APLIKASI PEMESANAN KEBUTUHAN BUDIDAYA NILA BERBASIS ANDROID MENERAPKAN SISTEM PAKAR DENGAN METODE *DEMPSTER-SHAFER* UNTUK DIAGNOSA PENYAKIT IKAN NILA

SKRIPSI

Disusun Oleh :

SUKRON MAKMUN

13.18.043

PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2019

LEMBAR KEASLIAN

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini:

Nama : Sukron Makmun

NIM : 13.18.043

Program Studi : Teknik Informatika S-1

Fakultas : Fakultas Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul :

"APLIKASI PEMESANAN KEBUTUHAN BUDIDAYA NILA BERBASIS ANDROID MENERAPKAN SISTEM PAKAR DENGAN METODE *DEMPSTER-SHAFER* UNTUK DIAGNOSA PENYAKIT IKAN NILA" Adalah skripsi sendiri bukan duplikasi serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 09 Januari 2019

Yang membuat pernyataan

Sukron Makmun

1318043

APLIKASI PEMESANAN KEBUTUHAN BUDIDAYA NILA BERBASIS ANDROID MENERAPKAN SISTEM PAKAR DENGAN METODE *DEMPSTER-SHAFER* UNTUK DIAGNOSA PENYAKIT IKAN NILA

Sukron Makmun
Teknik Informatika – ITN Malang
Sukronmakmun795@yahoo.co.od
Pembimbing : 1. Ahmad Faisol, ST, MT
2. Moh. Miftakhur Rokhman, S.kom, M.kom

ABSTRAK

Penelitian ini bertujuan untuk mengembangkan teknologi informasi brbasis Android di bidang usaha pembudidayaan ikan Nila. Dalam hal ini, CV. Naldy Putra belum memanfaatkan teknologi untuk proses pemensanan kebutuhan budidaya. Maka dari itu penulis akan mengembangkan sebuah sistem berbasis android sebagai sarana pemesanan kebutuhan budidaya. kendala lain yang sering dialami petambak adalah mengenali penyakit yang menyerang ikan nila, karena kurangnya pengetahuan terhadap penyakit tersebut.

Dari permasalah diatas, penelitian ini juga akan menerapkan sistem pakar untuk mendiagnosa penyakit ikan Nila menggunakan metode *Dempster-Shafer*, dengan memperhatikan gejala-gejala yang di alami. Penyakit yang di bahas terdiri dari 8 jenis, yaitu : *Trichodina, Epistylis, Achyla, Aeromona Hydrophilia, Pseudomonas, Streptococcosis, Dactylogyriasis, Gyrodactyliasis*. Implementasi dari metode ini berbasis Android.

Hasil penelitian menunjukkan bahwa, dengan menggunakan sistem ini pengguna dipermudah dalam proses pemesanan kebutuhan budidaya serta mengetahui jenis penyakit ikan nila berdasarkan gejala-gejala yang dialami.

Kata kunci *: Teknologi Informasi, Android, Budidaya Nila, Pemesanan, Sistem Pakar, Dempster-Shafer.*

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT atas berkat, rahmat, taufik dan hidayah-Nya, penyusunan skripsi yang berjudul "APLIKASI PEMESANAN KEBUTUHAN BUDIDAYA NILA BERBASIS ANDROID MENERAPKAN SISTEM PAKAR DENGAN METODE *DEMPSTER-SHAFER* UNTUK DIAGNOSA PENYAKIT IKAN NILA" dapat diselesaikan dengan baik. Shalawat serta salam senantiasa tercurah kepada junjungan Nabi besar Muhammad SAW beserta keluarga, sahabat, dan pengikut beliau hingga akhir zaman.

Penulis menyadari bahwa dalam proses penulisan skripsi ini banyak mengalami kendala, namun berkat bantuan, bimbingan, kerjasama dari berbagai pihak dan berkah dari Allah SWT sehingga kendala-kendala yang dihadapi tersebut dapat diatasi. Untuk itu penulis menyampaikan ucapan terima kasih dan penghargaan kepada Bapak dan Ibu yang senantiasa mendoakan, memberikan bantuan moril, materi dan nasehat selama penulis menjalani pendidikan. Selanjutnya ucapan terima kasih penulis sampaikan pula kepada:

1. Bapak Dr. Ir. Lalu Mulyadi, MTA. Selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Dr. Ir. F. Yudi Limpraptono, MT, selaku Dekan Fakultas Teknologi Industrim Insitut Teknologi Nasional Malang.
3. Bapak Joseph Dedy Irawan, ST, MT, selaku Ketua Program Studi Teknik Informatika, Institut Teknologi Nasional Malang.
4. Bapak Suryo Adi Wibowo, ST, MT, selaku Sekertaris Program Studi Teknik Informatika, Institut Teknologi Nasional Malang.
5. Bapak Ahmad Faisol, ST, MT, selaku Dosen Pembimbing I yang selalu memberikan bimbingan dan masukan.
6. Bapak Moh. Miftakhur Rokhman, S.kom, M.kom, selaku Dosen Pembimbing II yang selalu memberikan bimbingan dan masukan.
7. Semua dosen Program Studi Teknik Informatika yang telah membantu dalam penulisan dan masukan.

8. Terima kasih kepada kedua orang tua yang selalu mendoakan dan mendukung penulis dalam pengerjaan skripsi.
9. Terimakasih kepada teman-teman yang telah membantu penulis selama proses pengerjaan skripsi ini.

Dengan segala kerendahan hati, penulis menyadari masih banyak terdapat kekurangan-kekurangan, sehingga penulis mengharapkan adanya saran dan kritik yang bersifat membangun demi kesempurnaan skripsi ini.

Malang 09 Januari 2019

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi perangkat bergerak seperti Mobile phone dan Smartphone dewasa ini cukup pesat, dilihat dari banyaknya masyarakat yang menggunakan perangkat ini. Smartphone tidak hanya menyediakan fitur panggilan suara dan mengirim pesan text, pengguna perangkat ini juga bisa terkoneksi dengan Internet dan memasang berbagai macam aplikasi sesuai dengan kebutuhan (Dewi, dkk., 2018:5). Dengan melihat potensi ini, penulis ingin membangun dan memperkenalkan aplikasi untuk badan usaha pembudidayaan ikan nila CV Naldy Putra yang berlokasi di Lombok. Badan usaha ini dipilih sebagai tempat penelitian karena didalam usaha ini belum memanfaatkan teknologi sebagai sarana mempermudah aktivitas usahanya. Badan usaha ini sendiri menyediakan segala kebutuhan budidaya seperti bibit ikan, pakan dan obat-obatan bagi siapapun yang ingin bermitra dengan syarat memiliki lahan untuk membuat kolam perawatan ikan nila. Salah satu kendala yang sering dihadapi usaha ini adalah selama proses pemesanan kebutuhan budidaya oleh mitra usaha atau pemilik kolam yang dilakukan secara manual. Tentunya ini memakan waktu dan biaya. Selain itu, data-data pesanan juga masih disimpan dengan pembukuan secara manual yang seringkali mempersulit badan usaha dalam mencari dan mengolah data di kemudian hari.

Selama proses budidaya, petambak/petani seringkali mengalami kesulitan dalam memutuskan jenis obat yang akan dipesan ketika ikan terjangkit suatu penyakit. Hal ini dikarenakan kurangnya pengetahuan petambak/petani tentang informasi penyakit, gejala dan penanganan untuk penyakit tersebut. Sehingga masalah ini menyebabkan tingginya tingkat kematian ikan selama proses budidaya akibat kurangnya penanganan dan dapat berdampak pada kerugian bagi petambak/petani.

Dari permasalahan diatas, penulis akan membangun suatu aplikasi pemesanan kebutuhan budidaya dengan memanfaatkan teknologi perangkat bergerak smartphone Android untuk mempermudah para pelaku usaha budidaya nila, serta menerapkan fitur sistem pakar dengan metode Dempster-Shafer untuk mempermudah petambak/petani mengenali gejala penyakit pada ikan sehingga dapat dilakukan penanganan yang lebih baik. Metode Dempster-Shafer digunakan sebagai pengganti pakar dalam memutuskan hasil diagnosa penyakit ikan nila karena tingkat keakuratan metode ini cukup tinggi dalam memberikan hasil diagnosa sesuai dengan nilai *belief* yang diberikan terhadap masing-masing gejala penyakit.

Dengan dibangunnya aplikasi ini, diharapkan dapat membantu kedua belah pihak, yaitu perusahaan dan mitra kerja dari segi waktu dan biaya selama proses budidaya.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan sebelumnya, maka perumusan masalah yang akan dibahas sebagai berikut:

1. Bagaimana perancangan *Interface* aplikasi yang menarik agar pengguna dipermudah dalam mengoperasikannya?.
2. Bagaimana membangun aplikasi *mobile* untuk pemesanan kebutuhan budidaya ikan Nila?.
3. Bagaimana menerapkan metode *Dempster-Shafer* kedalam sistem?.
4. Bagaimana cara merancang basis data untuk penyimpanan data-data pesanan kebutuhan budidaya?

1.3. Batasan masalah

Berdasarkan rumusan masalah yang telah diuraikan diatas , maka permasalahan yang dibahas dalam aplikasi ini adalah:

1. Perancangan dan pembuatan aplikasi ditujukan untuk Cv. Naldy Putra.

2. Aplikasi diterpkan dalam *MobilePhone* Android yang terhubung dengan jaringan Internet.
3. Dalam aplikasi ini meliputi proses pemesanan kebutuhan budidaya ikan Nila.
4. Aplikasi menerapkan fitur sistem pakar dengan metode *Dempster-Shafer*.
5. Pengguna aplikasi harus memiliki Email sebagai identitas.

1.4. Tujuan

Dari masalah-masalah yang telah dirumuskan, maka dapat diketahui tujuan dari skripsi ini, yaitu :

1. Merancang sistem berbasis Android yang mudah untuk dioperasikan.
2. Membangun aplikasi berbasis android yang bisa mempermudah proses pemesanan kebutuhan budidaya.
3. Menerapkan sistem pakar dengan metode *Dempster-Shafer* untuk mengenali jenis penyakit ikan nila berdasarkan gejala yang terlihat.
4. Mempermudah Cv. Naldy Putra dalam menyimpan data pesanan kebuthan budidaya dari mitranya.

1.5. Manfaat

Manfaat yang bisa di dapatkan dengan dilaksanakannya penelitian adalah sebagai berikut :

1. Hasil penelitian ini diharapkan dapat mengurangi tingkat kematian pada ikan nila yang dibudidayakan sehingga resiko kerugian bisa di minimalisir.
2. Petambak/mitra budidaya bisa melakukan pemesanan kebutuhan budidayanya melalui perangkat Android yang dimiliki.
3. Manfaat bagi peneliti sendiri yaitu mampu menerapkan media yang sesuai dalam materi pembelajaran tertentu. Serta peneliti mempunyai pengetahuan dan wawasan mengenai materi dan media pembelajaran yang sesuai.

1.6. Metode penelitian

Berikut ini adalah langkah-langkah metode yang digunakan dalam penelitian, yaitu:

1. Studi literatur

Pada tahap ini dilakukan pendalaman teori dan pengetahuan yang berhubungan dengan sistem berbasis android. Studi literratur pertama yaitu melalui *website* di internet. Melalui bebrapa *website* yang ada di internet, informasi tentang pembuatan aplikasi berbasis android mudah ditemui. Informasi dari studi literatur ini hanya untuk pedoman sementara. Karena kebenarannya tidak dapat di pertanggungjawabkan. Selanjutnya mencari jurnal tentang pembuatan sistem berbasis android di internet. Informasi dari jurnal lebih dapat di pertanggungjawabkan kebenarannya. Namun yang didapat dari jurnal hanya berupa teori saja. Dengan adanya studi literatur ini di harapkan dapat membantu menyelsaikan permasalahan yang di hadapi.

2. Pengumpulan data

Pada tahap ini adalah proses pengumpulan data yang dibutuhkan untuk pembuatan aplikasi pemesanan dan metode diagnosa penyakit untuk ikan nila, serta melakukan analisa atau pengamatan pada data yang sudah terkumpul untuk selanjutnya diolah lebih lanjut.

3. Analisa dan perancangan sistem

Setelah selesai pada tahap pengumpulan data dan analisis maka tahap selanjutnya adalah melakukan analisa dan perancangan sistem. Pada tahap ini adalah proses perancangan dari aplikasi pemesanan yang akan dibuat untuk selanjutnya akan diproses lebih lanjut.

1.7. Sistematika Penulisan Skripsi

Sistematika dalam penulisan skripsi ini akan dibagi menjadi beberapa bab sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan pada tugas akhir ini.

BAB II TINJAUAN PUSTAKA

Pada bab ini menjelaskan tentang dasar-dasar teori yang digunakan sebagai penunjang untuk penyusunan skripsi ini. Dalam dasar teori yang akan dibahas yaitu dasar teori yang berkaitan dengan permasalahan yang diambil.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi perancangan aplikasi budidaya nila yang akan dibuat yaitu, perancangan *interface*, dan bagaimana tahap tahap dalam membangun aplikasi.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini akan membahas tentang implementasi dari metode *Dempster-Shafer* dalam mendiagnosa penyakit ikan nila dan melihat bagaimana hasilnya.

BAB V PENUTUP

Bab ini akan berisi kesimpulan dari keseluruhan laporan skripsi dan saran yang diharapkan dapat bermanfaat untuk pengembangan aplikasi kedepannya.

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam membangun aplikasi berbasis *mobile* platform *Android*. Penulis menemukan beberapa penelitian tentang bagaimana membangun aplikasi *Android* diantarannya: Rancang bangun aplikasi berbasis android dengan penerapan *web service* pada sistem informasi perpustakaan, sebuah penelitian tentang bagaimana membangun sistem informasi berbasis android (Putera, dkk., 5.1 : 57-45). Bedanya dengan penelitian penulis adalah penelitian terdahulu ini membangun sebuah sistem informasi untuk perpustakaan, sedangkan penulis akan membangun aplikasi pemesanan untuk industri usaha CV Naldy putra. Kedua, penelitian tentang Rancang bangun aplikasi pemesanan tiket online kapal laut berbasis android(Sede, dkk., 2015:4). Penelitian ini dilakukan untuk mempermudah proses pemesanan tiket online untuk kapal laut. Terdapat kesamaan Antara penelitian penulis yaitu sama sama membangun aplikasi pemesanan yang dilakukan secara online melalui perangkat bergerak *mobile* platform *android.* Ketiga, penelitian tentang diagnosa penyakit ikan nila menggunakan metode *dempster-shafer* berbsis *website*(Yusnita, dkk., 2016). Penelitian ini memiliki kesamaan dengan penelitian yang dilakukan penulis yaitu sama-sama membangun sistem untuk diagnosa penyakit ikan nila menggunakan metode *dempster-shaer.* Bedanya adalah sitem yang dibangun Yusnita dkk berbasis *website,* sedangkan penulis berbasis android. Selain itu, data-data nilai *belief* gejala terhadap penyakit didapatkan penulis dari penelitian ini.

2.2. CV. Naldy Putra

CV. Naldy Putra merupakan salah satu badan usaha di Lombok yang membudidayakan dan mendistribusikan ikan air tawar konsumsi jenis Nila. Usaha ini menawarkan kemudahan bagi siapa saja yang ingin bermitra dan bergabung dalam membudidayakan ikan nila dengan syarat memiliki lahan untuk membuat kolam. Selama proses budidaya, kebutuhan seperti bibit, pakan dan obat-obatan akan disediakan oleh badan usaha. Bibit ikan nila yang diberikan kepada mitra kerja bervariasi, tergantung dari luas kolam yang dimiliki, mulai dari 8000 ekor untuk kolam ukuran 50x50 m^2 sampai 30.0000 ekor untuk kolam ukuran 300 m^2. Hingga saat ini, CV Naldy putra sudah memiliki mitra kerja sekitar 250 anggota yang tersebar di wilayah Lombok tengah dan Lombok barat.

Selama proses budidaya. Proses pemesanan kebutuhan budidaya dilakukan secara manual. Dimana mitra atau anggota usaha harus datang ke kantor untuk memesan kebutuhan budidaya seperti bibit, pakan atau obat-obatan. Atau bias juga melalui telepon, hal ini masih menjadi salah satu kendala bagi badan usaha karena seringkali terjadi kekeliruan dalam menyimpan data-data pesanan mengingat banyaknya mitra dari usaha ini. Akibatnya, penanganan pesanan seringkali mengalami keterlambatan. Untuk pembagian keuntungan dilakukan setiap kali panen.

2.3. Metodologi

2.3.1. Sistem pakar

Sistem pakar merupakan cabang dari *Artificial intelligent* (AI). Implementasi sistem pakar banyak digunakan untuk kepentingan komersial karena sistem pakar dipandang sebagai cara penyimpanan pengetahuan pakar dalam bidang tertentu ke dalam program sehingga komputer dapat memberikan keputusan dan melakukan penalaran secara cerdas(Sulistyohati dan Hidayat., 2008: 12).

Kecerdasan buatan sebagaimana telah diketahui, saat ini merupakan suatu inovasi baru dalam bidang ilmu pengetahuan. Kecerdasan buatan atau *Artificial Intelligent* merupakan salah satu bagian ilmu komputer yang membuat agar mesin (*computer*) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Teknologi kecerdasan buatan dipelajari dalam bidang-bidang seperti : robotika, penglihatan komputer (*computer vision*), jaringan saraf tiruan (*artificial neural system*), pengolahan bahasa alami *(natural language processing*), pengenalan suara (*speech recognition*) dan sistem pakar (*expert system*).

Program ini bertindak sebagai seorang konsultan yang cerdas dalam suatu keahlian tertentu. Sehingga seorang user dapat melakukan konsultasi kepada komputer, seolah-olah user tersebut berkonsultasi kepada seorang ahli. Komputer harus dapat dengan efektif menggunakan pengetahuan heuristik, pengetahuan harus dibuat dalam format yang mudah diakses yang membedakan antara data, pengetahuan, dan kontrol struktur(Yusnita dan Aprilianto., 2016:8).

Konsep-konsep dasar dari sebuah sistem pakar adalah:

1. Keahlian (*Expertise*): Keahlian merupakan pengetahuan khusus yang dimiliki oleh seseorang melalui latihan, belajar, serta pengalaman-pengalaman yang dialami pada suatu bidang tertentu dalam jangka waktu yang cukup lama. Dengan pengetahuan tersebut seorang pakar dapat memberikan keputusan yang lebih baik dan cepat dalam menyelesaikan suatu permasalahan yang sulit.

2. Ahli atau pakar (*Expert*): Seorang pakar harus memiliki kemampuan menyelesaikan permasalahan pada bidang tertentu yang ditanganinya, kemudian memberikan penjelasan mengenai hasil dan kaitannya dengan permasalahan yang ada. Untuk meniru kepakaran seorang manusia, perlu dibangun sebuah sistem komputer yang menunjukan seluruh karakteristik tersebut. Namun hingga saat ini, pekerjaan dibidang sistem pakar terfokus pada aktifitas penyelesaian masalah dan memberikan penjelasan mengenai solusinya.
3. Memindahkan Keahlian (*Transfering Expertise*): Tujuan dari sistem adalah memindahkan keahlian yang dimiliki oleh seorang pakar ke dalam sebuah sistem komputer, kemudian dari sebuah system computer kepada orang lain yang bukan pakar. Proses ini dapat meliputi empat kegiatan :
   a. Perolehan pengetahuan (*Knowledge Acquistion*).
   b. Representasi pengetahuan (*Knowledge Representation*).
   c. Menyimpulkan pengetahuan (*Knowledge Inferencing*).
   d. Memindahkan pengetahuan kepada pemakai (*Knowledge transfer to user*).
4. Kesimpulan (*Inference*): Keistimewaan dari sistem pakar adalah kemampuannya dalam memberikan saran, yaitu dengan menempatkan keahlian ke dalam basis pengetahuan dan membuat program yang mampu mengakses basis pengetahuan sehinggga sistem dapat memberikan kesimpulan. Kesimpulan dibentuk di dalam komponen yang dinamakan mesin pengambil kesimpulan (*Inference Engine*), dimana berisi aturan-aturan untuk menyelesaikan masalah.
5. Aturan (*Rule*): Umumnya sistem pakar adalah sistem berbasis aturan, yaitu pengetahuan yang terdiri dari aturan-aturan sebagai prosedur penyelesaian masalah. Pengetahuan tersebut digambarkan sebagai suatu urutan seri dari kaidah- kaidah yang sudah dibuat.

6. Kemampuan Penjelasan (Explanation Capability): Keistimewaan lain dari sistem pakar adalah kemampuannya dalam memberikan saran atau rekomendasi serta menjelaskan mengapa tindakan tertentu tidak dianjurkan. Pemberian penerangan dan pendapat ini dilakukan dalam suatu subsistem yang dinamakan subsistem penjelasan (*explanation subsystem*).

2.3.2. Dempster Shafer

Metode *Dempster-Shafer* pertama kali diperkenalkan oleh Dempster, yang melakukan percobaan model ketidakpastian dengan range probabilitas sebagai probabilitas tunggal. Kemudian pada tahun 1976 Shafer mempublikasikan teori Dempster tersebut pada sebuah buku yang berjudul *Mathematical Theory of Evident*.

Secara umum, teori *Dempster-Shafer* ditulis dalam suatu *Interval* : (*Belief, Plausibility*).

Belief (Bel) adalah ukuran kekuatan *evidence* dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengindikasikan bahwa tidak ada *evidence*, dan jika bernilai 1 menunjukkan adanya kepastian. fungsi belief dapat dirumuskan pada Persamaan 1 :

$$Bel(X) = \sum_{Y \subseteq X} m(Y)$$

sedangkan *Plausibility* (Pls) dirumuskan pada Persamaan 2 :

$$Pl(s) = 1 - Bel(s') = 1 - \sum_{Y \subseteq X'} m(s')$$

Dimana :

Bel(X) = *Belief* (X)

Pls(X) = Plausibility (X)

m(X) = *mass function* dari (X)

m(Y) = *mass function* dari (Y)

Plausibility juga bernilai 0 sampai 1, jika yakin akan X' maka dapat dikatakan *Belief* (X') = 1 sehingga dari rumus nilai *Pls* (X) = 0.

Pada teori *Dempster-Shafer* juga dikenal adanya *frame of discernment* (FOD). yang dinotasikan dengan Θ . FOD ini merupakan semesta pembicaraan dari sekumpulan hipotesis sehingga sering disebut dengan *environment* dapat dirumuskan pada Persamaan 3 :

$$\Theta = \{\theta 1, \theta 2, \ldots\ldots \theta n\}$$

Dimana :

Θ = FOD atau *environment*

θ1....θn = elemen/unsur bagian dalam *environment*

Environment mengandung elemen-elemen yang menggambarkan kemungkinan sebagai jawaban dan hanya ada satu yang akan sesuai dengan jawaban yang dibutuhkan. Kemungkinan ini dalam teori *Dempster-Shafer* disebut dengan *power set* dan dinotasikan dengan P(Θ), setiap elemen dalam *power set* ini memiliki nilai interval antara 0 sampai 1, sehingga dapat dirumuskan pada Persamaan 4 :

$$m = P(\Theta) \longrightarrow [0,1]$$

sehingga dapat dirumuskan seperti Persamaan 5:

$$\sum_{X \subset P(\Theta)} m(X) = 1 \approx \sum_{X \subset P(\theta)} m(X) = 1$$

dengan P(Θ) = *power set* dan m(X) = *mass function* dari (X)

Pada aplikasi sistem terdapat sejumlah *evidence* yang akan digunakan pada faktor ketidakpastian dalam pengambilan keputusan untuk diagnosa suatu penyakit. Untuk mengatasi sejumlah *evidence* tersebut pada teori *Dempster-Shafer* bisa dilihat pada persamaan 5 menggunakan aturan yang lebih dikenal dengan *Dempster's Rule of Combination,* yaitu pada Persamaan 6 :

$$m1 \oplus m2(Z) = \sum_{X \cap Y = Z} m1(X) m2(Y)$$

dimana:

m1⊕ m2(Z) = *mass function* dari *evidence*(Z)

m1(X) = *mass function* dari *evidence*(X)

m2(Y) = *mass function* dari *evidence* (Y)

⊕ = *operator direct sum*

Secara umum formulasi untuk *Dempster's Rule of Combination* bisa dilihat pada Persamaan 7 :

$$m1 \oplus m2(Z) = \frac{\sum_{X \cap Y = Z} m1(X) m2(Y)}{1 - k}$$

dimana:

k = Jumlah evidential *conflict*. Besarnya jumlah *evidential conflict* (k) dirumuskan dengan Persamaan 8 :

$$k = \sum_{X \cap Y = \theta} m1(X) m2(Y)$$

sehingga bila Persamaan 8 disubstitusikan ke Persamaan 9 :

$$m1 \oplus m2(Z) = \frac{\sum_{X \cap Y = Z} m1(X) m2(Y)}{1 - \sum_{X \cap Y = \theta} m1(X) m2(Y)}$$

Dimana :

m1⊕ m2(Z) = *mass function* dari *evidence* (Z)

m1(X) = *mass function* dari *evidence* (X)

m2(Y) = *mass function* dari *evidence* (Y)

k= jumlah *evidential conflict*

2.4. Dasar teori

2.4.1. Budidaya Perikanan

Dalam membangun aplikasi, penulis menerapkan teknologi informasi pada industri usaha budidaya perikanan. Budidaya perikanan adalah usaha pemeliharaan dan pengembang biakan ikan atau organisme air lainnya. Budidaya perikanan disebut juga sebagai budidaya perairan atau akuakultur mengingat organisme air yang dibudidayakan bukan hanya dari jenis ikan saja tetapi juga organisme air lain seperti kerang, udang maupun tumbuhan air.

Dilihat dari asal katanya, istilah akuakultur diambil dari istilah dalam Bahasa Inggris yaitu *Aquaculture.* Terdapat beberapa definisi akuakultur seperti dikemukakan dalam beberapa sumber, dan berikut ini adalah definisi akuakultur menurut ahli :

Akuakultur merupakan upaya produksi biota atau organisme perairan melalui penerapan teknik domestikasi (membuat kondisi lingkungan yang mirip dengan habitat asli organisme yang dibudidayakan), penumbuhan hingga pengelolaan usaha yang berorientasi ekonomi(Putra, dkk., 2016:10).

Berdasarkan kata penyusunnya budidaya perikanan tentunya tersusun dari dua kata yakni budidaya dan perikanan. Menurut Kamus Besar Bahasa Indonesia Budidaya adalah usaha yang bermanfaat dan memberikan hasil, Sedangkan perikanan adalah segala sesuatu yang berhubungan dengan penangkapan, pemeliharaan dan pembudidayaan ikan. Jadi budidaya perikanan adalah usaha pemeliharaan ikan guna mendapatkan manfaat atau hasil.

2.4.2. Android

Aplikasi yang akan dibangun berjalan pada perangkat bergerak mobile *Smartphone android.* Android adalah sistem operasi yang digunakan di smartphone dan juga tablet PC. Android tidak terikat ke satu merek Handphone saja, beberapa vendor terkenal yang sudah memakai Android antara lain

Samsung, Sony Ericsson, HTC, Nexus, Motorolla, dan lain-lain. Keunggulan utama Android adalah gratis dan *open source*, Beberapa fitur utama dari Android antara lain WiFi hotspot, Multi-touch, Multitasking, GPS, accelerometers, support java,mendukung banyak jaringan serta juga kemampuan dasar handphone pada umumnya .

2.4.3. Android studio

Android studio digunakan sebagai media perancangan aplikasi. Android Studio adalah sebuah IDE untuk Android Development yang dimiliki oleh google. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android.

2.4.4. Firebase

Sistem yang dibangun akan memanfaatkan fitur yang disediakan oleh Firebase seperti firebase authentication untuk pendaftaran user menggunakan email, dan layanan real-time database untuk menyimpan data-data pesanan dan anggota usaha budidaya ikan CV. Naldy Putra.

Firebase adalah BaaS (Backend as a Service) yang saat ini dimiliki oleh Google. Firebase ini merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan Mobile Apps Developer. Dengan adanya Firebase, apps developer bisa fokus mengembangkan aplikasi tanpa harus memberikan effort yang besar untuk urusan backend.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis sistem

Tahap analisis sistem ini akan menganalisis mulai dari tahap tahap awal pembuatan aplikasi

1. Analisis masalah

   Aplikasi yang dibuat akan membantu user dalam memesan kebutuhan budidaya ikan nila melalui perangkat *mobile android*. Selain itu, user juga bisa melakukan diagnosa terhadap ikan yang dibudidayakan menggunakan aplikasi tanpa harus melakukan konsultasi langsung ke pakar atau mencari solusinya di internet. Dengan adanya aplikasi ini user bisa sedikit dipermudah dalam proses budidaya ikan Nila.

2. Analisis kebutuhan

   Dalam perancangan aplikasi, data yang akan diolah adalah data nilai gejala terhadap jenis penyakit ikan nila. Data ini memiliki peran yang sangat penting dalam pembuatan aplikasi, mengingat data ini akan digunakan untuk menentukan jenis penyakit pada ikan nila berdasarkan gejala-gejala yang dialami. Data gejala ini didapatkan dari sebuah jurnal yang berjudul “Sistem pakar untuk diagnosa penyakit ikan nila berbasis Website”.

3. Analisis target user

   Aplikasi yang dibuat ditujukan untuk pelaku usaha di CV. Naldy putra yaitu pembudidaya sebagai pemesan dan pemilik usaha sebagai penerima pesanan dari mitra.

4. Analisis kebutuhan perangkat

   Analisis kebutuhan perangkat menjelaskan tentang spesifikasi perangkat keras dan perangkat lunak yang dibutuhkan untuk menjalankan dan mengembangkan aplikasi yang akan dibangun. Kebutuhan perangkat dibagi menjadi 3 yaitu :

1. Perangkat keras

Dalam pengembangan aplikasi ini menggunakan Android Studio. Untuk menjalankan Android Studio, dibutuhkan perangkat keras (*hardware*) yang cukup bagus untuk dapat melakukan proses yang lebih cepat. Spesifikasi perangkat keras yang dibutuhkan untuk membangun aplikasi ini dapat dilihat pada tabel 3.1. spesifikasi perangkat keras berikut.

Tabel 3.1. Spesifikasi perangkat keras.

Nama Perangkat	Spesifikasi
Prosesor	Mininum Intel Core i3 4720HQ
Memory	RAM 4 GB
Resolusi layar	Minimum 1280x800
Mouse	Logitech G102
Harddisk	Minimum 4 GB

2. Perangkat lunak

Dalam pengembangan aplikasi ini, perangkat lunak digunakan untuk proses pembuatan user interface dan coding program. Perangkat lunak yang dibutuhkan dalam pengembangan aplikasi ini dapat dilihat pada Tabel 3.2 Spesifikasi perangkat lunak berikut.

Tabel 3.2. Spesifikasi perangkat lunak.

Nama Perangkat	Spesifikasi
OS	Windows 10
Aplikasi	Google Chrome Android Studio 3.2

3. Perangkat android

Agar aplikasi dapat dijalankan dengan baik pada sistem operasi Android, maka dibutuhkan perangkat Android yang sesuai dengan kebutuhan aplikasi. Kebutuhan perangkat androi dapat dilihat pada Tabel 3.3 kebutuhan perangkat android berikut.

Tabel 3.3. Kebutuhan perangkat android.

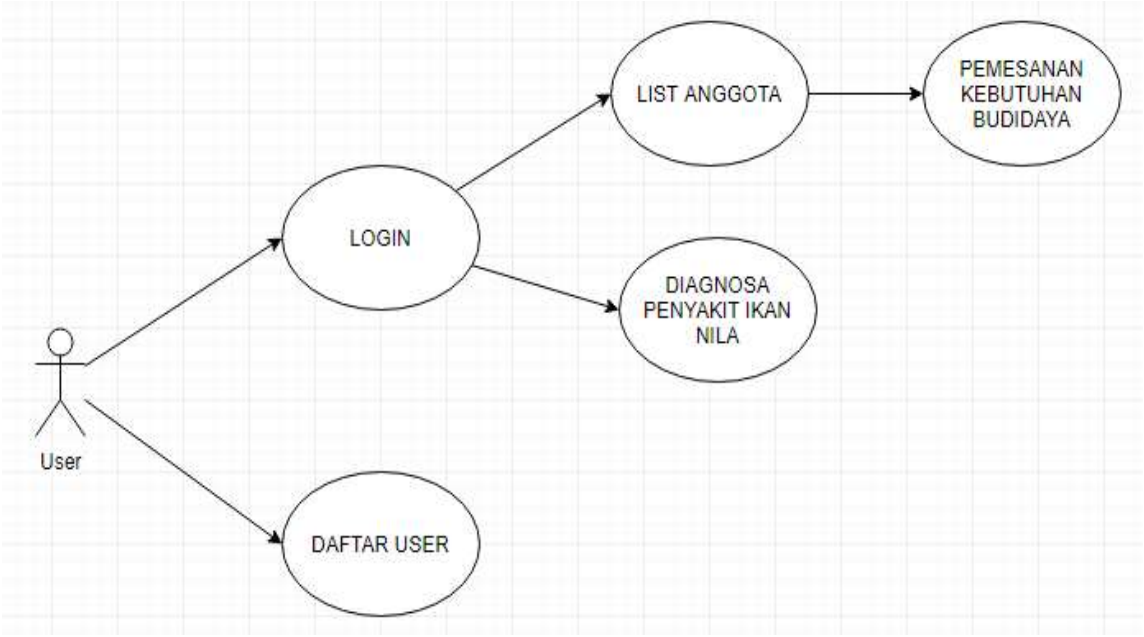
Nama Perangkat	Minimum Spesifikasi	Maximum Spesifikasi
Prosesor	Quad-Icore 1.2 GHz Cortex-A7	Exynos 8895 Octa
Memory	RAM 1 GB	RAM 4 GB
GPU	Adreno 302	Adreno 540
Resolusi Layar	720 x 1280px	1440 x 2960px
OS	Android KitKat (4.1)	Android Oreo (8.0)

3.2. Desain sistem

Perancangan sistem dilakukan untuk mengumpulkan informasi yang berhubungan dengan aplikasi yang akan dibangun dan juga untuk memudahkan pemahaman terhadap sistem.

3.2.1. Use Case Diagram User

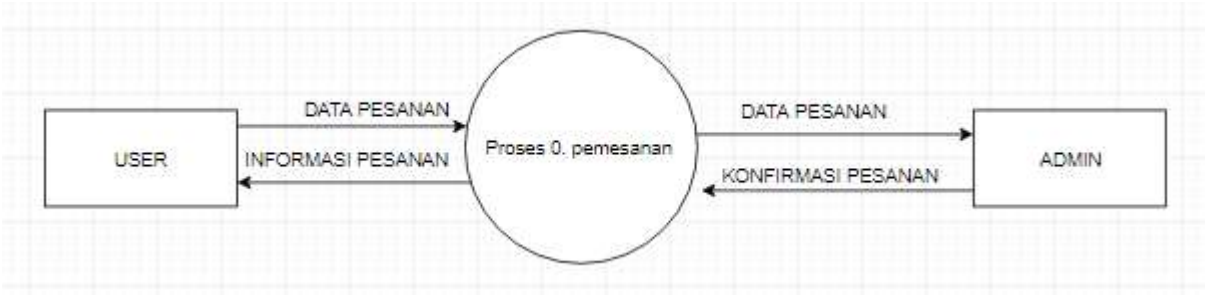
Perancangan use case diagram user berfungsi untuk mengetahui bagaimana alur program saat dijalankan. Use case diagram dapat dilihat pada gambar 3.2.1 berikut:



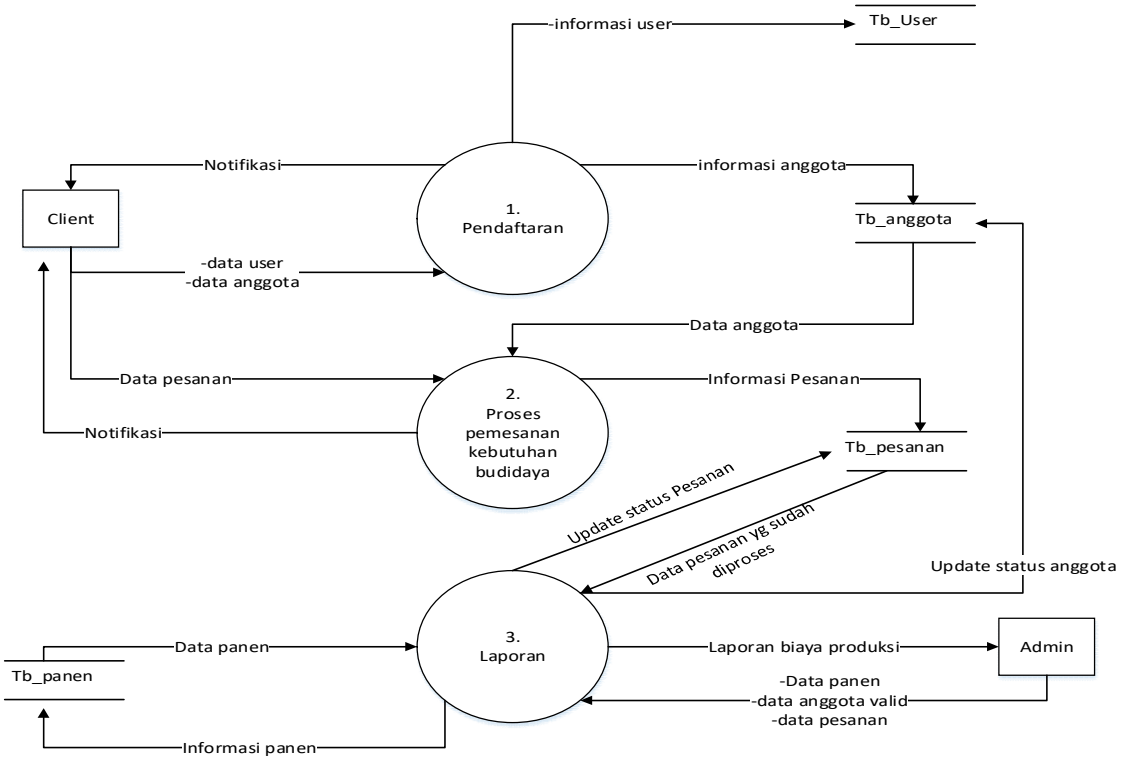
Gambar 3.1. Use Case Diagram User.

3.2.2. DFD

Perancangan DFD (Data Flow Diagram berfungsi untuk mengetahui alur data pada sistem yang akan dibangun. DFD pemesanan kebutuhan budidaya nila dapat dilihat pada gambar berikut:



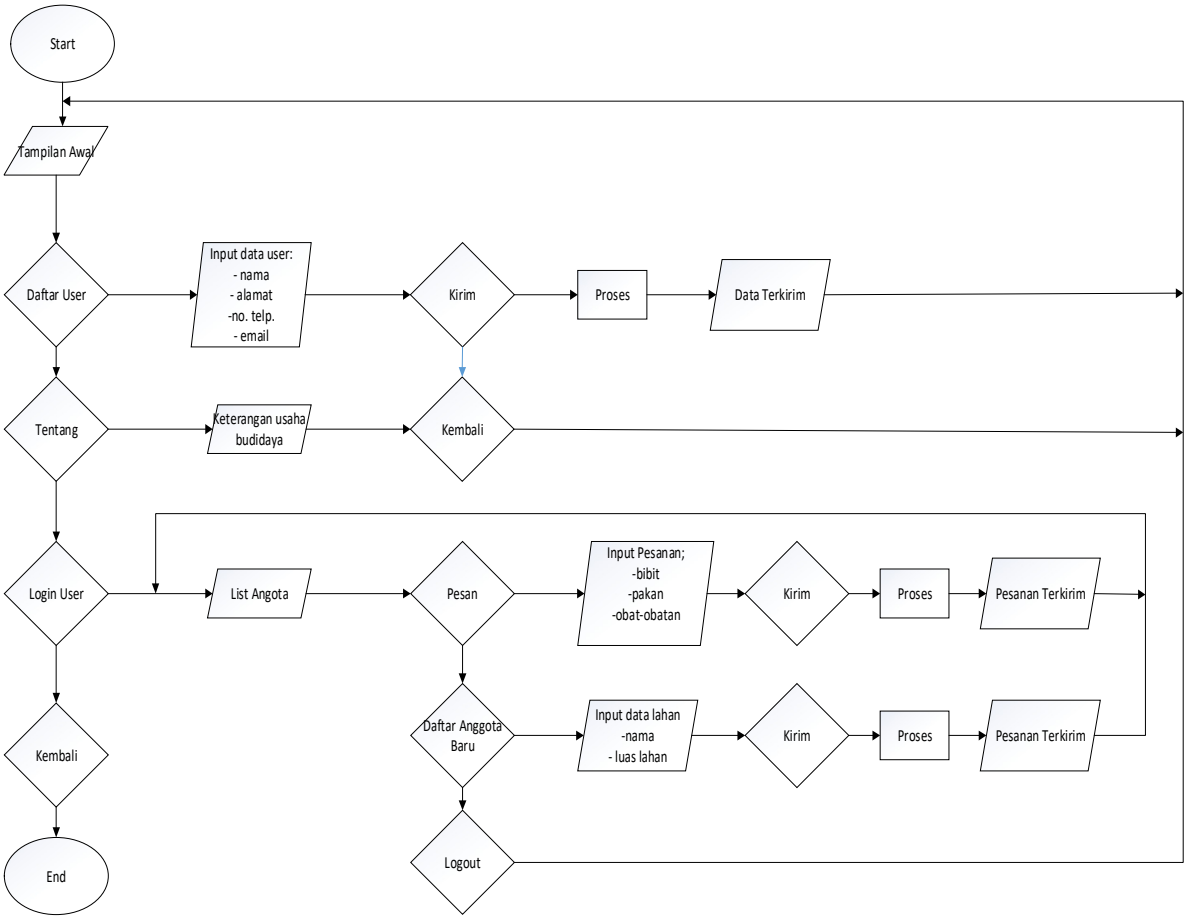
Gambar 3.1. DFD Lv. 0



Gambar 3.2. DFD Lv. 1

3.2.3. Flowchart User

Perancangan flowchart berfungsi untuk mengetahui alur program dari awal program dijalankan sampai program berakhir. Flowchart sistem pemesanan kebutuhan budidaya dapat dilihat pada gambar berikut:



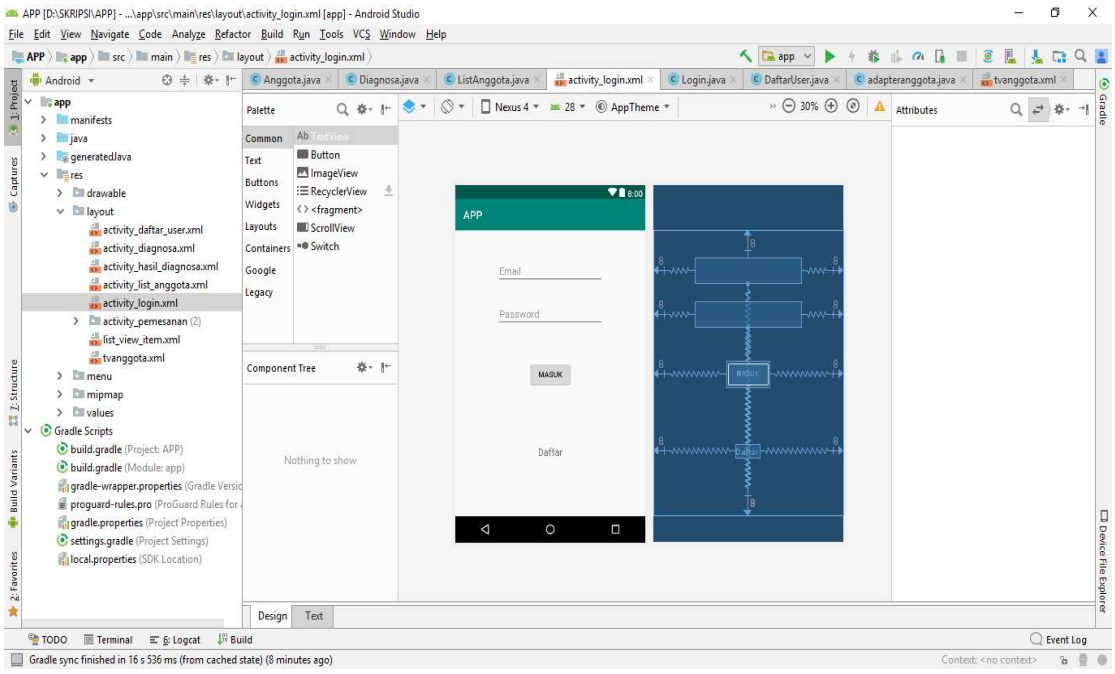
Gambar 3.1. Flowchart User

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi aplikasi

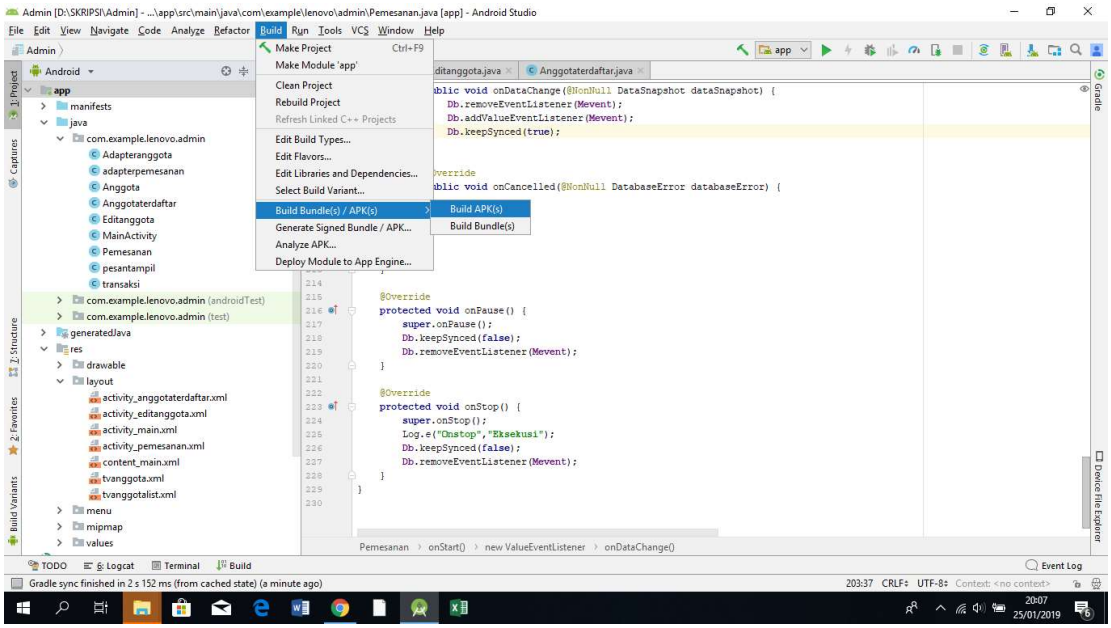
Implementasi aplikasi adalah proses penerapan rancangan sistem yang telah dibuat menjadi satu aplikasi yang bisa dijalankan pada kenyataannya. Disamping itu, implementasi ini juga berfungsi untuk mengetahui tingkat keberhasilan dari rancangan yang telah dibuat. Fungsi dari jendela Android studio dapat dilihat pada gambar 4.1. pembuatan layout untuk login User



Gambar 4.1. Membuat Layout di Android Studio.

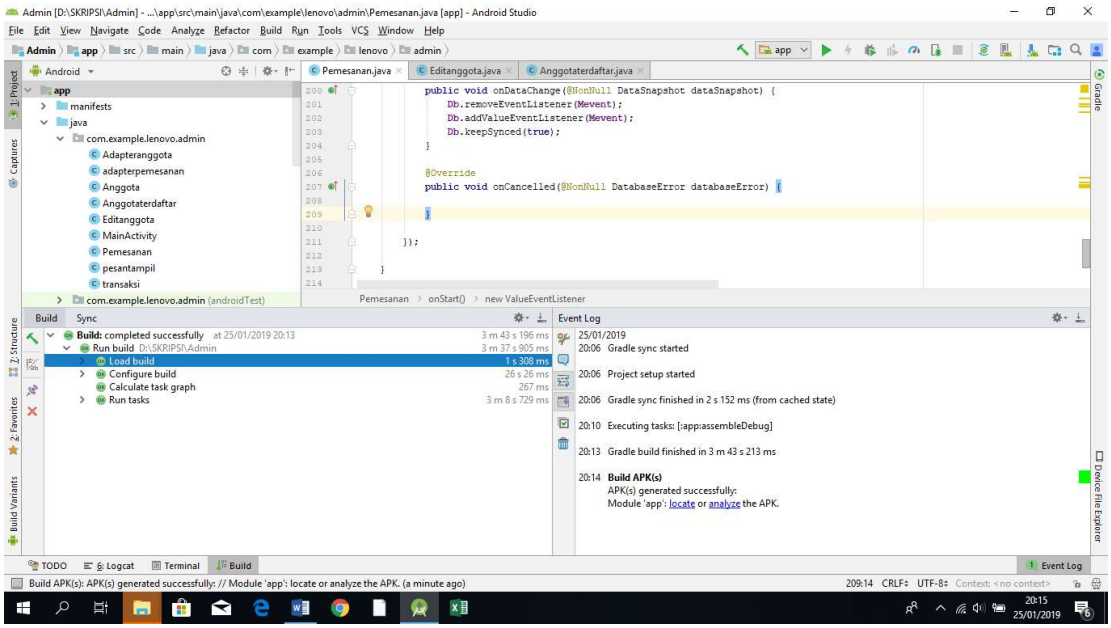
Setelah pembuatan project selesai dikerjakan, langkah selanjutnya adalah membuat file APK (*Application Package File*) untuk project agar bisa digunakan/ diinstal kedalam moile android. Langkah-langkah untuk membuat file APK di Android Studio adalah :

1. klik menu Build > Build Bundle(s) / APK(s) > Build APK (s) seperti pada gambar 4.2 berikut :



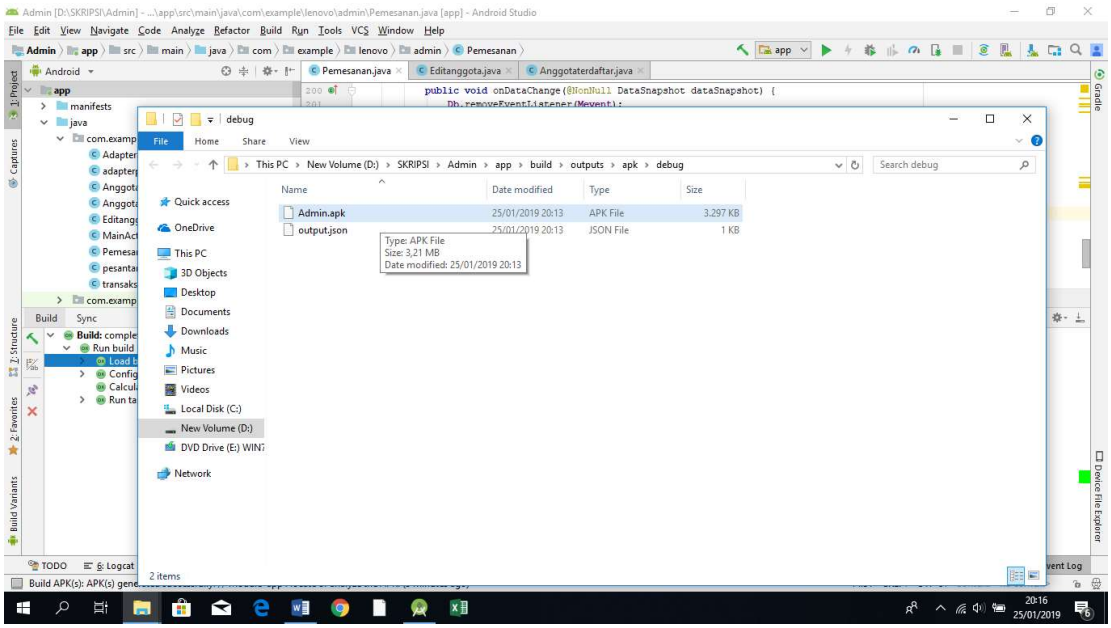
Gambar 4.2. Membuat file APK

2. Tunggu proses build project selesai (membutuhkan waktu yang berbedda tergantung spesifikasi komputer yang digunakan untuk membuat project) sampai ada keterangan generated succsessfully seperti pada gambar 4.3 berikut



Gambar 4.3. Proses build APK selesai dilakukan

3. Kemudian klik location pada event log untuk menampilkan lokasi dari file APK project yang sudah dibuat dan bisa dicopykan ken perangkat android untuk diinstall dan digunakan. Lokasi file APK yang sudah dibuat dapat dilihat pada gambar 4.4 berikut:



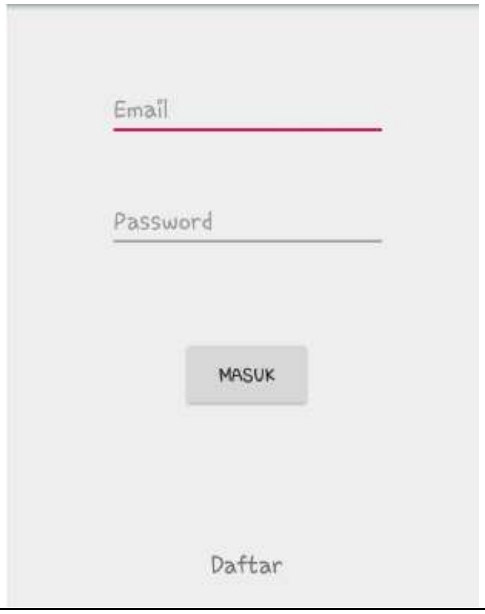
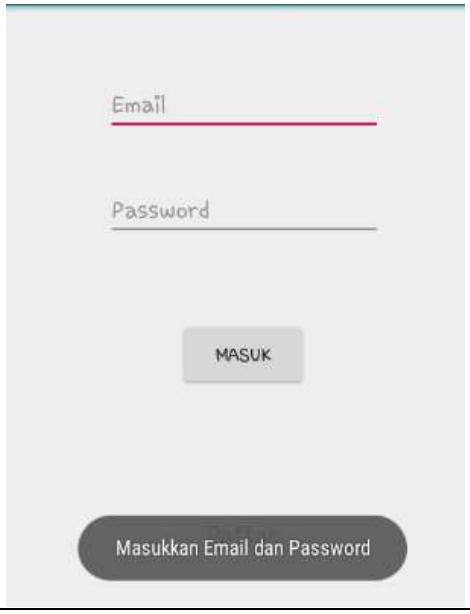
Gambar 4.4. Lokasi file APK yang sudah dibuat.

4.2. Pengujian Aplikasi

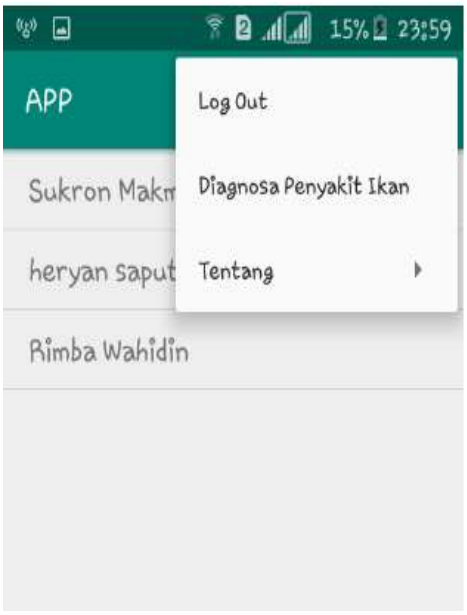
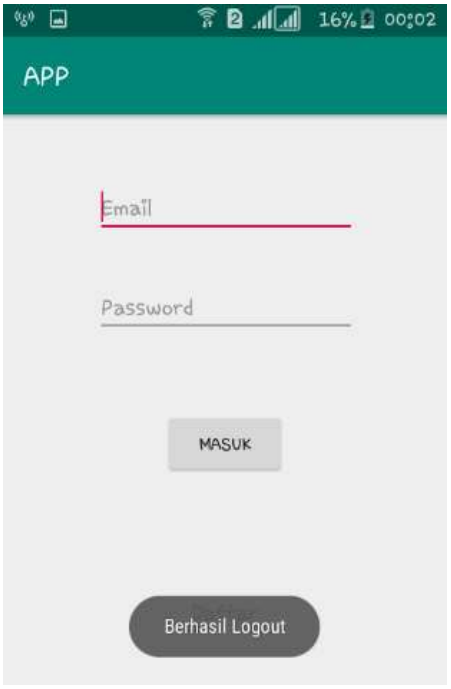
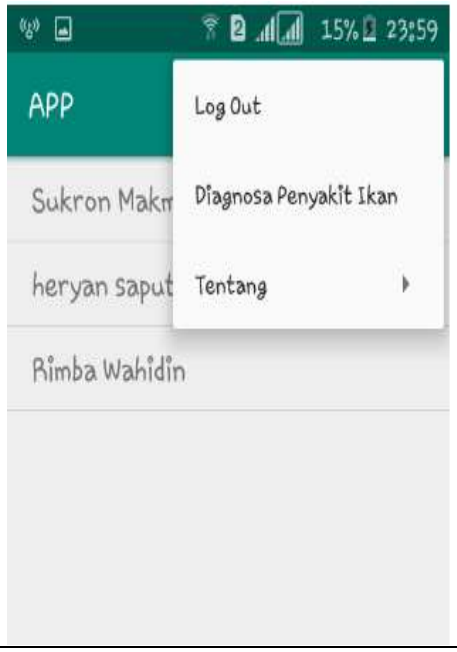
4.2.1. Pengujian blackbox

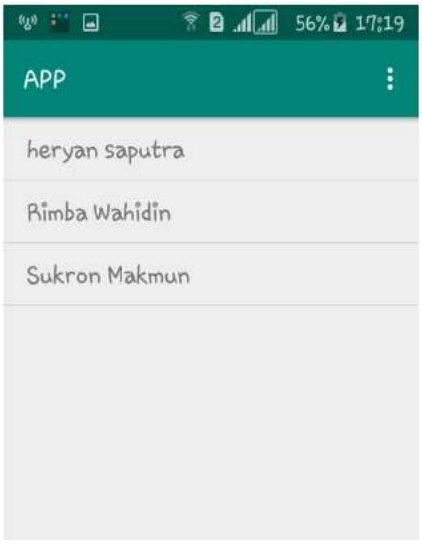
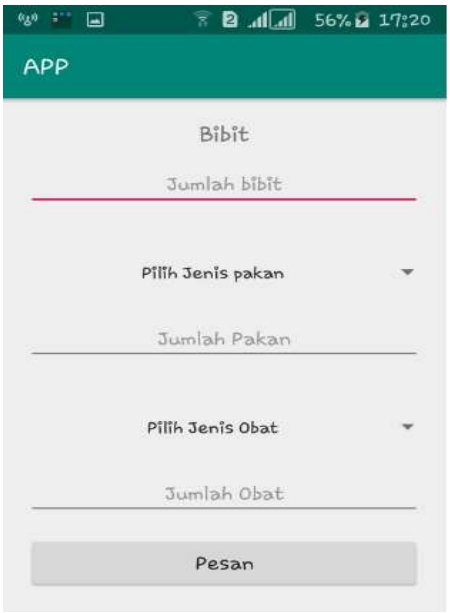
Pengujian yang dilakukan terhadap aplikasi dilakukan dengan metode Balckbox untuk mengetahui apakah aplikasi sudah berjalan sesuai dengan yang diinginkan atau belum. Pengujian aplikasi dapat dilihata pada tabel 4.1 pengujian aplikasi User dengan metode blackbox berikut :

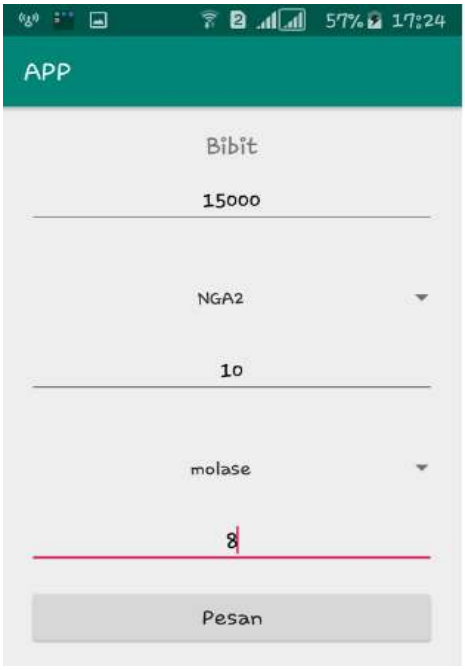
Tabel 4.1. pengujian aplikasi User dengan metode blackbox.

No.	Skenario	Hasil yang diharapkan	Kesimpulan
1	Mengosongkan Email dan Password pada menu login User, lalu langsung klik tombol masuk 	Sistem akan menolak hasil login dan menampilkan pesan "Masukkan Email dan Password" 	Valid

No.	Skenario	Hasil yang diharapkan	Kesimpulan
2	Memasukkan Email dan Password user yang sesuai kemudian klik tombol Masuk 	Sistem akan masuk ke menu List Anggota dan mengeluarkan pesan "Login Berhasil" 	Valid
3	Memilih Logout pada menu list anggota	Sistem akan Logout dari list anggota dan kembali ke menu Login dengan pesan "Berhasil Logout	Valid

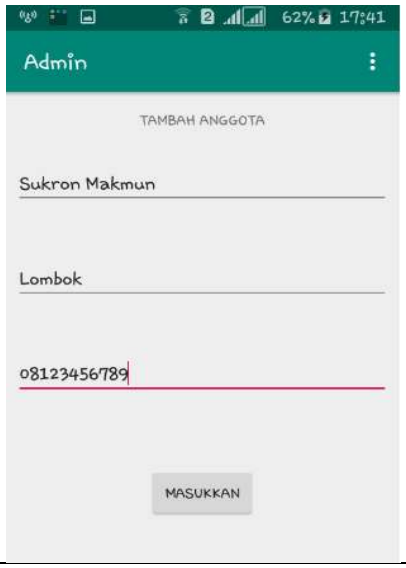
No.	Skenario	Hasil yang diharapkan	Kesimpulan
			
4	Memilih Diagnosa Penyakit Ikan pada menu List Anggota 	Sistem akan berpindah ke menu diagnosa dan mengeluarkan pesan “diagnosa” 	Valid

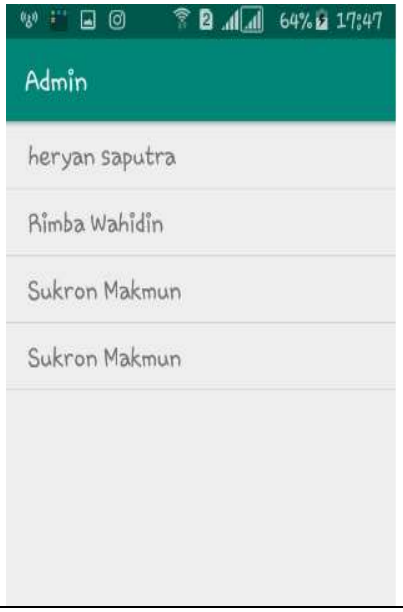
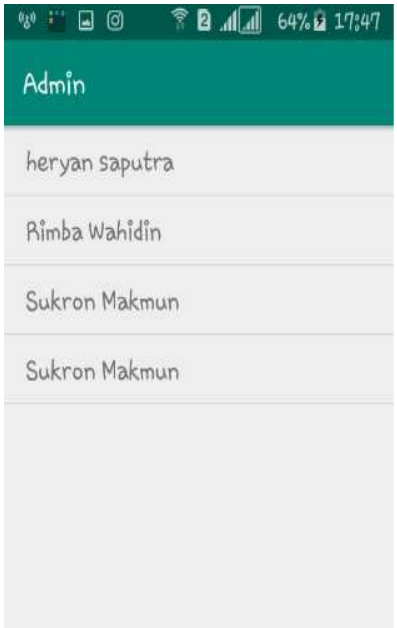
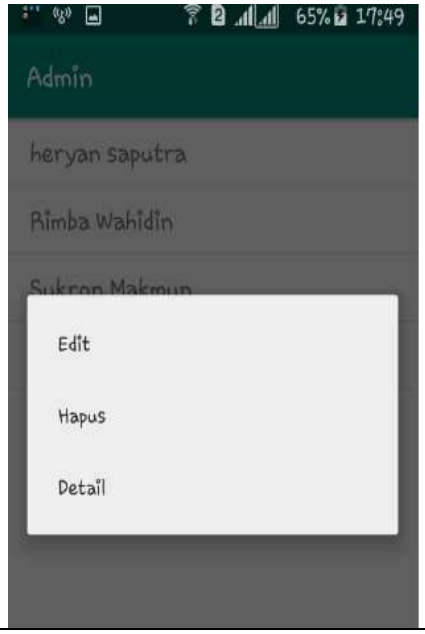
No.	Skenario	Hasil yang diharapkan	Kesimpulan
5	Memilih Gejala Penyakit pada menu diagnosa 	Sistem akan menampilkan nama penyakit dan persentase berdasarkan gejala yang dipilih 	Valid
6	Memilih anggota yang akan memesan kebutuhan budidaya pada menu List anggota 	Sistem akan beralih k menu input pesanan 	Valid

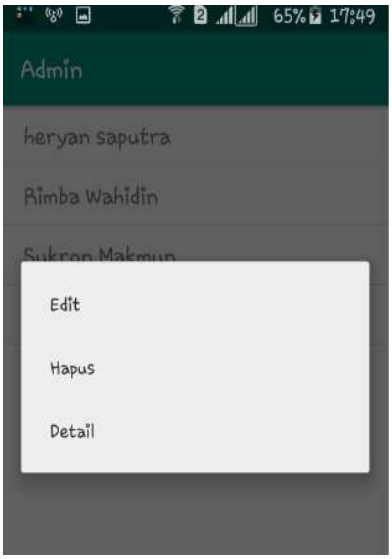
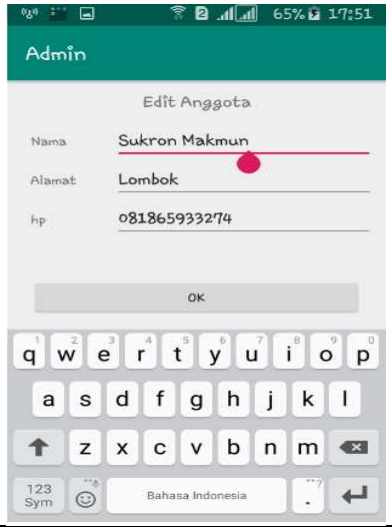
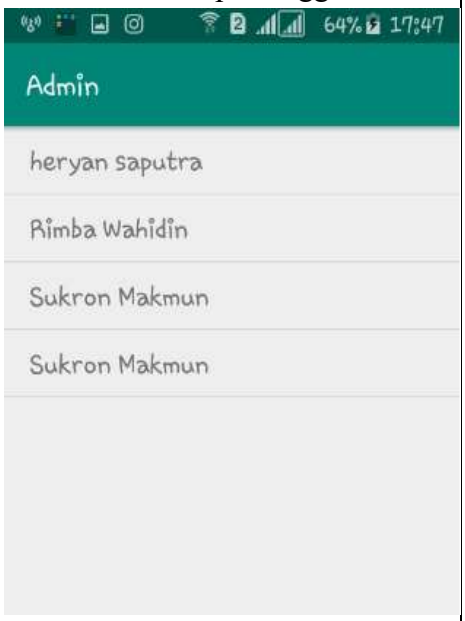
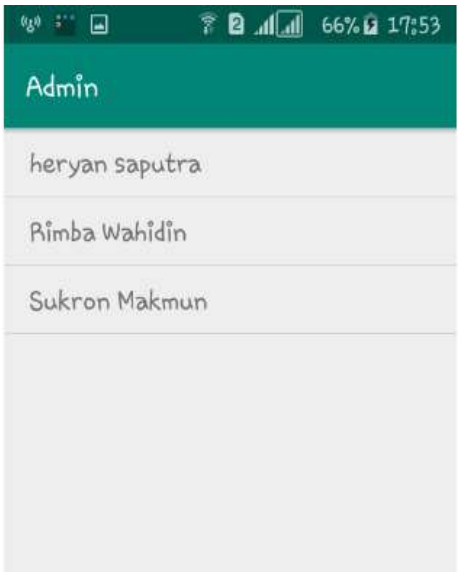
No.	Skenario	Hasil yang diharapkan	Kesimpulan
7	Memasukkan data pesanan dan menekan tombol pesan 	Sistem akan mengirim data pesanan kepada admin 	valid

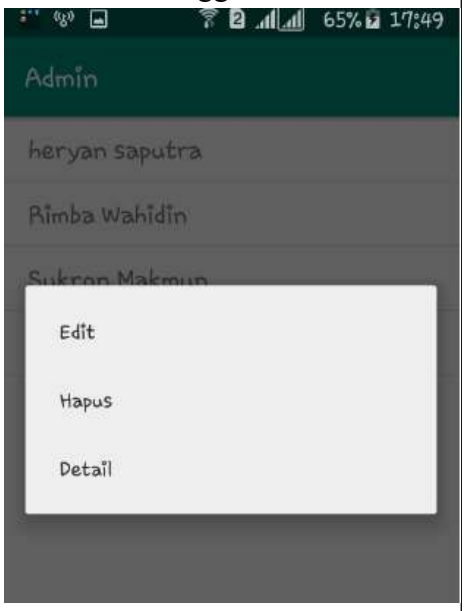
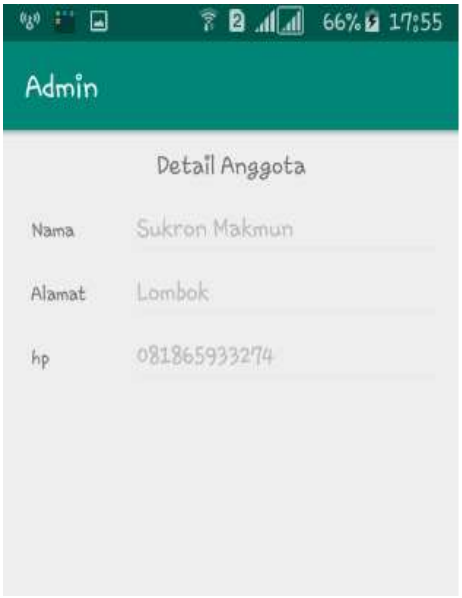
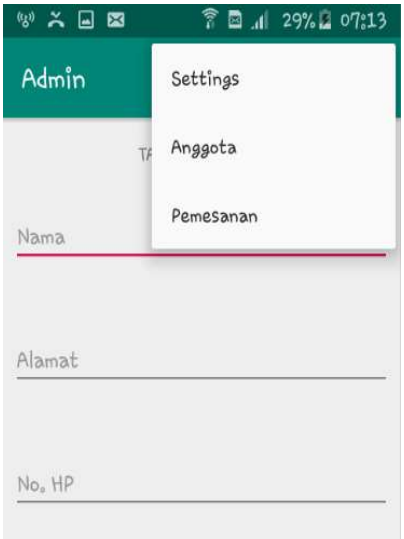
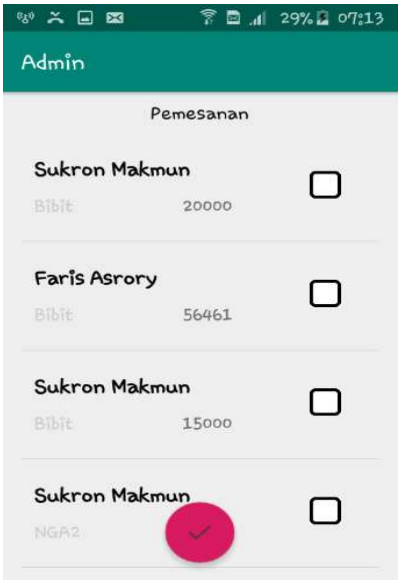
Pengujian aplikasi untuk admin dengan metode Blackbox dapat dilihat pada tabel 4.2 berikut :

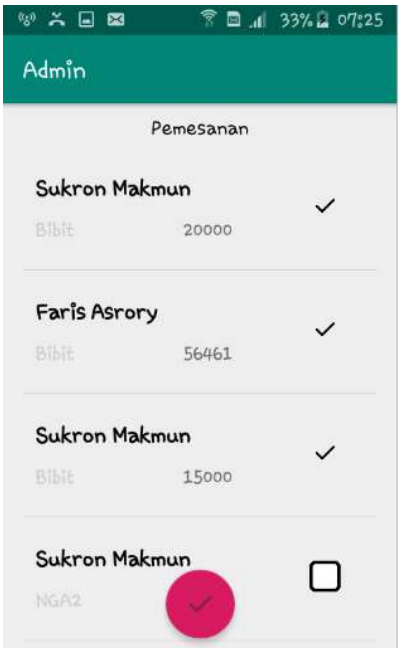
Tabel 4.2. pengujian aplikasi admin dengan metode Blackbox

No	Skenario	Hasil yang diharapkan	kesimpulan
1	Menekan tombol “Massukkan” pada menu tambah anggota tanpa mengisi data 	Sistem tidak akan mengirim data ke Firebase dan menampilkan pesan “tidak boleh ada yang kosong 	Valid
2	Memasukan data anggota yang akan ditambah dan menekan tombol “Masukkan 	Sistem akan menyimpan data anggota dan menampilkan pesan “berhasil disimpan” 	Valid

No	Skenario	Hasil yang diharapkan	kesimpulan
3	Memilih nggota pada menu bar tambah anggota 	Sistem akan beralih ke menu anggota terdaftar 	Valid
4	Menekan lama pada salah satu anggota pada menu anggota terdaftar 	Sistem akan menampilkan piliha untuk edit, hapus, dan detail anggota 	Valid

No	Skenario	Hasil yang diharapkan	kesimpulan
5	Memilih edit anggota 	Sistem akan menampilkan menu untuk edit anggota 	valid
6	Memilih hapus anggota 	Sistem akan menghapus anggota terdaftar 	Valid

No	Skenario	Hasil yang diharapkan	kesimpulan
7	Memilih detail untuk melihat data anggota 	Sistem akan menampilkan detail dari anggota yang dipilih 	Valid
8	Memilih pemesanan pada menu Admin 	Sistem akan menampilkan pesanan masuk dari mitra 	valid

No	Skenario	Hasil yang diharapkan	kesimpulan
9	Menekan lama pada salah satu pesanan masuk 	Sistem akan menampilkan pilihan untuk melihat detai pesanan 	valid
10	Mengkonfirmasi beberapa pesanan masuk 	Sistem akan mengkonfirmasi pesanan yang dipilih dan menyisakan pesanan yang belum di konfirmasi 	valid

4.2.2. Pengujian Metode

Pengujian metode ini dilakukan untuk mengetahui keakuratan hasil diagnosa dari sistem yang dibangun. Pengujian dilakukan dengan membandingkan hasil dari sistem dan perhitungan manual berdasarkan data gejala yang dimiliki. Data nilai *belief(bel)* gejala terhadap penyakit dapat dilihat pada tabel 4.2.2a. Tabel nilai *bel* gejala terhadap penyakit berikut :

Tabel 4.1. Tabel nilai *bel* gejala terhadap penyakit.

Kode	Nama Gejala	p 1	p 2	p 3	p 4	p 5	p 6	p 7	p 8	Belief(bel)
G1	Nafsu makan menurun	√			√	√				0,3
G2	Tubuh ikan melemah	√					√		√	0,3
G3	Ikan kurus	√								0,3
G4	Ikan gelisah (*Nervous*)	√						√		0,6
G5	Pergerakan melamban	√			√					0,6
G6	Pertumbuhan ikan melambat		√							0,6
G7	Proses ganti kulit terhambat		√							0,6
G8	Timbul peradangan di kulit		√			√				0,6
G9	Sirip rusak, menguncup atau rontok			√						0,6
G10	Sirip, insang dan kulit rusak				√					0,7
G11	Kulit menjadi merah				√					0,6
G12	Sisik lepas				√					0,6
G13	Tubuh kotor dan berwarna gelap				√	√				0,6
G14	Warna gelap di bagian rahang					√				0,85
G15	warna tubuh pucat					√				0,4
G16	Insang terinfeksi dan berwarna kemerahan						√			0,85

G17	Insang pucat dan bengkak						√			0,75
G18	sering meloncat-loncat						√			0,9
G19	Berenang ke permukaan air							√		0,9
G20	Ikan berkumpul dekat saluran pembuangan							√		0,75
G21	Berkumpul atau mendekat ke air masuk							√		0,6
G22	Kulit kasat								√	0,6
G23	Lendir berlebih								√	0,6
G24	Pendarahan di sekitar sirip, ekor dan sekitar anus serta bagian tubuh lainnya	√							√	0,7
G25	Luka di sekitar mulut dan bagian tubuh lainnya								√	0,85
G26	Luka yang menjadi borok						√			0,9
G27	Perut lembek dan bengkak yang berisi cairan merah kekuningan						√			0,35
G28	perut ikan gembung						√			0,6
G29	Mata menonjol				√					0,75
G30	Pergerakan tidak terarah					√				0,95
G31	Menggosokkan badan pada benda di sekitarnya					√				0,95
G32	Bernafas mengap-mengap ke permukaan air						√			0,45
G33	Terdapat bercak-bercak merah di bagian luar tubuh ikan							√		0,8
G34	terlihat benang benang halus menyerupai kapas di bagian tubuh ikan			√						0,9

Tabel 4.2. Keterangan penyakit

No.	Kode Penyakit	Nama Penyakit
1	P1	Trichodina
2	P2	Epistylis
3	P3	Achlya
4	P4	Aeromona Hydrophilia
5	P5	Pseudomonas
6	P6	Streptococcosis
7	P7	Dactylogyriasis
8	P8	Gyrodactyliasis

Pengujian hasil diagnosa sistem yang dibandingkan dengan hasil perhitungan manual dengan memilih beberapa gejala sebagai yang akan dihitung oleh sistem dapat dilihat pada tabel 4.5 pengujian hasil diagnosa sistem berikut :

Tabel 4.3. Pengujian hasil diagnosa penyakit

No.	Gejala yang dipilih	Hasil diagnosa sistem	Hasil perhitungan manual	keterangan
1	Sisik lepas, Tubuh kotor dan berwarna gelap, Warna gelap dibagian rahang.	Penyakit = Pseudomonas Dengan persentase 58,62068965517242 %	Penyakit = Pseudomonas Dengan persentase 58,62068966 %	99 % akurat
2	Luka disekitar mulut, Lendir berlebih, Mata menonjol	Penyakit = Gyrodacctyliasis dengan persentase 50,74626865671643 %	Penyakit = Gyrodactyliasis dengan persentase 50, 74626875 %	99 % akurat

BAB V

PENUTUP

5.1. Kesimpulan

Kesimpulan yang bisa diambil dari penelitian ini sebagai berikut:

1. Aplikasi yang diangun sebagai sarana pemesanan kebutuhan selama proses budidaya ikan nila di CV. Naldy Putra.
2. Didalam aplikasi dapat dilakukan diagnosa penyakit ikan nila dengan memasukkan gejala-gejala yang dialami ikan.
3. Sistem yang dibangun dapat mempermudah CV. Naldy Putra dalam menyimpan data - data pesanan dari mitra kerja.

5.2. Saran

Saran yang dapat dijadikan acuan untuk pengembangan aplikasi ini kedepannya sebagai berikut:

1. Kedepannya user dapat melihat riwayat pesanan sebelumnya
2. Admin dapat mengolah data-data pesanan untuk menghitung biaya yang dihabiskan selama budidaya.

DAFTAR PUSTAKA

1. Dewi, N.K.C., Anandita, I.B.G., Atmaja, K.J. and Aditama, P.W., 2018. RANCANG BANGUN APLIKASI MOBILE SISKA BERBASIS ANDROID. *SINTECH (Science and Information Technology) Journal*, *1*(2), pp.100-107.
2. Putera, J.M., RANCANG BANGUN APLIKASI BERBASIS ANDROID DENGAN PENERAPAN WEB SERVICE PADA SISTEM INFORMASI PERPUSTAKAAN (Studi Kasus: Perpustakaan Daerah Kalimantan Barat). *Jurnal Sistem dan Teknologi Informasi (JustIN)*, *5*(1), pp.47-51.
3. Sede, D.W., Sinsuw, A.A. and Najoan, X.B., 2015. Rancang Bangun Aplikasi Pemesanan Tiket Online Kapal Laut Berbasis Android. *JURNAL TEKNIK INFORMATIKA UNIVERSITAS SAM RATULANGI*, *6*(1).
4. Sulistyohati A, Hidayat T. Aplikasi sistem pakar diagnosa penyakit ginjal dengan metode DEMPSTER-SHAFER. InSeminar Nasional Aplikasi Teknologi Informasi (SNATI) 2008 (Vol. 1, No. 1).
5. Nugraha, D. and Winiarti, S., 2013. Pengembangan media pembelajaran sistem pelacakan pada mata kuliah kecerdasan buatan berbasis multimedia. *JSTIE (Jurnal Sarjana Teknik Informatika)(E-Journal)*, *2*(1), pp.67-77.
6. Yusnita E, Aprilianto H. Sistem Pakar Diagnosa Penyakit Ikan Nila Menggunakan Dempster Shafer Berbasis Web. JUTISI. 2016 Oct 26;4(2).
7. Putra, I., Setiyanto, D.D. and Wahyjuningrum, D., 2011. Pertumbuhan dan kelangsungan hidup ikan nila Oreochromis niloticus dalam sistem resirkulasi. *Jurnal Perikanan dan Kelautan*, *16*(01).

LAMPIRAN

Source Code Aplikasu User

```
// Menu Login
package com.example.lenovo.app;
import android.app.ProgressDialog;
import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class Login extends AppCompatActivity {

   private EditText nama, password;
   private Button masuk;
   private TextView daftar;
   private FirebaseAuth firebaseAuth;
   private ProgressDialog progressDialog;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_login);

      nama = findViewById(R.id.etnama);
      password = findViewById(R.id.etpassword);
      masuk = findViewById(R.id.btnmasuk);
      daftar = findViewById(R.id.tvdaftar);
      progressDialog = new ProgressDialog(this);

      firebaseAuth = FirebaseAuth.getInstance();
      FirebaseUser user = firebaseAuth.getCurrentUser();
      if (user != null){
         finish();
         startActivity(new Intent(Login.this, ListAnggota.class));
      }
```

```
      masuk.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            if (nama.length()==0 || password.length()==0){
               Toast.makeText(Login.this, "Masukkan Email dan Password",
Toast.LENGTH_LONG).show();
            }else{
            validate(nama.getText().toString(), password.getText().toString());
            }
         }
      });
      daftar.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            startActivity(new Intent(Login.this, DaftarUser.class));
         }
      });
   }

   private void validate (String userName, String userPassword){
      progressDialog.setMessage("Mohon Tunggu");
      progressDialog.show();
      firebaseAuth.signInWithEmailAndPassword(userName,
userPassword).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
         @Override
         public void onComplete(@NonNull Task<AuthResult> task) {
            progressDialog.dismiss();
            if (task.isSuccessful()){
               Toast.makeText(Login.this, "Login Berhasil",Toast.LENGTH_LONG).show();
               startActivity(new Intent(Login.this, ListAnggota.class));
            }else {
               Toast.makeText(Login.this, "Login Gagal",Toast.LENGTH_LONG).show();
               progressDialog.dismiss();
            }
         }
      });
   }
}

// Menu Daftar User
package com.example.lenovo.app;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
```

```
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class DaftarUser extends AppCompatActivity {

   private EditText daftaruser, daftarpassword, daftaremail;
   private Button Daftarkan;
   private TextView menulogin;
   private FirebaseAuth firebaseAuth;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_daftar_user);
      Daftar();
      firebaseAuth = FirebaseAuth.getInstance();
      Daftarkan.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            if (validasi()){
               //upload data ke database
               String user_email = daftaremail.getText().toString().trim();
               String user_password = daftarpassword.getText().toString().trim();

               firebaseAuth.createUserWithEmailAndPassword(user_email,
user_password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
                  @Override
                  public void onComplete(@NonNull Task<AuthResult> task) {
                     if (task.isSuccessful()){
                        Toast.makeText(DaftarUser.this, "Daftar Berhail",
Toast.LENGTH_LONG).show();
                        startActivity(new Intent(DaftarUser.this, Login.class));
                     }else {
                        Toast.makeText(DaftarUser.this, "Daftar Gagal",
Toast.LENGTH_LONG).show();
                     }
                  }
               });
            }
         }
```

```
        });

        menulogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(DaftarUser.this, Login.class));
            }
        });
    }
    private void Daftar(){
        daftaruser = findViewById(R.id.etuserdaftar);
        daftarpassword = findViewById(R.id.etpassworddaftar);
        daftaremail = findViewById(R.id.etemaildaftar);
        Daftarkan = findViewById(R.id.btndaftaruser);
        menulogin = findViewById(R.id.tvmenulogin);
    }
    private boolean validasi(){
        boolean hasil = false;
        String nama = daftaruser.getText().toString();
        String pass = daftarpassword.getText().toString();
        String eml = daftaremail.getText().toString();

        if (nama.isEmpty() || pass.isEmpty() || eml.isEmpty()){
            Toast.makeText(DaftarUser.this, "Masukkan Data", Toast.LENGTH_LONG).show();
        }else {
            hasil = true;
        }
        return hasil;
    }
}
//Menu List Anggota
public class ListAnggota extends AppCompatActivity {

    //Firebase Variable
    Anggota anggota;
    private DatabaseReference databaseReference;
    private DatabaseReference mref;
    private ValueEventListener mEvent;
    //Android layout
    private Button btTambah;
    private ListView lvAnggota;
    private EditText etTambah;
    private Context context;
    //Array list
    private String nama,alamat,hp,key;
```

```
  public void cleaner()
  {
    mref.removeEventListener(mEvent);mref.keepSynced(false);
  }

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_list_anggota);
    context=this;
    databaseReference=FirebaseDatabase.getInstance().getReference();
    mref=databaseReference.child("Anggota");
    mEvent=new ValueEventListener(){
      List<Anggota> array;
      @Override
      public void onDataChange(@NonNull final DataSnapshot dataSnapshot) {
        array = new ArrayList<>();
        for (DataSnapshot j: dataSnapshot.getChildren())
        {
          nama=j.getValue(Anggota.class).getNama();
          alamat=j.getValue(Anggota.class).getAlamat();
          hp=j.getValue(Anggota.class).getHp();
          key=j.getKey();
          array.add(new Anggota(nama,alamat,hp,key));
        }
        lvAnggota=(ListView)findViewById(R.id.lvanggota);
        adapteranggota adapter= new adapteranggota(context,array);
        lvAnggota.setAdapter(adapter);
        lvAnggota.setOnItemClickListener(new AdapterView.OnItemClickListener() {
          @Override
          public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
            Intent next=new Intent(ListAnggota.this, pemesanan.class);
            next.putExtra("key",array.get(position).getKey());
            startActivity(next);
          }
        });
      }
      @Override
      public void onCancelled(@NonNull DatabaseError databaseError) {

      }
    };
    mref.addValueEventListener(mEvent);
  }
  //menu list anggota
```

```
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.listanggotamenu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()){
            case R.id.logout:
                FirebaseAuth.getInstance().signOut();
                startActivity(new Intent(ListAnggota.this, Login.class));
                Toast.makeText(this, "Berhasil Logout",Toast.LENGTH_SHORT).show();
                return true;
            case R.id.diagnosa:
                Toast.makeText(this, "Diagnosa",Toast.LENGTH_SHORT).show();
                startActivity(new Intent(ListAnggota.this, Diagnosa.class));
                Toast.makeText(this, "Diagnosa",Toast.LENGTH_SHORT).show();
                return true;
            case R.id.tentang:
                Toast.makeText(this, "Tentang",Toast.LENGTH_SHORT).show();
                return true;
            case R.id.app:
                Toast.makeText(this, "Tentang Aplikasi",Toast.LENGTH_SHORT).show();
                return true;
            case R.id.developer:
                Toast.makeText(this, "Tentang Developer",Toast.LENGTH_SHORT).show();
                return true;
            default: return super.onOptionsItemSelected(item);
        }
    }
    @Override
    protected void onStart() {
        super.onStart();
        mref.keepSynced(true);
    }
    @Override
    protected void onStop() {
        super.onStop();
        cleaner();
    }
}
//Menu Pemesanan
public class pemesanan extends AppCompatActivity {
```

```
    private EditText jbibit,jpakan,jobat;
    private Spinner pakan,obat;
    private Button oke;

    public void setJpakan(EditText jpakan) {
        this.jpakan = jpakan;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pemesanan);
        jbibit = findViewById(R.id.jbibit);
        jpakan= findViewById(R.id.jpakan);
        jobat= findViewById(R.id.jobat);
        pakan= findViewById(R.id.pakan);
        obat= findViewById(R.id.obat);
        oke= findViewById(R.id.pesan);
        oke.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                TextView tes;
                String jenis,jumlah;
                tes=findViewById(R.id.tvbibit);
                Bundle a = getIntent().getExtras();
                FirebaseDatabase database= FirebaseDatabase.getInstance();
                DatabaseReference dbAnggota=database.getReference("pemesanan");
                //bibit
                if (jbibit.getText().length()!=0)
                {
                    dbAnggota.push().setValue(new
datapesanan(a.getString("key").toString(),"Bibit",jbibit.getText().toString(),"0"));
                    Toast.makeText(pemesanan.this, "jumlah bibit=
"+jbibit.getText().toString(),Toast.LENGTH_SHORT).show();
                    Log.e("input di eksekusi =","jumlah bibit= "+jbibit.getText().toString());
                }
                //
                //pakan
                if(periksa(pakan,jpakan)) {
                    dbAnggota.push().setValue(new
datapesanan(a.getString("key").toString(),pakan.getSelectedItem().toString(),jpakan.getT
ext().toString(),"0"));
                    Toast.makeText(pemesanan.this, "jumlah pakan=
"+jpakan.getText().toString()+"  *jenis pakan
="+pakan.getSelectedItem().toString(),Toast.LENGTH_SHORT).show();
                    Log.e("input di eksekusi =","jumlah pakan= "+jpakan.getText().toString()+"
```

```
*jenis pakan ="+pakan.getSelectedItem().toString());
            }
            //obat
            if (periksa(obat,jobat)) {
               dbAnggota.push().setValue(new
datapesanan(a.getString("key").toString(),obat.getSelectedItem().toString(),jobat.getText(
).toString(),"0"));
               Toast.makeText(pemesanan.this, "jumlah obat= "+jobat.getText().toString()+"
*jenis pakan ="+obat.getSelectedItem().toString(),Toast.LENGTH_SHORT).show();
               Log.e("input di eksekusi =","jumlah obat= "+jobat.getText().toString()+"  *jenis
pakan ="+obat.getSelectedItem().toString());
            }
            jbibit.setText("");
            jobat.setText("");
            jpakan.setText("");
         }
      });
   }
   public boolean periksa(Spinner spinner,EditText editText){
      return (spinner.getSelectedItemPosition()!=0&&editText.getText().length()!=0);
   }
}

//Menu Diagnosa
package com.example.lenovo.app;

import android.content.Intent;
import android.os.Build;
import android.support.annotation.RequiresApi;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import java.math.BigDecimal;
import java.math.RoundingMode;
import java.util.ArrayList;
import java.util.List;

public class Diagnosa extends AppCompatActivity
{
```

```
    private Button diagnosa;
    private ArrayAdapter<String>arrayAdapter;
    private  ListView listView;
    public String hasil;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_diagnosa);
        diagnosa = findViewById(R.id.btndiagnosa);
        final ListView listView = findViewById(R.id.lvgejala);
        final List<UserModel> users = new ArrayList<>();
        users.add(new UserModel(false,"Nafsu Makan Menurun"));
        users.add(new UserModel(false,"Tubuh Ikan Melemah"));
        users.add(new UserModel(false,"Ikan Kurus"));
        users.add(new UserModel(false,"Ikan Gelisah (Nervous)"));
        users.add(new UserModel(false,"Pergerakan Melamban"));
        users.add(new UserModel(false,"Pertumbuhan Ikan Melambat"));
        users.add(new UserModel(false,"Proses Ganti Kulit Terhambat"));
        users.add(new UserModel(false,"Timbul Peradangan Pada Kulit"));
        users.add(new UserModel(false,"Sirip Rusak, Menguncup Atau Rontok"));
        users.add(new UserModel(false,"Sirip, Insang dan kulit Rusak"));
        users.add(new UserModel(false,"Kulit Menjadi Merah"));
        users.add(new UserModel(false,"Sisik Lepas"));
        users.add(new UserModel(false,"Tubuh Kotor dan Berwarna Gelap"));
        users.add(new UserModel(false,"Warna Gelap di Bagian Rahang"));
        users.add(new UserModel(false,"Warna Tubuh Pucat"));
        users.add(new UserModel(false,"Insang terinfeksi dan berwarna kemerahan"));
        users.add(new UserModel(false,"Insang pucat dan bengkak"));
        users.add(new UserModel(false,"Sering meloncat-loncat"));
        users.add(new UserModel(false,"Berenang ke permukaan air"));
        users.add(new UserModel(false,"Ikan berkumpul ke dekat saluran pembuangan"));
        users.add(new UserModel(false,"Berkumpul/Mendekat ke Air Masuk"));
        users.add(new UserModel(false,"Kulit kasat"));
        users.add(new UserModel(false,"Lendir berlebihh"));
        users.add(new UserModel(false,"Pendarahan di sekitar sirip, ekor dan anus"));
        users.add(new UserModel(false,"Luka di sekitar mulut dan bagian lainnya"));
        users.add(new UserModel(false,"Luka yang menjadi borok"));
        users.add(new UserModel(false,"Perut lembek dan bengkak berisi cairan merah
kekuningan"));
        users.add(new UserModel(false,"Perut ikan Gembung"));
        users.add(new UserModel(false,"Mata menonjol"));
        users.add(new UserModel(false,"Pergerakan tidak terarah"));
        users.add(new UserModel(false,"Menggosokkan badan pada benda di
sekitarnya(gatal)"));
        users.add(new UserModel(false,"Bernafas mengap-mengap ke permukaan air"));
        users.add(new UserModel(false,"Terdapat bercak-bercak merah di bagian luar tubuh
```

```
ikan"));
      users.add(new UserModel(false,"Terlihat benang-benang halus menyerupai kapas di
bagian tubuh"));

      final CustomAdapter adapter = new CustomAdapter(this, users);
      listView.setAdapter(adapter);
      diagnosa.setOnClickListener(new View.OnClickListener() {
         @Override


         @RequiresApi(api = Build.VERSION_CODES.N)

         public void onClick(View v) {
            Dempter h = new Dempter(users);
            int count;
            count=0;
               for(UserModel i : users)
               {
                  if(i.isSelected)
                  {
                     count++;
                  }
               }
               if(count>=2)
               {

                  Intent next=new Intent(Diagnosa.this, HasilDiagnosa.class);
                  next.putExtra("Penyakit",h.getDiagnosapenyakit());
                  next.putExtra("kalkulasi",h.persentase());
                  next.putExtra("penanganan",h.getPenangananterpilh());

                  startActivity(next);
               }
               else
               {
                  Toast.makeText(Diagnosa.this,"Minimal Gejala yang dimasuukkan adalah
2",Toast.LENGTH_LONG).show();
               }
         }
      });
      listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
         @Override
         public void onItemClick(AdapterView<?> parent, View view, int i, long l) {
            UserModel model = users.get(i);
            if (model.isSelected())
               model.setSelected(false);
```

```
            else
               model.setSelected(true);
            users.set(i, model);
            adapter.updateRecords(users);
         }
      });
   }
}

//Hasil Diagnosa
package com.example.lenovo.app;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.method.ScrollingMovementMethod;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

public class HasilDiagnosa extends AppCompatActivity {
   private TextView penyakit,kalkulsi,penanganan;
   Penanganan x;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      Bundle a = getIntent().getExtras();
      Double hasil;
      x = new Penanganan();
      setContentView(R.layout.activity_hasil_diagnosa);
      penyakit =findViewById(R.id.tvjenispenyakit);
      kalkulsi=findViewById(R.id.tvpersentase);
      penanganan = findViewById(R.id.viewpenanganan);
      this.penanganan.setMovementMethod(new ScrollingMovementMethod());
      hasil=a.getDouble("kalkulasi");
      this.penyakit.setText(a.getString("Penyakit"));
      this.kalkulsi.setText(String.valueOf(hasil*100)+"%");
      this.penanganan.setText(a.getString("penanganan"));

   }
}

//Perhitungan Dempster-Shafer
package com.example.lenovo.app;

import android.content.res.Resources;
```

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Dempter extends Diagnosa {


   public double hasil[] = new double[34];
   public double bel[]= new double[34];
   public List<UserModel> setusers = new ArrayList<>();
   private double teta;
   public int gejalaterpilih;
   public String penyakit[] = new String[]{"Trichodina","Epistylis","Achlya","Aeromona
Hydrophilia","Pseudomonas","Streptococcosis","Dactylogyriasis","Gyrodactyliasis"};
   public double believe[] = new
double[]{0.3,0.3,0.3,0.6,0.6,0.6,0.6,0.6,0.6,0.7,0.6,0.6,0.6,0.85,0.4,0.85,0.75,0.9,0.9,0.75,
0.6,0.6,0.6,0.7,0.85,0.9,0.35,0.6,0.75,0.95,0.95,0.45,0.8,0.9};
   public int rule[][] = new
int[][]{{1,4,5},{1,6,8},{1,0,0},{1,7,0},{1,4,0},{2,0,0},{2,0,0},{2,5,0},{3,0,0},{4,0,0},{4,0,0},{4,0,
0},{4,5,0},{5,0,0},{5,0,0},{6,0,0},{6,0,0},{6,0,0},{7,0,0},{7,0,0},{7,0,0},{8,0,0},{8,0,0},{1,8,0},{
8,0,0},{6,0,0},{6,0,0},{6,0,0},{4,0,0},{5,0,0},{5,0,0},{6,0,0},{7,0,0},{3,0,0}};
   private Integer imax;
   private String Penangananterpilh;
 // private String penanganan1;

   public int ipenyakit;

   public Integer getImax() {
      return imax;
   }

   public Dempter(List<UserModel> setusers) {
      this.setusers = setusers;
   }
public void rumus()
{
   Integer k=0;
   int hold[] = new int[34];
   int l=0;
   for(UserModel i:setusers)
   {
      if(i.isSelected)
      {
         bel[k]=believe[l];
         hold[k]=l;
         k++;
```

```
      }
      l++;
   }
   gejalaterpilih=k;
   dst();
   Integer max=0;
   Integer temp=0;
   for(double cek: bel)
   {
      if(hasil[max]<hasil[temp])
      {
         max=temp;
      }
      temp++;
   }
   ipenyakit=hold[max];
   imax=max;
}

  // public String getPenanganan1() {
 //     return penanganan1;
  // }

   public double persentase()
   {
      rumus();
      return bel[imax];
   }

   public String getPenangananterpilh() {
      return Penangananterpilh;
   }

   public String getDiagnosapenyakit() {
      rumus();
//       Resources res= getResources();
//       String[] penanganan=getResources().getStringArray(R.array.Penangananisi);
      Penanganan x=new Penanganan();
      Penangananterpilh="";
      String j="";
      int status=0;
      for (int a=0;a<3;a++)
      {
         int b;
         b=rule[ipenyakit][a];
         if(b>0)
```

```
        {
          if(status>0)
          {
            j+=", ";
            Penangananterpilh+="\n\n----------\n\n";
  //          penanganan1+="/n_____/n";
          }
          j+=penyakit[b-1];
          Penangananterpilh+=x.getPengananan()[b-1];
//          penanganan1+=penanganan[b-1];
          status++;
        }
      }
      return j;
  }
  private double env(double o, Integer a){
      double x = 0;
      for (int i = 0; i<=a; i++){
        x=x+(bel[i]*o);
      }
      return 1-x;
  }

  public void dst (){
        for (int i = 0; i <= gejalaterpilih - 2; i++) {
          if (i == 0) {
            teta=1-bel[i];
          }
            for (int j = 0; j <= i; j++)
            {
              hasil[j] = (bel[j] * (1 - bel[i + 1])) / env(bel[i + 1], i);
            }
            hasil[i + 1] = (teta * (bel[i + 1])) / env(bel[i + 1], i);
            teta = (teta * (1 - bel[i + 1])) / env(bel[i + 1], i);

            for (int k = 0; k <= i + 1; k++) {
              bel[k] = hasil[k];
            }
        }
  }
}
```

Source Code Aplikasi Admin

```
//Menu Tambah Anggota
package com.example.lenovo.admin;

import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class MainActivity extends AppCompatActivity {

   private EditText nama,alamat,hp;
   private Button insert;
   private TextView tes;
   FirebaseDatabase database;
   DatabaseReference dbAnggota,dbIndex;
   Anggota anggota;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
      tes=(TextView) findViewById(R.id.tes);
      setSupportActionBar(toolbar);

      nama = findViewById(R.id.etnama);
      alamat = findViewById(R.id.etalamat);
      hp = findViewById(R.id.ethp);
      insert = findViewById(R.id.btninput);
```

```
        database = FirebaseDatabase.getInstance();
        dbAnggota = database.getReference("Anggota");
        anggota = new Anggota();
    }
    private void getValue(){
        anggota.setNama(nama.getText().toString());
        anggota.setAlamat(alamat.getText().toString());
        anggota.setHp(hp.getText().toString());
    }
    private void hapus_isi(){
        nama.setText("");
        alamat.setText("");
        hp.setText("");
    }

    private boolean periksa(){
        getValue();
        if (nama.getText().toString().equals("") || alamat.getText().toString().equals("") ||
hp.getText().toString().equals("")){
            return false;
        }
        else
        {
            return true;
        }
    }
//button insert into database firebase
    public void btninsert (View view){
        //update anggota
        if(periksa()) {
            dbAnggota.push().setValue(anggota);
            dbAnggota.addChildEventListener(new ChildEventListener() {
                @Override
                public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable
String s) {
                    Toast.makeText(MainActivity.this,"berhasil di
simpan",Toast.LENGTH_SHORT).show();
                }

                @Override
                public void onChildChanged(@NonNull DataSnapshot dataSnapshot,
@Nullable String s) {

                }

                @Override
```

```
            public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {

            }

            @Override
            public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable
String s) {

            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {
               Toast.makeText(MainActivity.this,"kesalahan jaringan/ kesalahan dalam
proses menyimpan",Toast.LENGTH_SHORT).show();
            }
          });
    }
    else
    {
       Toast.makeText(MainActivity.this,"Tidak boleh ada
kosong",Toast.LENGTH_SHORT).show();
    }
    hapus_isi();
  }

  @Override
  public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
  }

  @Override
  public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId())
    {
       case R.id.setting:
          return true;
       case R.id.pemesanan:
          startActivity(new Intent(MainActivity.this,Pemesanan.class));
          return true;
          default:
             startActivity(new Intent(MainActivity.this, Anggotaterdaftar.class));
             return super.onOptionsItemSelected(item);
    }
```

```
    }
}

//Menu Anggota Terdaftar
package com.example.lenovo.admin;

import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.ContextMenu;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.Toast;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
import java.util.List;

public class Anggotaterdaftar extends AppCompatActivity {

    Anggota anggota;
    private DatabaseReference reference;
    private DatabaseReference databaseReference;
    private DatabaseReference mref;
    private ValueEventListener mEvent;
    private ListView lvanggota;
    //kmponen
    private Context context;
    //array vriable
    private String nama,alamat,hp,key;
    //firebase varibl

    @Override
    public void onCreateContextMenu(ContextMenu menu, View v,
```

```
ContextMenu.ContextMenuInfo menuInfo) {
      getMenuInflater().inflate(R.menu.menuanggota,menu);
   }

   public void setNama(String nama) {
      this.nama = nama;
   }

   public void setAlamat(String alamat) {
      this.alamat = alamat;
   }

   public void setHp(String hp) {
      this.hp = hp;
   }

   public void setKey(String key) {
      this.key = key;
   }

   @Override
   public boolean onContextItemSelected(MenuItem item) {
      Log.e("peruahan setter","nama ="+nama);
      Log.e("peruahan setter","alamat ="+alamat);
      Log.e("peruahan setter","hp ="+hp);
      Log.e("peruahan setter","key ="+key);
      Intent i=new Intent(Anggotaterdaftar.this,Editanggota.class);
      i.putExtra("nama",nama);
      i.putExtra("alamat",alamat);
      i.putExtra("hp",hp);
      i.putExtra("key",key);
      i.putExtra("status",item.getTitle().toString());
      switch (item.getItemId())
      {
         case R.id.edit:
            Log.e("menu item",item.getTitle().toString());
            startActivity(i);
            break;
         case R.id.detail:
            startActivity(i);
            Log.e("menu item",item.getTitle().toString());
            break;
            default:

FirebaseDatabase.getInstance().getReference("Anggota").child(key).setValue(null);
               Log.e("menu item",item.getTitle().toString());
```

```
    }
    Log.e("menu item","idnya adalah = "+item.getItemId());

    return super.onContextItemSelected(item);
  }

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_anggotaterdaftar);
    context=this;
    //arrayc.add(new Anggota("nama","alamat","hp","key"));
    databaseReference=FirebaseDatabase.getInstance().getReference();
    mref=databaseReference.child("Anggota");
    mEvent=new ValueEventListener(){
      List<Anggota> array;
      @Override
      public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        array=new ArrayList<>();
        Log.e("data","looping di mulai");
        for(DataSnapshot j:dataSnapshot.getChildren())
        {
          nama=j.getValue(Anggota.class).getNama();
          alamat=j.getValue(Anggota.class).getAlamat();
          hp=j.getValue(Anggota.class).getHp();
          key=j.getKey();
          array.add(new Anggota(nama,alamat,hp,key));
          Log.e("data",nama+"___"+alamat+"___"+hp+"___"+key);
        }
        Log.e("perubahan setter",nama);

        lvanggota =findViewById(R.id.lvanggota);
        Adapteranggota adapteranggota=new Adapteranggota(context,array);
        lvanggota.setAdapter(adapteranggota);
        //memasukkan menu ke setiap item
        lvanggota.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
          @Override
          public boolean onItemLongClick(AdapterView<?> parent, View view, int
position, long id) {
            setNama(array.get(position).getNama());
            setAlamat(array.get(position).getAlamat());
            setHp(array.get(position).getHp());
            setKey(array.get(position).getKey());
            return false;
          }
```

```
            });
            registerForContextMenu(lvanggota);



//Toast.makeText(context,array.get(0).getAlamat().toString(),Toast.LENGTH_SHORT).sho
w();
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
      };
//
Toast.makeText(context,arrayc.get(0).getAlamat().toString(),Toast.LENGTH_SHORT).show
();
      mref.addValueEventListener(mEvent);

   }

   @Override
   protected void onStart() {
      super.onStart();
      mref.addValueEventListener(mEvent);
      mref.keepSynced(true);
   }

   @Override
   protected void onStop() {
      super.onStop();
      mref.keepSynced(false);
      mref.removeEventListener(mEvent);
   }
}

//Menu Anggota
package com.example.lenovo.admin;

import com.google.firebase.database.Exclude;

public class Anggota {

   private String Nama;
   private String Alamat;
   private String Hp;
```

```
    private String key;
    public Anggota() {
    }

    public Anggota(String nama, String alamat, String hp) {
        Nama = nama;
        Alamat = alamat;
        Hp = hp;
    }

    public Anggota(String nama, String alamat, String hp, String key) {
        Nama = nama;
        Alamat = alamat;
        Hp = hp;
        this.key = key;
    }
@Exclude

    public String getKey() {
        return key;
    }

    public String getNama() {
        return Nama;
    }

    public void setNama(String nama) {
        Nama = nama;
    }

    public String getAlamat() {
        return Alamat;
    }

    public void setAlamat(String alamat) {
        Alamat = alamat;
    }

    public String getHp() {
        return Hp;
    }

    public void setHp(String hp) {
        Hp = hp;
    }
}
```

```
//Menu Pemesanan
package com.example.lenovo.admin;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.design.widget.FloatingActionButton;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.ContextMenu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AbsListView;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.Toast;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
import java.util.EventListener;
import java.util.List;

public class Pemesanan extends AppCompatActivity {

    private Activity contextt;
    private ListView lvpesanan;
    private String key;
    private Boolean status;
    private DatabaseReference Db,Dbanggota,Db1,Dbanggota1,konf;
    private ValueEventListener Mevent;
    private List<transaksi> transaksiList;
    private List<Anggota> anggotaList;
    private List<pesantampil> pesantampilList;
    private int position;
    private FloatingActionButton konfirmasi;

    public void setPosition(int position) {
```

```
        this.position = position;
    }
    public void setTransaksiList(transaksi transaksiList) {
        this.transaksiList.add(transaksiList);
    }
    public void setAnggotaList(Anggota anggotaList) {
        this.anggotaList.add(anggotaList);
    }
    public void setPesantampilList(pesantampil pesantampilList) {
        this.pesantampilList.add(pesantampilList);
    }
    public void clear()
    {
        this.transaksiList.clear();
        this.anggotaList.clear();
        this.pesantampilList.clear();
    }
    public Boolean getStatus() {
        return status;
    }
    public void setKey(String key) {
        this.key = key;
    }
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
        getMenuInflater().inflate(R.menu.menu_pemesanan,menu);
    }
    @Override
    public boolean onContextItemSelected(MenuItem item) {
        Toast.makeText(contextt,"yang di pilih adalah =
"+transaksiList.get(position).getItem()+" sejumlah
"+transaksiList.get(position).getJumlah(),Toast.LENGTH_SHORT).show();
        setKey(transaksiList.get(position).getKey());
        Intent i=new Intent(Pemesanan.this,Editanggota.class);
        i.putExtra("nama",anggotaList.get(position).getNama());
        //alamat ini sebenarnya nama item dari pesanan hanya saja supaya mempermudah
pengolahan data diclass Editanggota.class maka di ganti dengan alamat karena edittext
yang akan di gunakan adalah exittext milik alamat
        i.putExtra("alamat",transaksiList.get(position).getItem().toUpperCase());
        //hp ini sebenarnya jumlah item yang di pesan hanya saja supaya mempermudah
pengolahan data di class Editanggota.class maka di ganti dengan hp karena edittext yang
akan di gunakan adalah exittext milik hp
        i.putExtra("hp",transaksiList.get(position).getJumlah());
        i.putExtra("status","pemesanan");
        startActivity(i);
```

```
    return super.onContextItemSelected(item);
  }
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_pemesanan);
    contextt=this;
    status=false;
    Db1=FirebaseDatabase.getInstance().getReference();
    Db=Db1.child("pemesanan");
    Dbanggota1=FirebaseDatabase.getInstance().getReference();
    Dbanggota=Dbanggota1.child("Anggota");
    Mevent= new ValueEventListener() {
      @Override
      public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        Log.e("array","deklarasi mulai");
        pesantampilList=new ArrayList<>();
        anggotaList=new ArrayList<>();
        transaksiList=new ArrayList<>();
        clear();
        Log.e("array","deklarasi berakhir");
        for(final DataSnapshot a:dataSnapshot.getChildren())
        {
          final String id;
          id=a.getValue(transaksi.class).getIdanggota();
          Dbanggota.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
              for (DataSnapshot j:dataSnapshot.getChildren()) {
                if (id.equals(j.getKey())&&
a.getValue(transaksi.class).getStatus().equals("0")) {
                  //Toast.makeText(context, j.getValue(Anggota.class).getNama(),
Toast.LENGTH_SHORT).show();
                  setTransaksiList(new
transaksi(a.getValue(transaksi.class).getIdanggota(),a.getValue(transaksi.class).getItem().t
oString(),a.getValue(transaksi.class).getJumlah(),a.getValue(transaksi.class).getStatus(),a.
getKey()));
                  Log.e("benar","key ="+j.getValue(Anggota.class).getNama());
                  setAnggotaList(new
Anggota(j.getValue(Anggota.class).getNama(),j.getValue(Anggota.class).getAlamat(),j.get
Value(Anggota.class).getHp(),j.getKey()));
                  setPesantampilList(new
pesantampil(j.getValue(Anggota.class).getNama(),a.getValue(transaksi.class).getItem(),a.
getValue(transaksi.class).getStatus(),a.getValue(transaksi.class).getJumlah()));
                }
```

```
                    }
                    Log.e("pemesanan","array ="+anggotaList.size());
                    Log.e("pemesanan","memulai pemasangan adapter");
                    lvpesanan=findViewById(R.id.lvpesanan);
                    lvpesanan.setAdapter(null);
                    final adapterpemesanan adapterpemesanan1=new
adapterpemesanan(contextt,pesantampilList);
                    lvpesanan.setAdapter(adapterpemesanan1);
                    lvpesanan.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
                        public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
                            pesantampil a=pesantampilList.get(position);
                            if (a.getStatus().equals("0")) {
                               a.setStatus("1");
                            }else a.setStatus("0");
                            pesantampilList.set(position,a);
                            adapterpemesanan1.updateRecords(pesantampilList);
                        }
                    });
                    lvpesanan.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
                        @Override
                        public boolean onItemLongClick(AdapterView<?> parent, View view, int
position, long id) {
                            setPosition(position);
                            return false;
                        }
                    });
                    registerForContextMenu(lvpesanan);
                    Log.e("pemesanan","menghentikan pemasangan adapter");
                }
                @Override
                public void onCancelled(@NonNull DatabaseError databaseError) {

                }
            });
        }
    }

    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {

    }
};
konfirmasi=findViewById(R.id.konfirmasi);
```

```
    konf=FirebaseDatabase.getInstance().getReference().child("pemesanan");
    konfirmasi.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            for (int a = 0; a< pesantampilList.size(); a++)
            {
                if (pesantampilList.get(a).getStatus().equals("1"))
                {
                    konf.child(transaksiList.get(a).getKey()).child("status").setValue("1");
                }
            }
        }
    });

    //Log.e("Pemesanan","nilai arr ="+arr.size());
    /*adapterpemesanan adapterpemesanan1= new adapterpemesanan(this,arr);
    listView=findViewById(R.id.lvpesanan);
    listView.setAdapter(adapterpemesanan1);
    */
}

@Override
protected void onStart() {
    super.onStart();
    Log.e("Onstart","eksekusi");
    Db1.child("pemesanan").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            Db.removeEventListener(Mevent);
            Db.addValueEventListener(Mevent);
            Db.keepSynced(true);
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }

    });

}

@Override
protected void onPause() {
    super.onPause();
    Db.keepSynced(false);
```

```
        Db.removeEventListener(Mevent);
    }

    @Override
    protected void onStop() {
        super.onStop();
        Log.e("Onstop","Eksekusi");
        Db.keepSynced(false);
        Db.removeEventListener(Mevent);
    }
}

//Pesan Tampil
package com.example.lenovo.admin;

import com.google.firebase.database.Exclude;

public class pesantampil {
    private String nama1,keterangan,status,jumlah,key;

    public pesantampil() {
    }
    public pesantampil(String nama1, String keterangan, String status, String jumlah) {
        this.nama1 = nama1;
        this.keterangan = keterangan;
        this.status = status;
        this.jumlah = jumlah;
    }
    public void setJumlah(String jumlah) {
        this.jumlah = jumlah;
    }
    @Exclude
    public String getJumlah() {
        return jumlah;
    }
    public void setNama1(String nama1) {
        this.nama1 = nama1;
    }
    public void setKeterangan(String keterangan) {
        this.keterangan = keterangan;
    }
    public String getKeterangan() {
        return keterangan;
    }
    public void setStatus(String status) {
        this.status = status;
```

```
    }
    public String getStatus() {
        return status;
    }
    public String getNama1() {
        return nama1;
    }
}
//Adapter Pemesanan
package com.example.lenovo.admin;
import android.app.Activity;
import android.content.Context;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;
import java.util.List;
public class adapterpemesanan extends BaseAdapter {
    Context context;
    List<pesantampil> array;
    public adapterpemesanan(Context context, List<pesantampil> array) {
        this.context = context;
        this.array = array;
    }
    @Override
    public int getCount() {
        return array.size();
    }
    @Override
    public Object getItem(int position) {
        return array.get(position);
    }
    @Override
    public long getItemId(int position) {
        return position;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        adapterpemesanan.Textv1 viewHolder;
        if (convertView == null) {
            convertView = LayoutInflater.from(context).inflate(R.layout.tvanggota, parent,
false);
            viewHolder = new Textv1(convertView);
```

```
            convertView.setTag(viewHolder);

        } else
            viewHolder = (adapterpemesanan.Textv1) convertView.getTag();

        pesantampil currentItem = (pesantampil) getItem(position);
        viewHolder.tvnamahold.setText(currentItem.getNama1());
        viewHolder.tvketeranganhold.setText(currentItem.getKeterangan());
        viewHolder.tvjumlah.setText(currentItem.getJumlah());
        if (currentItem.getStatus().equals("0"))
        {
            viewHolder.stat.setVisibility(View.VISIBLE);
            viewHolder.stat.setBackgroundResource(R.drawable.check);
            viewHolder.stat.setEnabled(false);
            viewHolder.tvketeranganhold.setEnabled(false);
            viewHolder.tvnamahold.setEnabled(false);
        }
        else
        {
            viewHolder.stat.setBackgroundResource(R.drawable.checked);
            viewHolder.stat.setEnabled(true);
        }
        return convertView;
    }

    public void updateRecords(List<pesantampil> array){
        this.array = array;
        notifyDataSetChanged();
    }

    private class Textv1
    {
        TextView tvnamahold,tvketeranganhold,tvjumlah;
        ImageView stat;
        public Textv1(View view) {
            tvnamahold=(TextView) view.findViewById(R.id.namapesanan);
            tvketeranganhold=(TextView) view.findViewById(R.id.keterangan);
            stat=(ImageView) view.findViewById(R.id.caca);
            tvjumlah=(TextView) view.findViewById(R.id.keteranganjumlah);
        }
    }
}
```