

SISTEM PAKAR IDENTIFIKASI CEDERA PADA PEMAIN SEPAK BOLA MENGGUNAKAN METODE FORWARD CHAINING PADA PERSATUAN SEPAK BOLA INDONESIA PASER (PERSIPAS)

Azdar Irawan¹⁾, Ita Arfyanti²⁾, Heny Pratiwi³⁾

¹⁾Teknik Informatika, STMIK Widya Cipta Dharma

¹⁾Jl. M. Yamin No.25, Samarinda, 75123

E-mail : azdarirawan777@gmail.com¹⁾, qonita23@yahoo.com²⁾, henypratiwi@gmail.com³⁾

ABSTRAK

Tujuan dari pembuatan Sistem Pakar Identifikasi Cedera Pada Pemain Sepakbola ini adalah untuk memberikan pengetahuan dan informasi sekaligus diagnosa cedera kepada atlet tanpa harus berkonsultasi terlebih dahulu kepada fisioterapis atau pakarnya.

Metode yang akan digunakan dalam penelitian ini adalah metode *Forward chaining* untuk menentukan jenis-jenis cedera yang diderita oleh pasien. Input yang dibutuhkan adalah gejala-gejala yang dialami oleh pasien. Nilai akan diperoleh dari aturan untuk gejala atau ciri-ciri akan digabungkan. Hasil dari penggabungan ini merupakan output solusi cedera pada pemain sepakbola.

Sistem pakar mengidentifikasi cedera pada pemain sepakbola ini dapat membantu pengguna dalam hal mengetahui secara tepat tentang nama cedera yang dialami berdasarkan pilihan gejala-gejala cederanya sehingga dapat dilakukan pengobatan secara cepat dan tepat.

Kata kunci : Sistem Pakar, Cedera Sepakbola, Forward Chaining

1. PENDAHULUAN

Sebagai pemain sepak bola diharuskan memiliki kondisi fisik yang prima karena merupakan faktor yang utama selain aspek lain seperti aspek teknik, taktis, dan mental. Dalam olahraga sepak bola, benturan-benturan fisik antar pemain merupakan hal yang sulit untuk dihindari dan tidak jarang dari terjadinya benturan fisik tersebut dapat membuat seorang pemain sepak bola mengalami cedera.

Cedera yang sering dialami oleh pemain sepakbola adalah *sprain*, *strain*, dan *anterior cruciate ligament*. Cedera secara umum disebabkan oleh pemanasan yang kurang atau berlebihan, gaya permainan dan gerakan yang salah, dan pada proses pendinginan yang tidak tepat. Cedera ringan dapat menyebabkan pemain harus beristirahat kurang dari sepekan, cedera sedang dapat menyebabkan pemain harus beristirahat selama sepekan hingga sebulan, dan cedera berat dapat menyebabkan pemain harus beristirahat lebih dari sebulan. Untuk penanganan pertama jika terkena cedera, istirahatkan kaki dari pergerakan yang tidak penting atau berlebihan dan kompres dengan es untuk mengurangi bengkak dan memar.

Cedera olahraga jika tidak ditangani dengan cepat dan benar dapat mengakibatkan gangguan atau keterbatasan fisik. Bahkan bagi atlet cedera ini bisa berarti istirahat yang cukup lama dan mungkin harus meninggalkan sama sekali hobi dan profesinya.

2. RUANG LINGKUP PENELITIAN

1. Rumusan Masalah

Berdasarkan latar belakang masalah diatas, maka dapat dirumuskan masalah yang akan diselesaikan dalam penelitian ini adalah “Bagaimana membuat suatu program sistem pakar untuk mengidentifikasi jenis cedera pada pemain sepak bola dengan menggunakan bahasa pemrograman visual basic 6.0”.

2. Batasan Masalah

Agar penelitian yang dilakukan lebih terarah dengan tujuan penelitian, serta memudahkan dalam mengumpulkan dan pengolahan data, maka ruang lingkup penelitian dibatasi dan diasumsikan sebagai berikut :

1. Sistem pakar ini ditujukan untuk pemain sepakbola amatir agar bisa membantu mereka mengenal dan menganalisa apabila mereka mengalami cedera.
2. Metode inferensi yang digunakan adalah metode pelacakan kedepan (*Forward Chaining*) yaitu sistem akan memberikan pertanyaan mengenai gejala atau ciri yang dialami, yang pada akhirnya dapat menentukan jenis cederanya.
3. Model representasi pengetahuan yang digunakan adalah model kaidah produksi, yaitu model yang dituliskan dalam bentuk *If-Then*.
4. *Output* berupa hasil dari diagnosa, yaitu jenis cedera dan solusi tentang tindakan yang dapat dilakukan.
5. Pakar pada proses penelitian skripsi ini adalah bapak Syarif Hidayatullah, S.Ft.

2.1. Kajian Emperik

Tabel 2.1 Kajian Empirik

No	Nama Penulisan dan Tahun	Judul Skripsi	Bahasa Pemrograman
1	M. Ali Imron, 2011	Aplikasi Sistem Pakar Kerusakan Mesin Fax Brother Dengan Metode <i>Backward Chaining</i> .	<i>Microsoft Visual Basic 6.0</i>
2	Berry Amanda, 2012	Sistem Pakar Diagnosa Penyakit Osteoporosis Dengan Menggunakan Metode <i>Forward Chaining</i> .	<i>Borland Delphi 7.0.</i>
3	Elizabeth Restu W., 2011	Sistem Pakar Diagnosis Kerusakan Mesin Mobil Izusu Panther Grand Touring Menggunakan Metode <i>Backward Chaining</i> .	<i>Microsoft Visual Basic 6.0.</i>

Penelitian ini merupakan penelitian dimana sistem pakar akan di bangun dengan metode pengembangan sistem pakar dan menggunakan bahasa pemrograman *Micosoft Visual Basic 6.0*

2.2. Landasan Teori

2.2.1 Sistem

Menurut Jogiyanto (2005), sistem adalah suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan.

Menurut Kadir (2007), sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Maka, sistem adalah sekelompok komponen dan elemen yang digabungkan menjadi satu untuk mencapai tujuan tertentu.

2.2.2 Pakar

Menurut Turban (2005), pakar adalah orang yang memiliki pengetahuan khusus, pendapat, pengalaman, dan metode, serta kemampuan untuk mengaplikasikan keahliannya tersebut guna menyelesaikan masalah.

Menurut Arhami (2005), pakar adalah orang yang mempunyai keahlian dalam bidang tertentu, yaitu pakar yang mempunyai *knowledge* atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya.

Maka, pakar adalah orang yang memiliki pengetahuan, pengalaman, keahlian, metode khusus,

serta kemampuan untuk menerapkan keahliannya untuk memecahkan masalah.

2.2.3 Sistem Pakar

Menurut Kusrini (2008), sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Dalam sistem pakar penggabungan antara pengetahuan dan fakta serta mekanisme pengambilan keputusan untuk memecahkan suatu masalah yang biasanya memerlukan keahlian seorang pakar.

Maka, sistem pakar adalah suatu program komputer yang dirancang untuk mengambil keputusan seperti keputusan yang diambil oleh seorang atau beberapa orang pakar.

2.2.3.1 Keuntungan Sistem Pakar

Menurut Kusumadewi (2007), keuntungan sistem pakar adalah sebagai berikut.

1. Masyarakat awam non-pakar dapat memanfaatkan keahlian di dalam bidang tertentu tanpa kesadaran langsung seorang pakar.
2. Meningkatkan produktivitas kerja, yaitu bertambahnya *efisiensi* pekerjaan tertentu serta hasil solusi kerja.
3. Penghematan waktu dalam menyelesaikan masalah yang kompleks.
4. Memberikan penyederhanaan solusi untuk kasus-kasus yang kompleks dan berulang-ulang.
5. Pengetahuan dari seorang pakar dapat dikombinasikan tanpa ada batas waktu.
6. Memungkinkan penggabungan berbagai bidang pengetahuan dari berbagai pakar untuk dikombinasikan.

2.2.3.2 Kelemahan Sistem Pakar

Menurut Kusumadewi (2007), kelemahan sistem pakar adalah sebagai berikut.

1. Daya kerja dan produktivitas manusia menjadi berkurang karena semuanya dilakukan secara otomatis oleh sistem.
2. Pengembangan perangkat lunak sistem pakar lebih sulit dibandingkan dengan perangkat lunak konvensional.

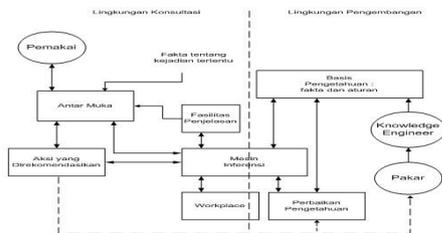
2.2.3.3 Tujuan Pengembangan Sistem Pakar

Tujuan pengembangan sistem pakar adalah untuk mensubstitusikan pengetahuan manusia ke dalam bentuk sistem, sehingga dapat digunakan oleh orang banyak.

2.2.3.4 Struktur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan sistem pakar digunakan untuk memasukkan pengetahuan pakar ke dalam

lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar. Komponen-komponen sistem pakar dalam dua bagian tersebut dapat dilihat pada Gambar 2.1 yaitu *user interface* (antarmuka pengguna), basis pengetahuan, akuisisi pengetahuan, mesin *inference*, *workplace*, fasilitas penjelasan, dan perbaikan pengetahuan.



Gambar 2.1 Arsitektur sistem pakar

Sumber : Arhami (2005), Konsep Dasar Sistem Pakar.

Seorang pakar mempunyai pengetahuan tentang masalah yang khusus. Dalam hal ini disebut *domain knowledge*. Penggunaan kata “*domain*” untuk memberikan penekanan pengetahuan pada *problem* yang spesifik. Pakar menyimpan *domain knowledge* pada *Long Term Memory (LTM)* atau ingatan jangka panjangnya.

Ketika pakar akan memberikan nasihat atau solusi kepada seseorang, pakar terlebih dahulu menentukan fakta-fakta dan menyimpannya ke dalam *Short Term Memory (STM)* atau ingatan jangka pendek. Kemudian pakar memberikan solusi tentang masalah tersebut dengan mengkombinasikan fakta-fakta pada STM dengan pengetahuan LTM. Dengan menggunakan proses ini pakar mendapatkan informasi baru dan sampai pada kesimpulan masalah.

2.2.3.5 Komponen Sistem Pakar

Sebuah program yang difungsikan untuk menirukan seorang pakar manusia harus bisa melakukan hal-hal yang dapat dikerjakan seorang pakar. Untuk membangun sistem seperti itu maka komponen-komponen dasar yang harus dimilikinya paling sedikit adalah sebagai berikut:

1. Antar muka pemakai (*User Interface*).
2. Basis pengetahuan (*Knowledge Base*).
3. Mesin inferensi.

Sedangkan untuk menjadikan sistem pakar menjadi lebih menyerupai seorang pakar yang berinteraksi dengan pemakai, maka dapat dilengkapi dengan fasilitas berikut:

1. Fasilitas penjelasan (*Explanation*).
2. Fasilitas Akuisisi pengetahuan (*Knowledge acquisition facility*).
3. Fasilitas pelatihan (*self-training*).

2.2.3.6 Metode Inferensi

Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Menurut Turban (2005), metode inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan

dalam *workplace*, dan untuk memformulasikan kesimpulan.

Kebanyakan sistem pakar berbasis aturan menggunakan strategi inferensi yang dinamakan *modus ponens*. Berdasarkan strategi ini, jika terdapat aturan “IF A THEN B”, dan jika diketahui bahwa A benar, maka dapat disimpulkan bahwa B juga benar. Strategi inferensi *modus ponens* dinyatakan dalam bentuk:

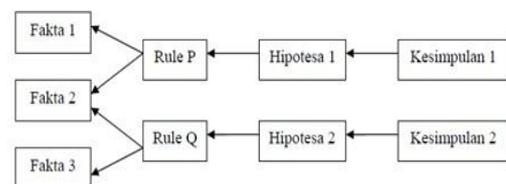
$$[A \text{ And } (A \rightarrow B)] \rightarrow B,$$

dengan A dan $A \rightarrow B$ adalah proposisi-proposisi dalam basis pengetahuan.

Terdapat dua pendekatan untuk mengontrol inferensi dalam sistem pakar berbasis aturan, yaitu pelacakan ke belakang (*Backward chaining*) dan pelacakan ke depan (*forward chaining*).

1. Pelacakan ke belakang (*Backward Chaining*)

Menurut Kusumadewi (2007), pelacakan ke belakang adalah pendekatan yang dimotori oleh tujuan (*goal driven*). Dalam pendekatan ini pelacakan dimulai dari tujuan, selanjutnya dicari aturan yang memiliki tujuan tersebut untuk kesimpulannya. Selanjutnya proses pelacakan menggunakan premis untuk aturan tersebut sebagai tujuan baru dan mencari aturan lain dengan tujuan baru sebagai kesimpulannya. Proses berlanjut sampai semua kemungkinan ditemukan. Gambar 2.2 menunjukkan proses runtu mundur (*backward chaining*).

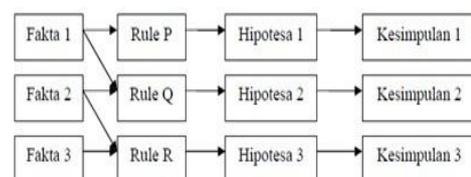


Gambar 2.2 Proses runtu mundur (*backward chaining*)

Sumber: Kusumadewi (2004), *Artificial Intelligence* (Teknik dan Aplikasinya).

2. Pelacakan ke depan (*forward chaining*)

Menurut Kusumadewi (2004), pelacakan kedepan adalah pendekatan yang dimotori data (*data driven*). Dalam pendekatan ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Pelacakan ke depan, mencari fakta yang sesuai dengan bagian IF dari aturan IF-THEN. Gambar 2.3 menunjukkan proses runtu maju (*forward chaining*).



Gambar 2.3 Proses runtu maju (*forward chaining*)

Sumber: Kusumadewi (2004), *Artificial Intelligence* (Teknik dan Aplikasinya).

2.2.3.7 Tahapan Pengembangan Sistem Pakar

Menurut Riley (2005), didalam pengembangan sistem pakar terdapat 6 tahap yang dilakukan, penjelasan berikut merupakan penjelesasan secara garis besar tentang tahap-tahap pengembangan tersebut.

1. Penilaian (*Assessment*)

Proses untuk menentukan kalayakan dan justifikasi atas permasalahan yang akan diambil. Setelah itu masalah diperiksa lebih lanjut untuk menentukan tujuan keseluruhan dari proyek. Upaya ini dilakukan untuk menentukan fitur-fitur penting dan ruang lingkup dari proyek, dan juga untuk menetapkan sumber daya yang diperlukan, termasuk diantaranya para pakar dan juga berbagai laporan harus diidentifikasi. Setelah tahap inisialisasi dilakukan persyaratan-persyaratan proyek ditetapkan.

2. Akuisisi Pengetahuan

Proses untuk mendapatkan pengetahuan tentang permasalahan yang dibahas dan akan digunakan sebagai panduan dalam upaya pengembangan. Pengetahuan ini digunakan untuk memberikan informasi tentang permasalahan yang menjadi bahan dalam mendesain pakar. Tahap ini meliputi studi dengan diadakannya pertemuan dengan pakar untuk membahas aspek dari permasalahan.

3. Perancangan

Pengetahuan yang diperoleh selama tahap akuisisi pengetahuan digunakan sebagai pendekatan dalam merepresentasikan pengetahuan pakar dan strategi pemecahan masalah ke dalam sistem pakar. Selama tahap perancangan, keseluruhan struktur dan organisasi dari sistem pengetahuan harus ditetapkan.

4. Pengujian

Tahap dimana dilakukan pengujian terhadap sistem pakar yang telah dibangun.

5. Dokumentasi

Dokumentasi diperlukan untuk mengkompilasi seluruh informasi proyek kedalam bentuk dokumen yang dapat memenuhi persyaratan pengguna dan pengembang dari sistem pakar serta untuk mengakomodasi kebutuhan pengguna yang memenuhi persyaratan yang ditemukan pada sebagian besar proyek perangkat lunak. Dokumentasi tersebut menjelaskan tentang bagaimana mengoperasikan sistem dan menyediakan tutorial dalam mengoperasikan fitur utama dari sistem.

6. Pemeliharaan Sistem

Setelah sistem digunakan dalam lingkungan kerja, maka selanjutnya diperlukan pemeliharaan secara berkala. Pengetahuan itu sifatnya tidak statis melainkan terus tumbuh dan berkembang. Pengetahuan dari sistem perlu diperbaharui dan disempurnakan untuk memenuhi kebutuhan saat ini.

2.2.4 Sepak Bola

Menurut Muhajir (2007), sepak bola adalah suatu permainan yang dilakukan dengan jalan menyepak, yang mempunyai tujuan untuk memasukkan bola ke gawang lawan dengan mempertahankan gawang tersebut agar tidak kemasukan bola.

Menurut Sukintaka (2005), permainan sepak bola adalah permainan bola besar yang dimanipulasi dengan

kaki dan seluruh anggota badan kecuali tangan yang dimainkan oleh dua buah regu yang masing-masing regu terdiri atas sebelas orang pemain.

Maka sepak bola adalah salah satu cabang plahraga beregu yang dimainkan 2 tim, setiap tim diisi oleh 11 orang, dengan tujuan memasukkan bola ke gawang lawan.

2.2.5 Cedera Dalam Olahraga

Menurut Syamsuri (2005), cedera dalam dunia olahraga yaitu rusaknya jaringan (lunak atau keras) baik otot, tulang, atau persendian yang disebabkan oleh kesalahan teknis, benturan, atau aktivitas yang melebihi batas beban latihan (*overtraining*) yang dapat menimbulkan rasa sakit atau nyeri dan atau akibat dari kelebihan latihan dalam memberikan pembebanan yang terlalu berat (*overload*) sehingga otot, tulang, atau persendian tidak lagi dalam keadaan atau posisi anatomis (dislokasi).

Cedera dalam dunia olahraga dapat dikategorikan menjadi tiga tingkatan, yaitu: cedera ringan/cedera tingkat pertama, cedera sedang/cedera tingkat kedua, dan cedera berat/cedera tingkat ketiga.

2.2.5.1 Cedera Ringan/Cedera Tingkat Pertama

Cedera ringan/cedera tingkat pertama adalah cedera yang diikuti kerusakan yang tidak berarti pada jaringan tubuh, namun dapat mengganggu penampilan atlet di antaranya. Pada tingkat cedera ini penderita tidak mengalami keluhan yang serius, namun dapat mengganggu penampilan atlet. Berbagai macam cedera tingkat pertama adalah sebagai berikut.

1. Memar



Gambar 2.4 Memar

Sumber : Syamsuri (2005), Cedera Dalam Olahraga

Pada gambar 2.4 memar adalah cedera yang disebabkan oleh benturan atau pukulan pada kulit. Jaringan dibawah permukaan kulit rusak dan pembuluh darah kecil pecah, sehingga darah dan cairan seluler merembes ke jaringan sekitarnya.

2. Kram Otot



Gambar 2.5 Kram Otot

Sumber : Syamsuri (2005), Cedera Dalam Olahraga

Pada gambar 2.5 kram otot adalah kontraksi yang terus menerus yang dialami oleh otot atau sekelompok otot dan mengakibatkan rasa nyeri. Penyebab kram adalah otot yang terlalu lelah, kurangnya pemanasan

serta peregangan, adanya gangguan sirkulasi darah yang menuju ke otot sehingga menimbulkan kejang.

3. Lecet



Gambar 2.6 Lecet

Sumber : Hadiano (2007), Pencegahan dan Penatalaksanaan Cedera Olahraga

Pada gambar 2.6 lecet adalah luka dangkal terbuka yang menimbulkan perdarahan dan kerusakan ujung-ujung saraf di kulit, yang menimbulkan rasa perih.

2.2.5.2 Cedera Sedang/Cedera Tingkat Kedua

Cedera sedang/cedera tingkat kedua ini ditandai dengan kerusakan jaringan yang nyata, nyeri, bengkak, memar, berwarna kemerah-merahan (suhu agak panas), dengan gangguan fungsi yang nyata dan berpengaruh pada penampilan atlet yang bersangkutan baik pada saat berlatih maupun bertanding. Berbagai macam cedera tingkat kedua adalah sebagai berikut.

1. Sprain



Gambar 2.7 Sprain

Sumber : Peterson (2006), *Sport Injuries*.

Pada gambar 2.7 sprain adalah bentuk cedera berupa penguluran atau kerobekan pada ligamen (jaringan yang menghubungkan tulang dengan tulang) atau kapsul sendi, yang memberikan stabilitas sendi. Kerusakan yang parah pada ligamen atau kapsul sendi dapat menyebabkan ketidakstabilan pada sendi. Sprain terjadi ketika sendi dipaksa melebihi lingkup gerak sendi yang normal, seperti melingkar atau memutar pergelangan kaki.

2. Strain



Gambar 2.8 Strain

Sumber : Syamsuri (2005), Cedera Dalam Olahraga

Pada gambar 2.8 strain adalah bentuk cedera berupa penguluran atau kerobekan pada struktur muskulo-tendinous (otot dan tendon). Jenis cedera ini terjadi akibat otot tertarik pada arah yang salah, kontraksi otot

yang berlebihan atau ketika terjadi kontraksi, otot belum siap. Strains sering terjadi pada bagian *groin muscles* (otot pada kunci paha), *hamstrings* (otot paha bagian bawah), dan otot *quadriceps*. Cedera tertarik otot betis juga kerap terjadi pada para pemain bola.

3. Cedera Osgood Schlatters



Gambar 2.9 Osgood Schlatters

Sumber : Brukner (2012), *Clinical Sport Medicine*.

Pada gambar 2.9 *osgood Schlatters* adalah suatu penyakit peradangan pada tulang, tulang rawan, dan / atau tendon di bagian atas tulang kering (*tibia*), di mana tendon dari tempurung lutut (*patella*) menempel. Paling sering hanya satu lutut terpengaruh.

2.2.5.3 Cedera Berat/Cedera Tingkat Ketiga

Cedera berat/cedera tingkat ketiga ini ditandai dengan kerusakan jaringan atau terjadi robekan lengkap atau hampir lengkap pada otot, ligamentum, dan fraktur pada tulang yang memerlukan waktu istirahat lebih lama atau total, dan membutuhkan terapi, pengobatan secara intensif, dan bahkan dimungkinkan untuk dioperasi.

1. Patah atau retak tulang (Fracture)

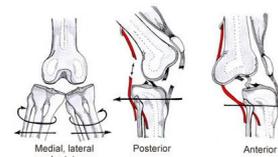


Gambar 2.10 Patah atau retak tulang (Fracture)

Sumber : Peterson (2006), *Sport Injuries*.

Pada gambar 2.10 cedera seperti ini dialami apabila pemain yang bersangkutan mengalami benturan dengan pemain lain atau sesuatu yang keras. Cedera *fractures* tidak hanya terjadi pada bagian kaki macam tulang paha, tulang kering, tulang selangkangan, atau tulang telapak kaki, tapi juga kerap terjadi pada lengan, bahu, hingga pergelangan tangan. Untuk menghindari cedera macam ini, penggunaan pelindung sangat dianjurkan untuk meminimalisir patah atau retak tulang.

2. Dislocation

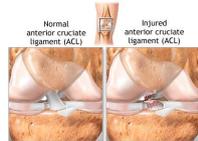


Gambar 2.11 Dislocation

Sumber : Peterson (2006), *Sport Injuries*.

Tempurung lutut penting sekali dalam setiap aktifitas olahraga yang banyak membutuhkan gerakan pada kaki bawah. *Patella* merupakan lapisan piringan sendi tulang yang terletak pada ujung *femur*. *Femur* ini memiliki celah pada ujungnya, yang merupakan tempat *patella* pada saat kaki melakukan gerakan menekuk. Jika *patella* keluar dari celahnya dan berpindah kesalah satu sisi akan menimbulkan pergerakan letak. Pergeseran yang tidak pada tempatnya ini merupakan subluksi, dimana tempurung lutut tidak menempati posisi sebagaimana mestinya tetapi menyelip sedikit ke salah satu sisi ini akan menimbulkan rasa sakit dan dapat diperkirakan telah terjadi pergeseran tempat *patella*.

3. Cedera ACL (Anterior Cruciate Ligament)



Gambar 2.12 Anterior Cruciate Ligament
Sumber : Peterson (2006), *Sport Injuries*.

Pada cedera ini adanya peregangan yang berlebihan atau robeknya ligamen anterior di lutut.

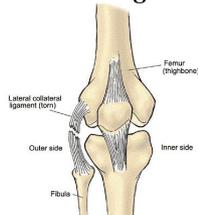
4. Cedera Tendinitis Patellar (Jumper Knee).



Gambar 2.13 Tendinitis Patellar
Sumber : Brukner (2012), *Clinical Sport Medicine*.

Pada gambar 2.13 *patellar tendinitis* adalah cedera akibat penggunaan berlebihan karena tekanan yang berulang pada tendon *patella*. Tekanan ini mengakibatkan putusnya tendon yang jika terlalu banyak dapat menyebabkan radang pada tendon. *Patellar tendinitis* adalah cedera akibat penggunaan berlebihan karena tekanan yang berulang pada tendon *patella*. Tekanan ini mengakibatkan putusnya tendon yang jika terlalu banyak dapat menyebabkan radang pada tendon.

5. Cedera Lateral Collateral Ligament



Gambar 2.14 Lateral Collateral Ligament
Sumber : Peterson (2006), *Sport Injuries*.

Pada gambar 2.14 cedera ini adanya robekan ligamen pada tulang paha ke *fibula*, atau tulang kecil kaki bagian bawah pada sisi samping atau luar lutut.

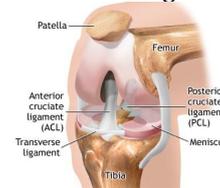
6. Cedera Patellofemoral Syndrome



Gambar 2.15 Patellofemoral Syndrome
Sumber : Brukner (2012), *Clinical Sport Medicine*.

Pada gambar 2.15 cedera ini terjadi ketika tulang rawan *patella* mendapat beban berlebih karena terlalu banyak digunakan (seringkali disebabkan oleh aktifitas yang tinggi) atau sebagai hasil dari keseimbangan pergerakan yang buruk.

7. Cedera Posterior Cruciate Ligament



Gambar 2.16 Posterior Cruciate Ligament
Sumber : Sumber : Peterson (2006), *Sport Injuries*.

Pada cedera ini adanya peregangan yang berlebihan atau robeknya ligamen posterior di lutut.

2.3 Alat Bantu Pengembangan Sistem

2.3.1 Flow Chart

Menurut Jogiyanto (2005), *flowchart* adalah alat bantu yang banyak digunakan untuk menggambarkan aliran kerja suatu sistem. Bagan alir program dibuat menggunakan simbol-simbol sebagai berikut :

Tabel 2.2 Simbol-Simbol *Flowchart*

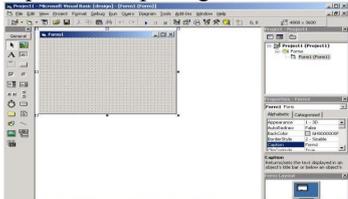
SIMBOL	NAMA	KETERANGAN
	Terminator	Digunakan untuk menunjukkan awal dan akhir dari suatu proses.
	Input/Output	Digunakan untuk mewakili data input/output.
	Garis alir	Digunakan untuk menunjukkan alur dari proses.
	Proses	Digunakan untuk mewakili suatu proses.
	Keputusan	Digunakan untuk suatu penyeleksian kondisi didalam program.
	Persiapan	Digunakan untuk memberi nilai awal suatu besaran.
	Penghubung	Digunakan untuk menunjukkan sambungan dari bagan alir yang terputus.

Sumber : Jogiyanto, 2005, Analisis dan Desain Sistem Informasi

2.4 Visual Basic

Madcoms (2008), *Visual Basic* selain disebut sebagai bahasa pemrograman (*Language Program*), juga sering disebut sebagai sarana (*Tool*) untuk menghasilkan program-program aplikasi berbasis windows.

Setelah *Visual Basic* dijalankan, akan muncul sebuah layar. Layar ini adalah lingkungan pengembangan aplikasi *Visual Basic* yang nantinya akan digunakan untuk membuat program-program aplikasi dengan *Visual Basic*. Bentuk tampilan awal jendela visual basic dapat dilihat pada gambar 2.17.



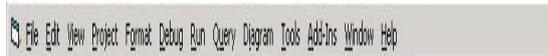
Gambar 2.17 Tampilan Awal Jendela *Visual Basic*

Sumber : Madcoms,(2008), *Microsoft Visual Basic 6.0 & Crystal Report 2008*.

Visual Basic IDE (Interface Development Environment) ialah tampilan antarmuka program dengan pengguna yang sudah bersifat GUI (*Grafical User Interface*), menyajikan banyak kemudahan bagi para programmer untuk membuat aplikasi. Tampilan IDE *Visual Basic* sebagai berikut :

2.4.1 Menu Bar

Menu Bar berfungsi memberikan kemudahan kepada pengguna dalam memilih aksi-aksi yang umum seperti mengedit, mengkopi atau menjalankan program. Beberapa fungsi yang ada di menu bar juga tersedia di *toolbar*. Bentuk tampilan menu bar dapat dilihat pada gambar 2.18.

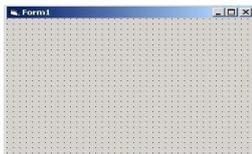


Gambar 2.18 Tampilan Menu Bar

Sumber : Madcoms,(2008), *Microsoft Visual Basic 6.0 & Crystal Report 2008*.

2.4.2 Jendela Form

Form di *Visual Basic* adalah media tempat kita membuat aplikasi atau antarmuka yang bersifat GUI (*Grafical User Interface*). Kita dapat menempelkan berbagai macam objek atau kontrol di atas *form* tersebut. Bentuk tampilan jendela *form* dapat dilihat pada gambar 2.19.



Gambar 2.19 Tampilan Jendela *Form*

Sumber : Madcoms,(2008), *Microsoft Visual Basic 6.0 & Crystal Report 2008*.

2.4.3 Toolbar

Toolbar berfungsi menyediakan fasilitas yang umum digunakan oleh para *programmer* ketika sedang mendesain aplikasi, seperti membuka atau menyimpan

file, menjalankan program dan lain-lain. Bentuk tampilan *toolbar* dapat dilihat pada gambar 2.20.

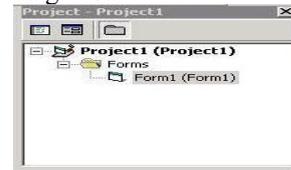


Gambar 2.20 Tampilan *Toolbar*

Sumber : Madcoms,(2008), *Microsoft Visual Basic 6.0 & Crystal Report 2008*.

2.4.4 Jendela Project

Dengan adanya jendela *project*, kita dapat melihat form atau objek apa saja yang ada didalam satu proyek, dan kita dapat menghapusnya dari jendela *project* tersebut. Apabila anda membuat suatu program aplikasi baru, maka secara otomatis *project* tersebut akan diisi dengan objek *Form1*. Bentuk tampilan jendela *project* dapat dilihat pada gambar 2.21.



Gambar 2.21 Tampilan Jendela *Project*

Sumber : Madcoms,(2008), *Microsoft Visual Basic 6.0 & Crystal Report 2008*.

2.4.5 Toolbox

Toolbox merupakan kotak perangkat yang berisi kumpulan tombol objek atau kontrol untuk mengatur desain dari aplikasi yang akan dibuat. Bentuk tampilan *toolbox* dapat dilihat pada gambar 2.22.



Gambar 2.22 Tampilan *Toolbox*

Sumber : Madcoms,(2008), *Microsoft Visual Basic 6.0 & Crystal Report 2008*.

2.4.6 Jendela Properties

Jendela *Properties* adalah sebuah jendela yang mengandung semua informasi mengenai objek yang terdapat pada aplikasi *Visual Basic*. Properti adalah sifat dari sebuah objek, misalnya seperti nama, warna, ukuran, posisi dan sebagainya. Bentuk tampilan jendela *properties* dapat dilihat pada gambar 2.23.



Gambar 2.23 Tampilan Jendela *Properties*

Sumber : Madcoms,(2008), *Microsoft Visual Basic 6.0 & Crystal Report 2008*.

2.4.7 Code Editor

Code Editor ini berisi kode-kode program yang merupakan instruksi untuk aplikasi *Visual Basic* agar dijalankan seperti menutup aplikasi, membatalkan perintah, mengaktifkan salah satu objek dan sebagainya.

Bentuk tampilan *code editor* dapat dilihat pada gambar 2.24.

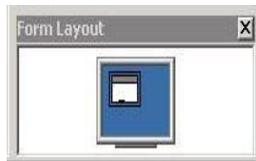


Gambar 2.24 Tampilan *Code Editor*

Sumber : Madcoms,(2008), *Microsoft Visual Basic 6.0 & Crystal Report 2008*.

2.4.8 Jendela *Form Layout*

Form layout window berfungsi untuk menetapkan posisi tampilan program ketika dijalankan. Posisi pada *form layout window* inilah yang merupakan petunjuk dimana aplikasi akan ditampilkan pada layar monitor saat dijalankan. Bentuk tampilan jendela *form layout* dapat dilihat pada gambar 2.25.



Gambar 2.25 Tampilan Jendela *Form Layout*

Sumber : Madcoms,(2008), *Microsoft Visual Basic 6.0 & Crystal Report 2008*.

2.5 Database Microsoft Access 2007

Menurut Madcoms (2008), *Microsoft Office Access* merupakan salah satu *software* pengolah *database*. Dengan *Microsoft Access* anda dapat mengolah berbagai jenis data serta membuat hasil akhir berupa laporan dengan tampilan yang lebih menarik. Dengan *Microsoft Access* anda tidak harus mempelajari program lain sebagai referensi karena *Access* dirancang untuk pendatang baru atau pemula sekalipun. *Database* ini dibuat secara apik dan menyediakan banyak template sesuai bentuk *database* yang kita inginkan.

2.6 Metode Pengujian Sistem

Menurut Pressman (2007), sebelum mengaplikasikan metode untuk mendesain *test case* yang efektif, perancang perangkat lunak harus memahami prinsip dasar yang menuntun pengujian perangkat lunak. Metode-metode pengujian sistem, antara lain :

1. Pengujian *Black Box*

Menurut Pressman (2007), Pengujian *Blackbox* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *Blackbox* memungkinkan perancang perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

2. Pengujian *White Box*

Menurut Pressman (2007), Pengujian *Whitebox* adalah metode desain *test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test case*.

3. Pengujian *Beta*

Pengujian *Beta* adalah pengujian yang dilakukan pada satu atau lebih pelanggan oleh pemakai akhir

perangkat lunak. Tidak seperti pengujian *alpha*, pengembang biasanya tidak ada sehingga pengujian beta merupakan sebuah aplikasi “*live*” dari perangkat lunak didalam suatu lingkungan yang tidak dapat dikontrol oleh pengembang. Pelanggan merekam semua masalah (*real* atau imajiner) yang mereka temui selama pengujian beta melaporkannya kepada pengembang dalam interval yang regular. Sebagai hasil dari pelaporan masalah selama pengujian beta ini, pengembang perangkat lunak melakukan modifikasi dan kemudian mempersiapkan pelepasan produk perangkat lunak ke seluruh pelanggan. Rumus pengujian *beta* adalah sebagai berikut :

$$Y = P/Q * 100\%$$

Keterangan :

P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

Y = Nilai persentase

3. METODE

3.1 Tahapan Pengembangan Sistem Pakar

3.3.1 Penilaian

Penilaian merupakan proses untuk menentukan kelayakan dan justifikasi atas permasalahan yang akan diambil. Setelah itu masalah diperiksa lebih lanjut untuk menentukan tujuan keseluruhan dari proyek.

Pada tahapan ini dilakukan proses analisis yang diambil adalah sampel dari beberapa data dan meliputi beberapa proses.

1. Analisis Data

Analisis Data yaitu analisis mengenai data apa saja yang akan diproses, baik sebagai masukan maupun keluaran. Data yang dipergunakan berasal dari hasil penelitian dari pakarnya maupun dari referensi tentang gejala-gejala cedera pada pemain sepakbola.

2. Analisis Kebutuhan

Ada dua analisis yang dibutuhkan dalam penelitian ini yaitu:

1) Analisis fungsional

Tahap ini menjelaskan bahwa aplikasi yang akan dibuat nantinya dapat digunakan untuk mengidentifikasi cedera berdasarkan gejala-gejala dan tanda-tanda yang teramat.

2) Analisis Non fungsional

Pada pengoperasian aplikasi pada sistem ini teknologi yang dibutuhkan menjadi 2 (dua) yaitu :

1. Kebutuhan perangkat lunak

Perangkat Lunak yang digunakan dalam pembuatan sistem ini adalah perangkat lunak sistem operasi *Microsoft Windows 7*, *database* yang digunakan *Microsoft Office Acces 2007*, bahasa pemrograman yang digunakan *Visual Basic 6.0*.

2. Kebutuhan Perangkat Keras

Perangkat komputer yang dibutuhkan dalam pembuatan sistem ini dengan spesifikasi : *processor intel dual core* atau yang setara, RAM dengan kapasitas 1 GB, *Hardisk* 320 GB.

3. Pemakai (*user*)

User atau pengguna aplikasi yang dibuat adalah yang memiliki pengetahuan atau kemampuan untuk melihat gejala-gejala dari cedera.

3.3.2 Akuisisi Pengetahuan

Proses untuk mendapatkan pengetahuan tentang permasalahan yang dibahas dan akan digunakan sebagai panduan dalam upaya pengembangan. Pengetahuan ini digunakan untuk memberikan informasi tentang permasalahan yang menjadi bahan dalam mendesain pakar.

1. Cedera Ringan/Cedera Tingkat Pertama
2. Cedera Sedang/Cedera Tingkat Kedua
3. Cedera Berat/Cedera Tingkat Ketiga

3.3.3 Perancangan

Dari semua data yang telah terkumpul dilakukan proses perancangan atau desain sistem, sehingga dapat mempermudah dalam penggambaran sistem dan memperjelas gambaran sistem yang ada. Teknik desain sistem yang dipergunakan dalam rancangan ini terbagi dua yaitu :

1. Desain antar muka pengguna (*user*) yang berguna agar *user* dapat melakukan penelusuran cedera pada pemain sepakbola.
2. Desain antar muka pakar yang berguna agar pakar dapat menambah, mengubah, menghapus data pengetahuan
3. *Flowchart* yang digunakan untuk menjelaskan secara rinci langkah-langkah dari proses program.

3.3.4 Pengujian

Setelah program selesai dibuat, selanjutnya hal yang perlu dilakukan adalah pengujian. Adapun metode yang digunakan untuk melakukan pengujian dalam perancangan sistem ini adalah sebagai berikut.

1. Pengujian *Beta*

Pada pengujian *beta* ini tahap pengujian yang dilakukan adalah terhadap *user*, dimana kesimpulan dari pengujian ini didapat dengan cara kuisioner yang diberikan kepada lima responden yang berbeda. Hal yang diujikan dalam *beta* ini adalah desain *interface* aplikasi, *input* bagian penyakit yang akan di diagnosa, *input* gejala yang akan di diagnosa dan *output* hasil diagnosa. Contoh pengujian *beta* dapat dilihat pada tabel 3.1 berikut.

Tabel 3.1 pengujian *Beta*

No	Kriteria	Jumlah (orang)					Persentase (%)					
		SS	S	CS	KS	TS	SS	S	CS	KS	TS	
1	Kemudahan											
2	Tombol Navigasi											
3	Tampilan											
4	Content											
5	Kompetibel											
6	Ketepatan informasi											
7	Fungsional											
8	Interaktif											
9	Kelayakan											
	Total											

2. *Blackbox*

Tujuan dari metode *Blackbox* adalah untuk mendapatkan kesalahan *output* yang dihasilkan program sebanyak-banyaknya.

Metode ini dilakukan dengan cara menjalankan atau mengeksekusi program yang dihasilkan. Kemudian diamati apakah hasil dari program tersebut sesuai dengan hasil yang diinginkan. Jika masih terdapat kesalahan atau terdapat hasil yang tidak sesuai dengan yang diinginkan, maka kesalahan ataupun ketidaksesuaian tersebut dicatat untuk selanjutnya dicek satu per satu dan diperbaiki.

Untuk mempermudah dalam proses pengujian maka perlu dibuatnya suatu tabel pengujian sebagai tolak ukur atau acuan dalam pengembangan sistem tersebut yang dapat dilihat pada tabel 3.2 dibawah ini.

Table 3.2 Pengujian *Blackbox*

APLIKASI	DATA INPUT	OUTPUT
Menu Diagnosa	Memilih gejala cedera yang telah disediakan	Tombol "lanjut" diklik maka akan masuk kehalaman selanjutnya berupa pertanyaan mengenai gejala cedera. Hasil diagnosa berupa nama cedera, penyebab, solusi dan gambar.
Menu Login	Input login dan password	Tombol "masuk" diklik maka akan masuk ke halaman basis pengetahuan
Menu Data Gejala Cedera	Baru, simpan, edit, hapus	Menampilkan No gejala dan nama gejala, penyakit
Menu Data Jenis Cedera dan Solusi	Baru, simpan, edit, hapus	Menampilkan No Jenis ,nama cedera, penyebab, solusi, dan gambar cedera.

3. *Whitebox*

Pengujian *Whitebox* berfokus pada struktur kontrol program. *Test case* dilakukan untuk memastikan bahwa semua statement pada program telah dieksekusi paling tidak satu kali selama pengujian dan bahwa semua kondisi logis telah diuji, untuk melakukan serangkaian pengujian yang independent secara linear yang akan memastikan cakupan.

3.3.5 Dokumentasi

Tahap dokumentasi diperlukan untuk mengkompilasi seluruh informasi proyek kedalam bentuk dokumen yang dapat memenuhi persyaratan pengguna dan pengembangan dari sistem pakar.

3.3.6 Pemeliharaan Sistem

Pemeliharaan sistem sangat mudah adapun pemeliharaan sistem memiliki tujuan sebagai berikut :

1. Membetulkan kesalahan yang dibuat selama proses sistem yang didesain dan diimplementasi.
2. Untuk memelihara bagian program yang benar dan untuk menghindari memperbaiki bagian tersebut.
3. Untuk menjamin keseluruhan proses sistem yang berhubungan.

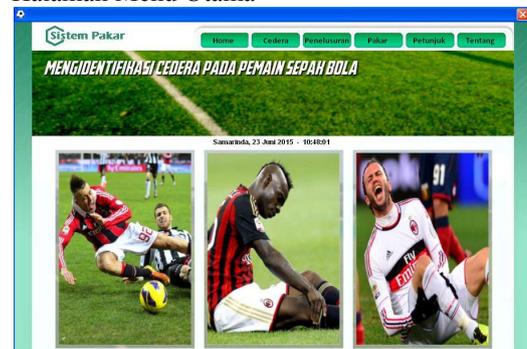
4. RANCANGAN SISTEM/APLIKASI

Rancangan menggunakan flowchart dan tabel

5. IMPLEMENTASI

Hasil implementasi berdasarkan analisis dan perancangan adalah sebagai berikut :

1. Halaman Menu Utama



Gambar 5.1 Tampilan Menu Utama

Pada gambar 5.1 merupakan *design* tampilan menu utama, disini pengguna dapat memilih tombol menu yang ada di menu utama.

6. KESIMPULAN

1. Sistem pakar mengidentifikasi cedera pada pemain sepakbola ini dapat membantu pengguna dalam hal mengetahui secara tepat tentang nama cedera yang dialami berdasarkan pilihan gejala-gejala cederanya sehingga dapat dilakukan pengobatan secara cepat dan tepat.
2. Sistem pakar mengidentifikasi cedera pada pemain sepakbola ini mempunyai proses penelusuran yang sangat mudah untuk dipahami dan dijalankan serta mempunyai penjelasan secara detail sehingga hasil penelusurannya bisa lebih terperinci dan tepat sasaran.
3. Sistem pakar ini dibuat dengan tampilan semenarik mungkin dan mudah dipahami dengan tujuan agar dapat mudah dipelajari dan dipergunakan dengan mudah oleh para pengguna aplikasi ini.

7. SARAN

Berdasarkan dari kesimpulan diatas maka penulis memberikan saran-saran sebagai berikut :

1. Aturan yang ada di sistem pakar ini masih terbatas pada beberapa jenis cedera yang umum ditemui. Untuk pengembangannya dapat ditambahkan lagi untuk berbagai jenis cedera lainnya seperti *dislocation* dan *fracture*, sehingga bisa mencakup segala jenis cedera pada pemain sepakbola maupun olahraga lainnya.
2. Sistem pakar ini penelusurannya tidak memiliki nilai kepastian, sehingga untuk pengembangannya bisa ditambahkan metode *certainty factor* untuk mengatasi ketidakpastian dalam pengambilan keputusan, agar penelusurannya bisa lebih terperinci dan akurat.
3. Sistem pakar ini masih menampilkan solusi berupa teks dan gambar sehingga dapat dikembangkan lagi untuk solusi berupa multimedia dengan tampilan suara dan juga video baik tentang pengobatannya dan juga tentang cara pencegahannya.

8. DAFTAR PUSTAKA

Buku:

- Amanda, Berry. 2012. *Sistem Pakar Diagnosa Penyakit Osteoporosis Menggunakan Metode Forward Chaining*. Samarinda: STMIK WICIDA
- Arhami, Muhammad. 2005. *Konsep Dasar Sistem Pakar*. Yogyakarta: Andi Offset.
- Brukner, Peter. 2012. *Clinical Sport Medicine*. Surabaya: EGC.
- Imron, M. Ali. 2011. *Sistem Pakar Kerusakan Mesin Fax Brother dengan Metode Backward Chaining*. Samarinda: STMIK WICIDA
- Jogianto, HM. 2005. *Analisis Dan Desain Sistem Informasi*. Yogyakarta: Andi Offset.
- Restu, Elizabeth W. 2011. *Sistem Pakar Diagnosis Kerusakan Mesin Mobil Izusu Panther Grand*

Touring Menggunakan Metode Backward Chaining. Samarinda: STMIK WICIDA

- Kadir, Abdul. 2007. *Pengenalan Sistem Informasi*. Yogyakarta: Andi Offset.
- Kusumadewi, Sri. 2007. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Kusrini. 2008. *Aplikasi Sistem Pakar*. Yogyakarta: Andi Offset.
- Madcoms. 2008. *Microsoft Access 2007 untuk pemula*. Yogyakarta: Andi Offset.
- Madcoms. 2008. *Microsoft Visual Basic 6.0 & Crystal Report 2008*. Yogyakarta: Andi Offset.
- Muhajir. 2007. *Pendidikan Jasmani Olahraga dan Kesehatan SMA Kelas 1*. Jakarta: Erlangga.
- Peterson, L, Renstrom, P. 2006. *Sport Injuries*. Surabaya: EGC.
- Pressman, Roger S. 2007. *Rekayasa Perangkat Lunak: pendekatan praktis (buku I)*. Yogyakarta : Andi Offset
- Riley, Giarratano. 2005. *Expert System Principles and Programming*. USA: Thomson Course Technology
- Sukintaka. 2005. *Teori Pendidikan Jasmani (Filosofi, Pembelajaran dan Masa depan)*. Bandung: Nuansa.
- Syamsuri. 2005. *Cedera Dalam Olahraga*. Jakarta: Rajagrafindo
- Turban. 2005. *Decision Support System And Intelligent System*. Yogyakarta: Andi Offset.
- Wibowo, Hardianto. 2007. *Pencegahan dan Penatalaksanaan Cedera Olah Raga Edisi II*. Surabaya: EGC.