

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Sistem Pakar**

Menurut Suhendro Yusuf (2011) Sistem pakar (*expert system*) merupakan sebagian kecil dari dunia kecerdasan buatan. Sistem ini dirancang dengan sedemikian rupa agar dapat menirukan keahlian yang dimiliki oleh seorang pakar pada bidang tertentu. Sistem pakar di rancang untuk dapat menjawab pertanyaan-pertanyaan dan memberikan solusi di berbagai bidang, seperti kesehatan bisnis, pendidikan, ekonomi bahkan perbankan. Sistem pakar dapat menyimpan suatu pengetahuan secara keseluruhan.

##### **2.1.1 Konsep Dasar Sistem Pakar**

Menurut Efrain Turban dalam (BH Hayadi, 2018), konsep dasar sistem pakar mengandung keahlian, ahli, pengalihan keahlian, inferensi, aturan dan kemampuan untuk menjelaskan. Keahlian adalah suatu kelebihan penguasaan pengetahuan di bidang tertentu yang diperoleh dari pelatihan, membaca dan pengalaman. Seorang ahli atau pakar adalah seseorang yang mampu menjelaskan suatu tanggapan.

Mempelajari hal-hal yang baru seputar topik permasalahan (domain), menyusun kembali pengetahuan jika dipandang perlu, memecah aturan-aturan jika dibutuhkan, dan menentukan relevan tidaknya keahlian mereka. Pengalihan keahlian yang dimaksud adalah pengalihan keahlian dari para ahli ke komputer untuk kemudian dialihkan lagi ke orang lain yang bukan ahli, merupakan tujuan utama dari sistem pakar. Proses ini membutuhkan empat aktivitas yaitu : tambahan pengetahuan (dari ahli atau sumber-sumber lainnya) → representasi pengetahuan ke komputer → inferensi pengetahuan → pengalihan pengetahuan ke *user*. Pengetahuan yang disimpan di komputer disebut dengan nama basis pengetahuan. Ada dua tipe pengetahuan, yaitu: fakta dan prosedur (biasanya

berupa aturan). Salah satu fitur yang harus dimiliki oleh sistem pakar adalah kemampuan untuk menalar. Jika keahlian- keahlian sudah tersimpan sebagai basis pengetahuan dan sudah tersedia program yang mampu mengakses basis data, maka komputer harus dapat diprogram untuk membuat inferensi. Proses inferensi ini dikemas dalam bentuk motor inferensi (*inference engine*). Jadi secara umum sistem pakar terdiri atas tiga komponen utama yaitu : *Knowledge base* (Basis Pegetahuan), *Motor Inferensi*, *User Interface*.

### 2.1.2 Bentuk Sistem Pakar

Ada empat bentuk sistem pakar, yaitu :

#### 1. Berdiri sendiri

Sistem pakar jenis ini merupakan *software* yang berdiri sendiri tidak tergabung dengan *software* yang lainnya.

#### 2. Tergabung

Sistem pakar jenis ini merupakan bagian program yang terkandung di dalam suatu *algoritma* atau merupakan program dimana di dalamnya memanggil *algoritma sub* rutin lain.

#### 3. Menghubungkan ke software lain.

Bentuk ini biasanya merupakan sistem pakar yang menghubungkan ke suatu paket program tertentu, misalnya dengan DBMS.

#### 4. Sistem mengabdikan

Sistem pakar ini merupakan bagian dari komputer khusus yang dihubungkan dengan suatu fungsi tertentu. Misalnya sistem pakar yang digunakan untuk membantu menganalisis data radar.

### 2.1.3 Arsitektur Sistem Pakar

Dalam Ramadhan, Fatimah & Pane (2018) Sistem pakar disusun oleh 6 bagian utama, yaitu : Basis Pengetahuan (*Knowledge Base*), Basis Data (*Database Spreadsheet*), Antarmuka Pengguna (*User Interface*), Fasilitas Penjelasan (*Explanation Subsystem*) dan Pengguna (*User*). Berikut ini penjelasan tentang

komponen-komponen arsitektur Sistem Pakar :

1. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memahami, memformulasikan, dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar, yaitu :

a. Fakta, misalnya situasi, kondisi, atau permasalahan yang ada.

b. *Rule* (Aturan), untuk mengarahkan pengguna pengetahuan dalam memecahkan masalah.

2. Basis Data (*Database Spreadsheet*)

Digunakan sebagai media yang berfungsi untuk menampung fakta-fakta, kondisi yang diperoleh dari basis pengetahuan untuk disimpan dan diproses oleh komputer.

3. Mesin Inferensi (*Inference Engine*)

Mesin Inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi pengendalian yang berfungsi sebagai panduan arah dalam melakukan proses penalaran.

4. Antar Muka (*User Interface*)

Digunakan sebagai media komunikasi antara pengguna dan sistem pakar. Komunikasi ini paling bagus jika dalam bahasa alami dan dilengkapi dengan *graphic*, *menu*, dan formulir elektronik. Pada bagian ini akan terjadi dialog antara sistem pakar dan pengguna.

5. Fasilitas Penjelasan (*Explanation Subsystem*)

Berfungsi memberi penjelasan kepada pengguna bagaimana suatu kesimpulan dapat diambil. Kemampuan seperti ini sangat penting bagi

pengguna untuk mengetahui proses pemindahan keahlian pakar dan pemecahan masalah.

#### 6. Pengguna (*User*)

Pada umumnya pengguna sistem pakar bukanlah seorang pakar (*Non-expert*) yang membutuhkan solusi atau saran dari berbagai permasalahan yang ada.

## 2.2 Metode Inferensi

Metode inferensi adalah sebuah mekanisme berfikir dengan pola penalaran yang digunakan oleh sistem untuk mencapai suatu kesimpulan. Metode ini akan menganalisa masalah tertentu dan selanjutnya mencari kesimpulan yang terbaik. Penalaran dimulai dengan mencocokkan kaidahkaidah dalam basis pengetahuan dengan fakta yang ada (Debora, 2015) .

### 2.2.1 Forward Chaining

Nurajizah & Saputra (2018) menyatakan inferensi adalah suatu prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada suatu komponen yang disebut mesin inferensi yang mencakup prosedur- prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya.

Terdapat berbagai cara pemecahan masalah didalam sistem pakar. Beberapa hal yang perlu diperhatikan adalah arah penelusuran dan topologi penelusuran. Dalam hal ini, pemecahan masalah yang ada pada sistem menggunakan *forward chaining*. *Forward Chaining* adalah teknik pencarian yang dimulai dari inputan beberapa fakta, kemudian menurunkan beberapa fakta dari aturan-aturan yang cocok pada *knowledge base* dan melanjutkan prosesnya sampai jawaban sesuai.

*Forward Chaining* adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari

*rule* IF- THEN. Bila ada fakta yang cocok dengan bagian IF, maka *rule* tersebut dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan ke dalam database.

Setiap kali pencocokan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi. Untuk mempermudah pemahaman mengenai metode ini, akan diberikan ilustrasi kasus pembuatan sistem pakar dengan daftar aturannya sebagai berikut:

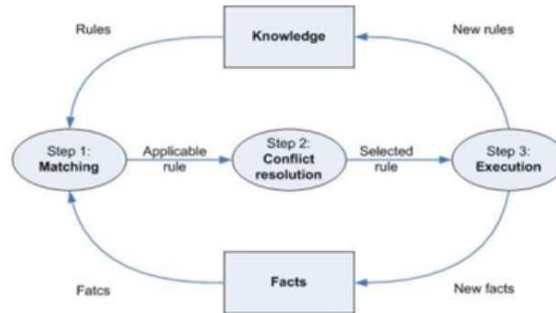
R1: Jika Premis 1 Dan Premis 2 Dan Premis 3 Maka Konklusi 1 R2:

Jika Premis 1 Dan Premis 3 Dan Premis 4 Maka Konklusi 2

R3: Jika Premis 2 Dan Premis 3 Dan Premis 5 Maka Konklusi 3

R4: Jika Premis 1 Dan Premis 4 Dan Premis 5 Dan Premis 6 Maka Konklusi 4

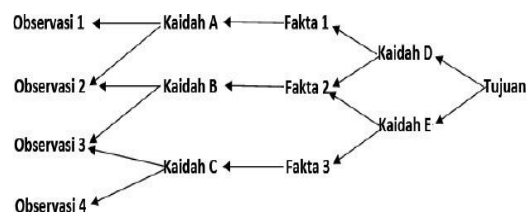
Penelusuran maju pada kasus ini adalah untuk mengetahui apakah suatu fakta yang dialami oleh pengguna itu termasuk konklusi 1, konklusi 2, konklusi 3, atau konklusi 4 atau bahkan bukan salah satu dari konklusi tersebut, yang artinya sistem belum mampu mengambil kesimpulan karena terbatas aturan. Seandainya *user* memilih premis 1, premis 2, dan premis 3, maka aturan yang terpilih adalah aturan R1 dengan konklusinya adalah konklusi 1. Seandainya *user* memilih premis 1 dan premis 6, maka sistem akan mengarah pada aturan R4 dengan konklusinya adalah konklusi 4, tetapi karena aturan tersebut premisnya adalah premis 1, premis 4, premis 5, dan premis 6, maka premis-premis yang dipilih oleh *user* tidak cukup untuk mengambil kesimpulan konklusi 4 sebagai konklusi terpilih.



Gambar 2.1 alur *Forward Chaining*

### 2.2.2 *Backward Chaining*

Metode *Backward Chaining* (Suryadi, 2011) merupakan strategi pencarian yang arahnya kebalikan dari runut maju (*forward chaining*). Proses pencarian dimulai dari tujuan, yaitu kesimpulan yang menjadi solusi permasalahan yang dihadapi. Mesin inferensi mencari kaidah-kaidah dalam basis pengetahuan yang kesimpulannya merupakan solusi yang ingin dicapai, kemudian dari kaidah-kaidah yang diperoleh, masing-masing kesimpulan dirunut balik jalur yang mengarah ke kesimpulan tersebut. Jika informasi-informasi atau nilai dari atribut-atribut yang mengarah ke kesimpulan tersebut sesuai dengan data yang diberikan maka kesimpulan tersebut merupakan solusi. Contoh gambar diagram backward chaining dapat dilihat pada gambar 2.1



Gambar 2.1 Diagram Backward Chaining

### 2.3 Ikan Channa

Prof. Dr. Ir. Eddy Suprayitno (2017) ikan Gabus atau Channa merupakan ikan air tawar yang berukuran cukup besar, dapat tumbuh hingga mencapai panjang 1 meter,

Berkepala besar agak gepeng mirip kepala ular (sehingga dinamai *snakehead*) dengan sisik-sisik besar di atas kepala. Masyarakat masih memiliki kesan bahwa makan ikan gabus sama halnya memakan ular. Ikan gabus mengandung albumin 6,2 persen dengan kandungan asam amino esensial dan asam amino non esensial yang lengkap dan berguna dalam penyembuhan pasien pasca bedah. Ikan gabus mengandung albumin yang lebih baik dari albumin telur. *Fish Albumin Ikan Gabus* dan terbukti dapat mempercepat penyembuhan luka hingga 30 % (dari rerata 10 hari menjadi 7 hari).

## **2.4 Android**

Rionaldi dan Yuni Puspita (2019) mengatakan bahwa android merupakan sebuah sistem operasi yang biasanya digunakan oleh ponsel genggam dan juga tablet. Sistem operasi ini bagaikan sebuah jembatan yang menghubungkan piranti yang digunakan dengan pengguna. Hal tersebut mengakibatkan pengguna dan piranti dapat saling berkomunikasi sehingga aplikasi-aplikasi yang terpasang dapat digunakan oleh pengguna.

## **2.5 Perangkat Lunak Pengembangan Sistem**

Pengembangan sistem untuk membangun aplikasi sistem pakar deteksi penyakit ikan *Channa* berbasis android diperlukan beberapa perangkat lunak yang digunakan dalam membangun aplikasi tersebut. Beberapa perangkat lunak yang digunakan adalah sebagai berikut:

### **2.5.1 Android studio**

Firly (2018) menjelaskan bahwa Android studio merupakan *integrated development environment* (IDE) atau dalam artian lain adalah sebuah lingkungan pengembangan terintegrasi resmi yang memang merancang khusus untuk pengembangan sistem operasi Google Android. Aplikasi ini dibangun di atas sebuah perangkat lunak yang dinamakan IntelliJ IDEA milik JetBrains. Bisa juga dibilang bahwa android studio merupakan pengganti dari *Eclipse*

*android development tool* atau ADT sebagai IDE utama dalam pengembangan aplikasi android yang asli.

Android studio diluncurkan pada tanggal 16 mei 2013 dalam konferensi google I/O yang pada saat itu masih dalam tahap pratinjau akses versi 0.1 sebagai perintis. Hingga pada akhirnya versi stabil 3.0 yang liris pada pertengahan bulan oktober 2017 dan menjadi software terlaris dikalangan *developer* muda. Aplikasi ini dapat digunakan diberbagai sistem operasi yaitu *windows, linux dan macOS*. Aplikasi ini menawarkan berbagai fitur canggih yang akan meningkatkan kemampuan produktivitas dalam proses pengembangan aplikasi. Berikut ini adalah beberapa hal yang akhirnya banyak mengundang *developer* untuk melirik android studio sebagai software pengembang :

1. Dukungan dari C++, NDK dan sekarang kotlin
2. Perkembangan yang *up to date*
3. Sistem berbasis *Gradle* yang dinilai *fleksibel*
4. Lingkungan yang mencakup seluruh perangkat android
5. Emulator yang cepat dan kaya akan fitur
6. Alat pengujian dan kerangka yang juga ekstensif
7. *Instant Run*
8. Dukungan *google cloud platform*

### 2.5.2 Java

Firly (2018) menjelaskan bahwa *Java* adalah bahasa pemrograman *multi platform*. *Java* tidak menyediakan IDE khusus seperti halnya bahasa pemrograman yang lain. Pemrogram bisa menggunakan IDE yang *support* ke *java*, misalnya *Netbeans, Eclips, TexPad*, dan lain-lain. Elemen-elemen dasar pemrograman *Java* terdiri dari Himpunan karakter, Pengenal (identifikasi), Kata Kunci, Tipe Data Primitif. Tipe data primitif yang didukung oleh bahasa pemrograman *Java* adalah *byte, short, int, long, float, double, Boolean, char*.



### 2.5.3 Adobe XD

Adobe XD adalah perangkat lunak yang bisa digunakan oleh para desainer aplikasi *mobile*. Adobe XD bisa memudahkan desainer aplikasi *mobile* dalam pengembangan UX/UI. Adobe XD ini sudah menyediakan fitur UI Desain dan juga UX Desain sebagai prototype tanpa membutuhkan *third-party* atau aplikasi lain untuk membantu membuat sebuah *prototype*. (garudapixel.com, dewaweb.com).


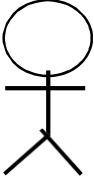

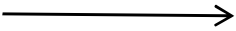



## 2.6 *Unified Modelling language (UML)*

Rosa (2018) menjelaskan bahwa *Unified Modeling Language (UML)* adalah metode perancangan berorientasi objek yang memeriksa syarat – syarat dari sudut pandang kelas-kelas dan objek yang ditemui pada ruang lingkup permasalahan dengan tujuan untuk memahami domain masalah dan meningkatkan ketelitian, konsistensi, kelengkapan analisis. *Unified Modeling Language (UML)* adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek.

### 2.6.1 Use Case Diagram

Rosa & Shalahuddin (2018) menguraikan bahwa *Use Case Diagram* merupakan deskripsi peringkat tinggi bagaimana perangkat lunak (aplikasi) akan digunakan oleh penggunanya. Selanjutnya *use case* tidak hanya penting pada tahap analisis, tetapi juga sangat penting untuk perancangan, untuk mencari kelas-kelas yang terlibat dalam aplikasi serta untuk melakukan pengujian. Diagram *use case* bersifat statis, diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Simbol – simbol *Use Case* dapat dilihat pada tabel 2.1






**Tabel 2.1** Simbol *Use Case Diagram*

Simbol	Keterangan
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit dan aktor.</p>
<p>Aktor / Actor</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi.</p>
<p>Asosiasi / Association</p> 	<p>Komunikasi antar aktor dan <i>Use Case</i> yang berpartisipasi.</p>
<p>Ekstensi / <i>extend</i></p> <p>&lt;&lt;extend&gt;&gt;</p> 	<p>Relasi Use Case tambahan ke sebuah Use Case dimana Use Case yang ditambah dapat berdiri sendiri walau tanpa <i>Use Case</i> tambahan.</p>
<p>Generalisasi / generalization</p> 	<p>Hubungan generalisasi dan spesialisasi antara dua buah <i>Use Case</i> yang mana fungsi yang satu lebih umum dari yang lainnya.</p>
<p>Include / Use Case</p> <p>&lt;&lt;include&gt;&gt;</p>  <p>Uses</p> 	<p>Relasi Use Case tambahan ke sebuah Use Case dimana Use Case yang ditambahkan memerlukan Use Case ini untuk menjalankan fungsinya.</p>

### 2.6.2 Activity Diagram

Rosa & Shalahuddin (2018) menguraikan bahwa *Activity diagram* menggambarkan *workflow* (alir kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Tahap perancangan *activity diagram* menjabarkan masing – masing *activity* pada perancangan *use case*. Simbol – simbol *Activity Diagram* dapat dilihat pada tabel 2.2.

**Tabel 2.2.** Simbol *Activity Diagram*



Simbol	Keterangan
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Status awal 	Bagaimana objek dibentuk atau diawali.
Status akhir 	Bagaimana objek dibentuk dan diakhiri.
Percabangan / join 	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.
Penggabungan / join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

### 2.6.3 Class Diagram

Class diagram menurut Melda san Sutedi (2018) *Class diagram* menggambarkan struktur sebuah sistem untuk mendefinisikan kelas-kelas yang akan dibuat unutup membuat sistem. Simbol – simbol yang ada pada *Class Diagram* ditunjukkan oleh Tabel 2.3.

Tabel 2.3. Simbol *Class Diagram*

Simbol	Keterangan
<p style="text-align: center;">Kelas</p> <div style="border: 1px solid black; padding: 2px; margin: 2px; width: fit-content; margin-left: auto; margin-right: auto;">Nama_kelas</div> <div style="border: 1px solid black; padding: 2px; margin: 2px; width: fit-content; margin-left: auto; margin-right: auto;">+atribut</div> <div style="border: 1px solid black; padding: 2px; margin: 2px; width: fit-content; margin-left: auto; margin-right: auto;">+operasi()</div>	Kelas pada struktur system.
<p style="text-align: center;">Kebergantungan / <i>dependency</i></p> <p style="text-align: center;">-----</p>	Relasi antarkelas dengan makna kebergantungan antarkelas.
<p style="text-align: center;">Agregasi / <i>aggregation</i></p> <p style="text-align: center;">—————◇</p>	Relasi antar kelas dengan makna semua bagian.
<p style="text-align: center;">Asosisasi / <i>association</i></p> <p style="text-align: center;">—————</p>	Relasi antarkelas dengan makna umum, asosiasi biasanya disertai dengan multiplicity.
<p style="text-align: center;">Asosisasi berarah / <i>directed association</i></p> <p style="text-align: center;">—————→</p>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.

<p>generalisasi</p> 	<p>Relasi antarkelas dengan makna Generalisasi spesialisasi (umum khusus).</p>
<p>Antarmuka / <i>Interface</i></p>  <p>Nama <i>interface</i></p>	<p>Sama dengan konsep interface dalam pemograman berorientasi objek.</p>

## 2.7 Penelitian Terdahulu

Berikut merupakan jurnal terkait dengan penelitian terdahulu dapat dilihat pada tabel 2.4 :

**Tabel 2.4** Penelitian Terdahulu

No	Nama	Judul	Tujuan	Metode	Kekuatan Penelitian
1	Bagus Fery Yanto, Indah Werdiningsih, Endah Purwanti	Aplikasi Sistem Pakar Diagnosa Penyakit Pada anak Bawah lima Tahun Menggunakan Metode Forward Chaining	Penelitian ini bertujuan untuk membuat sebuah aplikasi sistem pakar diagnosa penyakit pada Balita berbasis mobile	Forward Chaining	Sistem pakar dapat digunakan untuk mendiagnosa penyakit balita, misalnya sistem pakar untuk mendeteksi penyakit paru-paru kronis yang diderita oleh bayi secara dini
2	Nency Extise Putri	Sistem pakar Kerusakan hardware computer dengan	Membangun sebuah aplikasi yang akan membantu untuk mempermudah	Forward Chaining	Memperudahkan user komputer dalam memberikan solusi kerusakan

		Metode Forward Chaining	user komputer dalam memberikan solusi kerusakan computer pada hardware dengan cepat yaitu membuat sebuah sistem pakar kerusakan hardware komputer		computer pada hardware dengan cepat dan tepat
3	J Subhan	Sistem Pakar Diagnosa Penyakit Tanaman Padi Berbas Web dengan Forwad Chainin dan Backward Chaining	Pada penelitian ini dirancang sistem pakar berbasis web menggunakan basis aturan (rule based reasoning) dengan metode inferensi forward chaining dan backward chaining yang dimaksudkan untuk membantu petani dalam mendiagnosa penyakit tanaman padi	Forward Chaining dan Backward chaining	Sistem pakar diagnosa penyakit tanaman padi berbasis web yang telah dikembangkan mempunyai keunggulan dalam kemudahan akses dan kemudahan pemakaian
4	Esti Rahmawati, Hari Wibawanto	Sistem Pakar Diagnosis Penyakit Paru-paru Menggunakan Metode <i>Forward ChainingI</i>	untuk mengetahui langkah-langkah implementasi metode forward chaining ke dalam sistem pakar diagnosis penyakit paru-paru, serta untuk mengetahui	Forward Chaining	Hasil uji Blackbox dapat disimpulkan bahwa sistem pakar dapat berfungsi dengan baik

			efektivitas penggunaannya		
5	Muhamma d Burhannudi n, Suprpto, Nurul Hidayat	Pemodelan Sistem Pakar Diagnosis Penyakit Tanaman Apel Manalagi Dengan Metode Backward Chaining Menggunakan Certainty Factor	untuk pengolahan pengetahuan untuk mendapatkan kesimpulan berupa hasil diagnosis	Backwar d chaining	tingkat akurasi pengujian pemodelan system pakar menggunakan metode Certainty Factor pada sistem diagnosa penyakit tanaman apel sebesar 93,3%