika terlalu berlebihan dalam darah dapat membahayakan kesehatan, bila ditinjau dari sudut kimiawi kolesterol diklasifikasikan ke dalam golongan lipid (lemak) berkomponen alkohol steroid[7]. Menurut ahli analis kesehatan Bapak Iwan Priyatno A.Md.A.K., dampak kelebihan kolesterol dalam tubuh adalah terjadinya penyempitan pembuluh darah karena banyak lemak jenuh yang menempel pada pembuluh darah.

14. Asam Urat (Au)

Asam urat merupakan hasil metabolisme akhir dari purin yaitu salah satu komponen asam nukleat yang terdapat dalam inti sel tubuh. Peningkatan kadar asam urat dapat mengakibatkan gangguan pada tubuh manusia seperti perasaan linulinu di daerah persendian dan sering disertai timbulnya rasa nyeri yang teramat sangat bagi penderitanya. Hal ini disebabkan oleh penumpukan Kristal di daerah tersebut akibat tingginya kadar asam urat dalam darah. Penyakit ini sering disebut penyakit gout atau lebih dikenal di masyarakat sebagai penyakit asam urat. Hiperuricemia disebabkan oleh sintesa purin berlebih dalam tubuh karena pola makan yang tidak teratur dan proses pengeluaran asam urat dari dalam tubuh yang mengalami gangguan. Faktor-faktor yang diduga juga mempengaruhi penyakit ini adalah diet, berat badan dan gaya hidup[8].

15. Trigliserida (Tg)

Trigliserida atau triasilgliserol adalah lemak utama dalam makanan manusia karena merupakan lemak simpanan utama dalam tumbuhan dan hewan yang menjadi makanan kita. Trigliserida atau triasilgliserol memiliki sebuah rangka gliserol tempat 3 asam lemak diesterkan. Trigliserida merupakan lipid utama yang terdapat didalam plasma manusia, yang lain adalah kolesterol, fosfolipid dan asam lemak non esterifikasi. Trigliserol, lemak utama dalam

makanan, terutama dicerna dalam lumen usus. Produk-produk pencernaan tersebut diubah kembali menjadi triasilgliserol didalam sel epitel usus, yang lalu dikemas dalam lipoprotein yang dikenal sebagai kilomikron, dan mensekresikan nya melalui limfe kedalam darah.

16. Ureum (Ur)

Ureum adalah suatu molekul kecil yang mudah mendifusi ke dalam cairan ekstrasel, tetapi pada akhirnya dipekatkan dalam urin dan diekskresi. Kadar ureum dan kreatinin yang tinggi dapat menyebabkan komplikasi tambahan yaitu menyebabkan syock uremikum yang dapat berlanjut menjadi kematian[9].

17. Kreatinin (Kr)

Kreatinin adalah suatu produk penguraian otot. Kreatinin diekskresikan oleh ginjal melalui filtrasi dan sekresi. Konsentrasi kreatinin dalam plasma relatif tetap dari hari ke hari dan konsentrasi kreatinin yang terkandung di dalam urin merupakan petunjuk penting terhadap kerusakan ginjal, diabetic nephropathy, dan laju filtrasi glomerular ginjal[8]. Menurut dr. Fajar Jaka Pratama, kondisi kreatinin yang tinggi menyebabkan terjadinya fungsi ginjal yang bersifat kronis.

18. SGOT

Enzim Transaminase atau disebut juga enzim aminotransferase adalah enzim yang mengkatalisis reaksi transaminasi. Terdapat dua jenis enzim serum transaminase yaitu serum glutamat oksaloasetat transaminase dan serum glutamat piruvat transaminase (SGPT). Pemeriksaan SGOT adalah indikator yang lebih sensitif terhadap kerusakan hati dibanding SGPT. Hal ini dikarenakan enzim GOT sumber utamanya di hati, sedangkan enzim GPT banyak terdapat pada jaringan terutama jantung, otot rangka, ginjal dan otak[10]. Menurut ahli analis kesehatan Bapak Iwan Priyatno A.Md.A.K., kondisi SGOT yang tinggi menyebabkan terjadinya kerusakan pada sel-sel di hati seperti penyakit hepatitis.

19. SPGT

SGPT (Serum Glutamic Pyruvic Transaminase) atau juga dinamakan ALT (Alanin Aminotransferase) merupakan enzim yang banyak ditemukan pada sel hati serta efektif untuk mendiagnosis destruksi hepatoseluler. Enzim ini dalam jumlah yang kecil dijumpai pada otot jantung, ginjal dan otot rangka. Pada umumnya nilai tes SGPT/ALT lebih tinggi daripada SGOT/AST pada kerusakan parenkim hati akut, sedangkan pada proses kronis didapat sebaliknya[10]. Kondisi SPGT yang tinggi menyebabkan terjadinya kerusakan pada hati.

2.2.2 Uji Kekuatan Fisik

Fisik merupakan penyesuaian dari kemajuan yang terbentuk dari teknik dan taktik. Kondisi fisik adalah suatu kesatuan yang utuh dari komponen-komponen yang tidak dapat dipisahkan begitu saja, baik peningkatan maupun pemeliharanya.

Kekuatan fisik yaitu suatu kemampuan kondisi fisik manusia yang diperlukan dalam peningkatan prestasi belajar gerak. Kekuatan merupakan salah satu unsur kondisi fisik yang sangat penting dalam berolahraga karena dapat membantu meningkatkan komponen-komponen seperti kecepatan, kelincahan dan ketepatan[11]. Mencapai prestasi yang maksimal membutuhkan kondisi fisik yang baik adalah salah satu faktor menentukan dalam mencapai suatu prestasi olahraga[12].

Kondisi fisik merupakan salah satu faktor penting dalam meningkatkan prestasi atlet. Sehingga kondisi fisik merupakan salah satu faktor penentu kemenangan atlet dalam bertanding. Dalam mengetahui kondisi fisik atlet perlu dilakukan beberapa tes uji fisik. Pada penelitian ini, uji fisik yang digunakan diantaranya adalah *Sit and Reach*, *Trunk Lift*, *Sit Up*, *Push Up*, *Back Lift*, *Hardle Jump*, dan *VO2 Max*.

2.3 Normalisasi

Proses normalisasi merupakan proses menstrukturkan data dalam cara tertentu untuk membantu mengurangi dan mencegah timbulnya masalah yang

berhubungan dalam pengolahan data . Pada atribut data cenderung memiliki nilai dengan rentang yang bervariasi. Perbedaan rentang yang besar dapat terjadi pada *record-record* karena memiliki nilai yang bervariasi. Atribut dengan nilai rentang yang besar, memiliki lebih banyak pengaruh dalam menentukan jarak atribut dengan nilai yang lebih kecil. Oleh karena itu, perlu dilakukan transformasi data berupa normalisasi terhadap nilai untuk membakukan data skala pengaruh yang ada pada atribut. Beberapa cara transformasi data yang dilakukan sebelum menerapkan suatu metode, yaitu sebagai berikut :

1. *Centering*

Centering pada data dilakukan dengan mengurangi setiap data dengan rata – rata dari setiap atribut yang ada. Secara matematis dapat dituliskan seperti berikut :

$$\hat{X} = X - \bar{X} \quad (2.1)$$

2. Normalisasi

Setelah data dilakukan *centering*, data tersebut dapat juga dibagi dengan standart deviasi dari atribut yang bersangkutan. Secara matematis dapat dituliskan sebagai berikut :

$$\hat{X} = \frac{X - \bar{X}}{\sigma_x} \quad (2.2)$$

3. *Scaling*

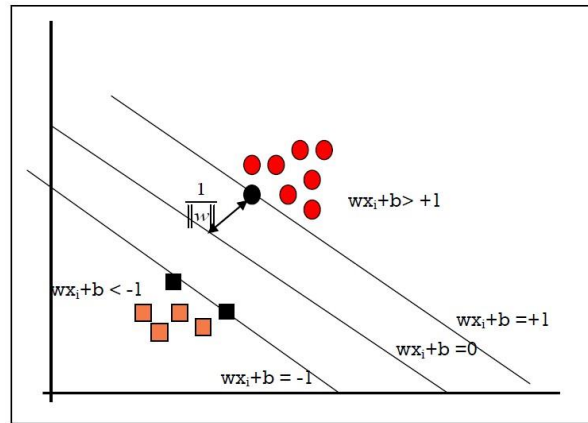
Scaling merupakan suatu prosedur merubah data sehingga data berada dalam skala tertentu, dimana skala bisa antara (0,1),(-1,1) atau yang lain sesuai yang dikehendaki. Untuk melakukan *scaling* dapat dilakukan dengan menggunakan rumus berikut[13] :

$$\hat{X} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2.3)$$

2.4 Metode *Support Vector Machine* (SVM)

Metode *Support Vector Machine* (SVM) merupakan salah satu metode *machine learning* yang memaksimumkan akurasi prediksi dengan mencari bidang pembatas (*hyperplane*) terbaik dari dua kelas dalam ruang fitur. Metode SVM membutuhkan data latih positif dan negatif. Data latih positif dan negatif ini

dibutuhkan SVM untuk membuat keputusan terbaik dalam memisahkan data positif dengan data negatif di ruang n-dimensi atau pembatas (*hyperplane*). Metode ini lebih dikenal dengan metode klasifikasi *supervised learning* untuk mencari garis pemisah *hyperplane* dengan mengoptimalkan *hyperplane*, dan memaksimalkan *margin* antara dua kelas. Data yang paling dekat dengan bidang pembatas disebut *support vector*[14].



Gambar 2. 1 Mencari fungsi pemisah yang optimal untuk obyek yang bisa dipisahkan secara linier

Metode SVM dalam mengklasifikasikan data uji ke dalam kelas positif dan negatif. Untuk kelas positif diberi label +1 yang artinya data uji tersebut memiliki potensi menang dalam bertanding, sedangkan untuk kelas negatif diberi label -1 yang artinya data uji tersebut memiliki potensi kalah dalam bertanding. Klasifikasi data uji dengan metode ini dilihat dari vektor data uji, jika nilai vektor melebihi nilai *hyperplane* maka data uji tersebut masuk ke dalam kelas positif dan jika tidak maka data uji tersebut masuk ke dalam kelas negatif.

Data pada ruang *input (input space) dataset* dinotasikan dengan $\{x_i, \dots, x_n\}$ sedangkan label kelas dinotasikan dengan $y_i \in \{-1, +1\}$ untuk $i = 1, 2, \dots, n$. Dimana n adalah banyaknya data. Diasumsikan kedua kelas -1 dan +1 dapat terpisah secara linear bidang pembatas[15], maka persamaan bidang pembatasnya didefinisikan pada persamaan (2.4) berikut :

$$w * x_i + b = 0 \quad (2.4)$$

Data x_i yang terbagi ke dalam dua kelas, yang termasuk kelas -1 (sampel negatif) didefinisikan sebagai vektor yang memenuhi pertidaksamaan (2.5) berikut :

$$w * x_i + b < 0 \text{ untuk } y_i = -1 \quad (2.5)$$

Sedangkan yang termasuk kelas +1 (sampel positif) memenuhi pertidaksamaan (2.6) berikut:

$$w * x_i + b > 0 \text{ untuk } y_i = +1 \quad (2.6)$$

Dimana:

x_i = data *input*

y_i = label yang diberikan

w = nilai dari bidang normal

b = posisi bidang relatif terhadap pusat koordinat

Parameter w dan b adalah parameter yang akan dicari nilainya. Bila label data $y_i = -1$, maka pembatas menjadi persamaan (2.7) berikut:

$$wx_i + b \leq -1 \quad (2.7)$$

Bila label data $y_i = +1$, maka pembatas menjadi persamaan (2.8) berikut:

$$wx_i + b \geq +1 \quad (2.8)$$

Margin terbesar dapat dicari dengan cara memaksimalkan jarak antar bidang pembatas kedua kelas dan titik terdekatnya, yaitu $2/|w|$. Hal ini dirumuskan sebagai permasalahan *quadratic programming* (QP) *problem* yaitu mencari titik minimal persamaan (2.9) dengan memperhatikan persamaan (2.10) berikut:

$$\text{minimize } \frac{1}{2} ||w||^2 \quad (2.9)$$

$$y_i(wx_i + b) - 1 \geq 0, (i = 1, \dots, n) \quad (2.10)$$

Permasalahan ini dapat dipecahkan dengan berbagi teknik komputasi. Lebih mudah diselesaikan dengan mengubah persamaan (2.9) ke dalam fungsi *Lagrangian* pada persamaan (2.11), dan menyederhanakannya menjadi persamaan (2.12) berikut:

$$L(w, b, a) = \frac{1}{2} ||w||^2 - \sum_{i=1}^n a_i (y_i ((w^T x_i + b) - 1)) \quad (2.11)$$

$$L(w, b, a) = \frac{1}{2} ||w||^2 - \sum_{i=1}^n a_i y_i (w^T x_i + b) + \sum_{i=1}^n a_i \quad (2.12)$$

Dimana a_i adalah *lagrange multiplier* yang bernilai nol atau positif ($a_i \geq 0$) dan $||w||^2 = w^T w$. Nilai optimal dari persamaan (2.12) dapat dihitung dengan meminimalkan L terhadap w , b dan memaksimalkan L terhadap a . Dapat dilihat pada persamaan (2.13) sampai (2.15) berikut:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^n a_i y_i x_i = 0 \quad (2.13)$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^n a_i y_i = 0 \quad (2.14)$$

$$\frac{\partial L}{\partial a} = \sum_{i=1}^n a_i y_i (w^T x_i + b) - \sum_{i=1}^n a_i = 0 \quad (2.15)$$

Maka masalah *Lagrange* dapat dinyatakan pada persamaan (2.16) berikut:

$$\text{Min } L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i y_i (w^T x_i + b) - \sum_{i=1}^n a_i \quad (2.16)$$

Dengan memperhatikan persamaan (2.17) dan (2.18) berikut:

$$w = \sum_{i=1}^n a_i y_i x_i \quad (2.17)$$

$$\sum_{i=1}^n a_i y_i = 0 \quad (2.18)$$

Jadi persoalan pencarian bidang model atau bidang pemisah terbaik persamaan (2.16) dengan memaksimalkan L terhadap a_i , persamannya menjadi persamaan (2.19) berikut:

$$\text{Max } \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1, j=1}^n a_i a_j y_i y_j^T x_i x_j^T \quad (2.19)$$

Dengan memperhatikan persamaan (2.20) berikut:

$$\sum_{i=1}^n a_i y_i = 0, a_i \geq 0 \ (i, j = 1, \dots, n) \quad (2.20)$$

Untuk mencari nilai x_i dapat dilakukan ketika sudah didapatkan nilai data masukan yang telah dinormalisasi dan diubah ke dalam bentuk format data svm, sedangkan nilai y_i merupakan label pada data masukan ditentukan oleh peraihan medali.

Pada data latih, untuk mendapatkan nilai a_i , langkah pertama adalah mengubah setiap data latih menjadi nilai vektor (*support vector*) = $\begin{pmatrix} x \\ y \end{pmatrix}$. Kemudian nilai vektor dari setiap data latih dimasukkan ke persamaan (2.21) kernel *trick phi* berikut:

$$S_i = \phi \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sqrt{x_n^2 + y_n^2} - x + (x - y)^2 \\ \sqrt{x_n^2 + y_n^2} - y + (x - y)^2 \end{bmatrix} \quad (2.21)$$

Nilai x didapatkan dari persamaan (2.23) kernel linear untuk x berikut:

$$\sum_{i=1, j=1}^n x_i x_j^T, (i, j = 1, \dots, n) \quad (2.22)$$

Nilai y didapatkan dari persamaan (2.24) kernel linear untuk y berikut:

$$\sum_{i=1, j=1}^n y_i y_j^T, (i, j = 1, \dots, n) \quad (2.23)$$

Untuk mendapatkan jarak tegak lurus yang optimal dengan mempertimbangkan vektor positif, maka hasil perhitungan dari substitusi nilai x dan nilai y ke persamaan (2.21) diberi nilai bias = 1 [14]. Kemudian cari parameter a_i , dengan terlebih dahulu mencari nilai fungsi setiap data latih menggunakan persamaan (2.25), lalu mencari nilai a_i pada persamaan linear menggunakan persamaan (2.26) dengan memperhatikan $i, j = 1, \dots, n$ berikut:

$$\sum_{i=1, j=1}^n a_i S_i^T S_j \quad (2.24)$$

$$\sum_{i=1, j=1}^n a_i S_i^T S_j = y_i \quad (2.25)$$

Setelah parameter a_i didapatkan, kemudian masukan ke persamaan (2.26) berikut:

$$\tilde{W} = \sum_{i=1}^n a_i S_i \quad (2.26)$$

Hasil yang didapatkan menggunakan persamaan (2.26), selanjutnya digunakan persamaan (2.27) untuk mendapatkan nilai w dan b :

$$y = wx + b \quad (2.27)$$

Sedemikian sehingga didapatkanlah nilai w dan nilai b sebagai model fitur untuk mengklasifikasikan kedua kelas.

2.5 Confusion Matrix

Pada tahap evaluasi bertujuan untuk mengetahui tingkat akurasi dari hasil penggunaan metode *support vector machine* dengan cara menghitung jumlah data uji yang kelasnya diprediksi secara benar. Adapun cara mengukur kinerja dari sistem akan dilakukan evaluasi dengan menggunakan *confusion matrix*. *Confusion matrix* adalah matriks yang berisi data aktual dan data prediksi yang telah diklasifikasikan oleh sistem.

Tabel 2. 1 Confusion Matrix

<i>Predictive</i>	<i>Actual</i>	
	<i>Negative</i>	<i>Positive</i>
<i>Negative</i>	TN	FN
<i>Positive</i>	FP	TP

Keterangan :

1. *True Positive* (TP) adalah kasus positif yang diklasifikasikan secara benar dan hasilnya positif.
2. *True Negative* (TN) adalah kasus negatif yang diklasifikasikan secara benar dan negatif.
3. *False Positive* (FP) adalah kasus dengan kelas negatif yang diklasifikasikan pada kelas positif
4. *False Negative* (FN) adalah kasus dengan kelas positif yang diklasifikasikan pada kelas negatif[13].

Performance Measurement (pengukuran kinerja) adalah suatu proses penilaian kemajuan pekerjaan terhadap tujuan dan sasaran yang telah ditentukan sebelumnya. Setelah memecahkan masalah optimasi yang tercantum dalam bagian sebelumnya, pengguna dapat menerapkan keputusan fungsi untuk memprediksi label (nilai targetss) pengujian data[16]. Adapun perhitungan *Performance Measuremen* dapat dihitung dari tabel 2.1 Tabel *Confusion Matrix* dengan persamaan (2.29).

$$Accuracy = \frac{TF+TN}{TP+TN+FP+FN} \times 100\% \quad (2.28)$$

2.6 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek (Inggris: *object-oriented programming* disingkat OOP) merupakan paradigma pemrograman yang berorientasikan kepada objek. Ini adalah jenis pemrograman di mana programmer mendefinisikan tidak hanya tipe data dari sebuah struktur data, tetapi juga jenis operasi (fungsi) yang dapat diterapkan pada struktur data. Dengan cara ini, struktur data menjadi objek yang meliputi data dan fungsi. Selain itu, pemrogram dapat membuat hubungan antara satu benda dan lainnya. Sebagai contoh, objek dapat mewarisi karakteristik dari objek lain.

Salah satu keuntungan utama dari teknik pemrograman berorientasi obyek atas teknik pemrograman prosedural adalah bahwa memungkinkan programmer untuk membuat modul yang tidak perlu diubah ketika sebuah jenis baru objek ditambahkan. Seorang pemrogram hanya dapat membuat objek baru yang

mewarisi banyak fitur dari objek yang sudah ada. Hal ini membuat program *object-oriented* lebih mudah untuk memodifikasi. Konsep dasar berorientasi objek diantaranya :

1. Kelas (*Class*) adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dari kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan dan kelas semula dapat diwariskan ke kelas yang baru.
2. Objek (*Object*) adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.
3. Metode (*Method*) adalah operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada terstruktur. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri.
4. Atribut (*Attribute*) dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.
5. Abstraksi (*Abstraction*) merupakan prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.
6. Enkapsulasi (*Encapsulation*) adalah pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerja.

7. Pewarisan (*Inheritance*) adalah mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya.
8. Antarmuka (*Interface*) sangat mirip dengan kelas, tetapi tanpa atribut kelas dan tanpa memiliki metode yang dideklarasikan. Antarmuka biasanya digunakan agar kelas lain tidak langsung mengakses ke suatu kelas.
9. Generalisasi dan Spesialisasi menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil dan motor.
10. Komunikasi antar objek dilakukan lewat pesan (*message*) yang dikirim dan satu objek ke objek lainnya.
11. Polimorfisme (*Polymorphism*) adalah kemampuan suatu objek untuk digunakan dibanyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.
12. *Package* adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda [18].

2.7 Unified Modeling Language (UML)

Pendekatan pembangunan perangkat lunak yang digunakan dalam penelitian ini adalah pendekatan berorientasi objek menggunakan *unified modeling language*. *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. *Unified Modeling Language* (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML

dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem [19]. Adapun UML yang digunakan dalam pembangunan perangkat lunak pada penelitian ini, yaitu :

2.7.1 Use Case Diagram

Use Case Diagram untuk memodelkan proses bisnis. Dalam konteks UML, tahap konseptualisasi dilakukan dengan pembuatan *use case diagram* yang sesungguhnya merupakan deskripsi peringkat tinggi bagaimana perangkat lunak (aplikasi) akan digunakan oleh penggunaanya. Selanjutnya, *use case diagram* tidak hanya sangat penting pada analisis, tetapi juga sangat penting untuk perancangan (*design*), untuk mencari (mencoba menemukan) kelas-kelas yang terlibat dalam aplikasi, dan untuk melakukan pengujian (*testing*).

Membuat *use case diagram* yang komprehensif merupakan hal yang sangat penting dilakukan pada tahap analisis. Dengan menggunakan *use case diagram*, akan mendapatkan banyak informasi yang sangat penting yang berkaitan dengan aturan –aturan bisnis. Dalam hal ini, setiap objek yang berinteraksi dengan sistem/perangkat lunak (misalnya, orang, suatu perangkat keras, sistem lain, dan sebagainya) merupakan *actor* untuk sistem/perangkat lunak yang dibangun, sementara *use case* merupakan deskripsi lengkap tentang bagaimana sistem/perangkat lunak berperilaku untuk para *actor*-nya. Dengan demikian, *use case program* merupakan deskripsi lengkap tentang interaksi yang terjadi antara para *actor* dengan sistem/perangkat lunak yang sedang dibangun.

Ketika mengembangkan *use case diagram*, hal yang pertama dilakukan adalah mengenali *actor* untuk sistem/aplikasi yang sedang dikembangkan. Dalam hal ini, ada beberapa karakteristik untuk para *actor*, yaitu *actor* ada diluar sistem yang sedang dikembangkan dan *actor* berinteraksi dengan sistem yang sedang dikembangkan[19].

2.7.2 Activity diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin

terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Use case diagram merupakan gambaran menyeluruh dan pada umumnya sangatlah tidak terperinci. Oleh karena itu, harus diperinci perilaku sistem untuk masing-masing *use case* yang ada. Penggunaan *activity diagram* dapat memberikan gambaran secara menyeluruh terhadap perangkat lunak yang dibangun[19].

2.7.3 Class diagram

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

Class diagram menggambarkan interaksi dan relasi antar kelas yang ada di dalam suatu sistem. Kelas memiliki atribut dan metode. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Atribut dan metode dapat memiliki salah satu sifat sebagai berikut :

1. *Private*, tidak dapat dipanggil dari luar kelas yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh kelas yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja.

Class diagram menggambarkan relasi atau hubungan antar kelas dari sebuah sistem. Berikut ini beberapa gambaran relasi yang ada dalam *class diagram*:

1. *Association*

Hubungan antar *class* yang statis. *Class* yang mempunyai relasi asosiasi menggunakan *class* lain sebagai atribut pada dirinya.

2. *Aggregation*

Relasi yang membuat *class* yang saling terikat satu sama lain namun tidak terlalu berkegantungan.

3. *Composition*

Relasi agregasi dengan mengikat satu sama lain dengan ikatan yang sangat kuat dan saling berkegantungan.

4. *Dependency*

Hubungan antar *class* dimana *class* yang memiliki relasi *dependency* menggunakan *class* lain sebagai atribut pada *method*.

5. *Realization*

Hubungan antar *class* dimana sebuah *class* memiliki keharusan untuk mengikuti aturan yang ditetapkan *class* lainnya[19].

2.7.4 *Sequence diagram*

Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. *Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya *sequence diagram* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup pada *sequence diagram* sehingga semakin banyak *use case* yang didefinisikan maka

sequence diagram yang harus dibuat juga semakin banyak. Penomoran pesan berdasarkan urutan interaksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu[19].

2.8 Pemrograman Java

Bahasa pemrograman yang digunakan dalam penelitian ini adalah bahasa pemrograman Java. Java adalah nama sebuah bahasa pemrograman yang sangat terkenal. Sebagai bahasa pemrograman, Java dapat digunakan untuk menulis program. Sebagaimana diketahui, program adalah kumpulan instruksi yang ditujukan untuk komputer. Melalui program, komputer dapat diatur agar melaksanakan tugas tertentu sesuai yang ditentukan oleh pemrogram.

Menurut Sun Microsystem, Java adalah sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada *computer stand alone* ataupun pada lingkungan jaringan. Seperti halnya C++, Java juga merupakan bahasa yang berorientasi objek. Dengan demikian, Java juga memudahkan dalam pembuatan aplikasi yang berskala besar[20].

2.9 Software Pendukung

Pada penelitian yang dilakukan diperlukan *software* pendukung yang membantu dalam pembangunan aplikasi. Adapun *software* pendukung, diantaranya:

2.9.1 NetBeans IDE

NetBeans IDE (*Integrated Development Environment*) adalah *open-source project* yang dedikasikan untuk pembangunan perangkat lunak. NetBeans IDE dapat juga berupa *Integrated Design Environment* dan *Integrated Debugging Environment*, yakni sebuah program atau alat bantu yang terdiri atas *Editor*, *Compiler*, *Debugger* dan *Design* yang terintegrasi dalam satu aplikasi. Netbeans IDE ditujukan untuk memudahkan dalam penggunaan bahasa pemrograman Java[21].

Untuk memakai Netbeans IDE maka, sebelumnya harus terlebih dahulu mempunyai *driver* JDK yang akan mendukung pembuatan perangkat lunak

dengan menggunakan Netbeans IDE, sehingga sebelum meng-*install* Netbeans, terlebih dahulu sebelumnya harus meng-*install driver* JDK.

2.9.2 Java Development Kit

JDK (Java Development Kit) adalah paket fungsi API untuk bahasa pemrograman Java, meliputi Java Runtime Environment (JRE) dan Java Virtual Machine (JVM). Dimana JDK juga merupakan *software development kit* yang merupakan alat bantu yang digunakan untuk membuat, manajemen dan membangun berbagai aplikasi dalam bahasa pemograman Java[22].

