# 3. ANALISIS DAN DESAIN SISTEM

## 3.1. Analisis Permasalahan dan Kebutuhan

Pakan memiliki peran yang sangat penting dalam pemeliharaan ayam petelur, karena kualitas pakan secara langsung mempengaruhi performa, produksi telur, dan kesehatan ayam. Namun, tantangan utama yang dihadapi para peternak adalah kemampuan untuk menyediakan pakan yang memenuhi kebutuhan nutrisi optimal bagi ayam petelur pada setiap tahap pertumbuhannya, mulai dari tahap awal hingga tahap produksi telur utama. Sementara pabrik menyediakan campuran pakan siap pakai, biaya operasional yang terus meningkat mengharuskan para peternak untuk segera mengganti pakan pabrik menjadi *Self-mixing Feed* atau pembuatan pakan sendiri. Langkah ini dapat menghemat biaya yang sangat besar karena pengeluaran pada peternakan petelur ini adalah pakan ayamnya. Dengan membuat program untuk merancang formula pakan yang tepat sesuai dengan kebutuhan nutrisi dan harga yang paling murah, hal ini bisa sangat menghemat biaya operasional kandang dan meningkatkan kualitas produksi dan kesehatan ayam petelur di setiap fasenya.

Akan tetapi, bahan-bahan yang digunakan dalam pembuatan pakan ayam sangatlah banyak dan bervariasi. Setiap bahan juga memiliki kandungan nutrisi dan harga yang berbeda-beda. Setiap bahan makanan bisa dipilih tergantung dengan keinginan *user* dan keadaan geografisnya. Misalnya di kandang-kandang yang ada di Indonesia, akan lebih mudah mendapatkan jagung dibandingkan dengan gandum atau harga jagung di Indonesia lebih murah dibandingkan gandum. Hal ini yang menjadikan permasalah dan memerlukan program untuk memilih bahan mana yang paling optimal untuk digunakan menjadi pakan ayam. Karena selain adanya perbedaan harga, perbedaan kandungan nutrisi pada jagung dan gandum sangatlah berbeda. Sedangkan kita memiliki lebih dari 100 bahan yang bisa digunakan dalam pemilihan bahan untuk *Self-mixing Feed.*

Karena selama ini, para peternak hanya mengandalkan perhitungan sederhana baik menggunakan kertas atau dengan *excel* saja (rumus sederhana) dan hanya menggunakan bahan yang sangatlah terbatas. Misal nya untuk memenuhi kebutuhan energi dan protein, para peternak hanya mengandalkan jagung, katul, kedelai, dan gandum. Sedangkan diluar sana masih banyak bahan-bahan yang bisa digunakan dalam pemenuhan nutrisi untuk parameter tersebut. Selain bahan yang terbatas, dengan perhitungan yang masih sederhana, para peternak di Indonesia hanya menghitung sedikit batasan nutrisi yang harus dipenuhi. Peternak hanya memperhatikan seperti energi, protein, kalsium, fosfor, dan natrium. Padahal masih banyak lagi

parameter yang sangat penting yang harus diperhatikan juga dalam pemenuhan kebutuhan nutrisi ayam seperti *DM (Dry Matter), M.E. (Metabolizable Energy), Crude Protein, True Protein, E.E (Ether Extract), C.F (Crude Fiber), Ca (Kalsium), Total P (Total Phosphorus), Avail. P (Available Phosphorus), Ca:P (Calcium to Phosphorus ratio), Na (Sodium), Cl (Chloride), Choline Folate (*Vitamin B*), dLYS, dMET, dTSAA, dTHR, dTRP, dARG,* dan *dVALn (*Asam Amino*).*

Jadi program ini di desain untuk bisa menghasilkan solusi bahan apa yang dapat digunakan agar bisa menghasilkan hasil yang paling optimal yaitu semua parameter nutrisi terpenuhi dan harga yang paling murah.

### 3.2. Analisis data

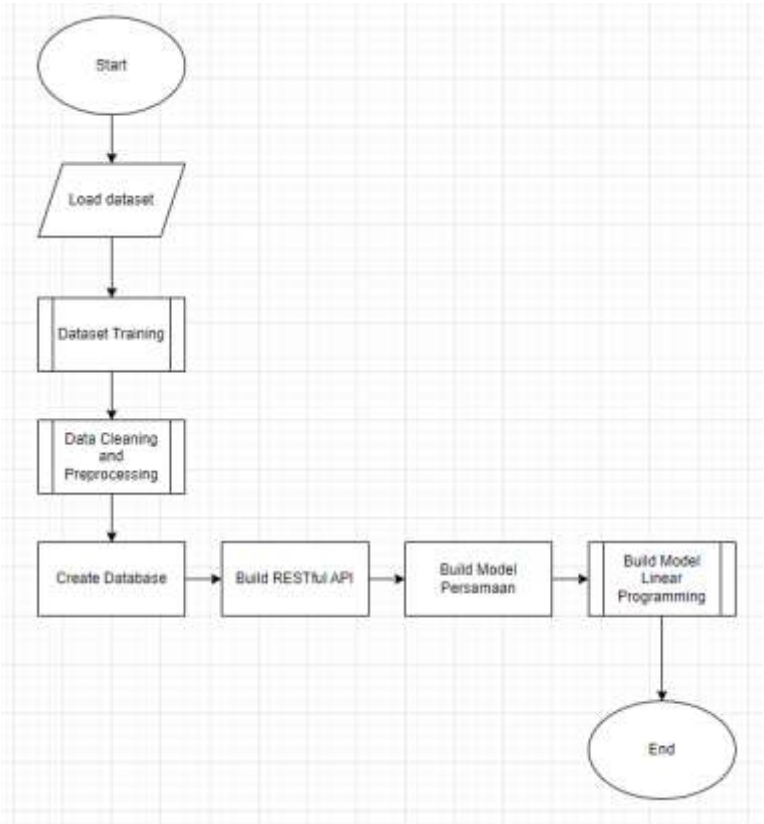
Data yang digunakan dalam penelitian ini berupa bahan-bahan makanan yang dapat digunakan dalam proses pembuatan S*elf-mixing feed* dan batasan nutrisi pada ayam petelur yang diambil dari beberapa buku *Poultry Book*. Data ini berupa file *CSV*. Data ini nantinya akan dimasukan kedalam database *MySQL* dan dihubungkan ke file *Python* menggunakan *RESTful API*.

Dataset yang pertama berisi tentang kumpulan lebih dari 100 bahan makanan yang terdapat juga harga dan kandungan nutrisi yang terkandung dari setiap bahan. Bahan makanan yang ada disana juga terbagi menjadi 2 tipe yaitu *CR* dan *SID*. Misalnya disana terdapat *CORN* [*SID]* dan *CORN [CR],* perbedaan antara *CORN [SID]* dan *CORN [CR]* adalah terletak pada aspek spesifiknya dalam konteks produk pertanian atau komoditas seperti jagung (*corn*) dapat mengacu pada berbagai hal, seperti varietas, kelas, atau metode pengolahan. Selain itu, perbedaan signifikan terletak pada sifat ketahanan masing-masing varietas terhadap faktor lingkungan yang berbeda. Jadi "*SID*" adalah varietas jagung yang tahan terhadap serangan serangga (*Superior to Insect Damage*) dan "*CR*" merujuk pada varietas jagung yang tahan terhadap suhu rendah (*Cold Resistant*). Varietas jagung yang tahan terhadap serangan serangga akan memiliki keunggulan dalam mengurangi kerugian hasil akibat serangan hama, sementara varietas yang tahan terhadap suhu rendah akan lebih cocok untuk daerah dengan musim dingin yang ekstrim. Oleh karena itu, perbedaan signifikan terletak pada adaptasi masing-masing varietas terhadap faktor lingkungan tertentu dan dampaknya terhadap hasil panen. Kandungan nutrisi yang terkandung meliputi *DM (Dry Matter), M.E. (Metabolizable Energy), Crude Protein, True Protein, E.E (Ether Extract), C.F (Crude Fiber), Ca (Kalsium), Total P (Total Phosphor), Avail. P (Available Phosphorus), Ca:P (Calcium to Phosphorus ratio), Na (Sodium), Cl (Chloride), Choline Folate* (Vitamin B), *dLYS, dMET, dTSAA, dTHR, dTRP, dARG,* dan *dVAL* (Asam Amino).

Dataset selanjutnya berisi tentang parameter nutrisi dari setiap jenis pakan ayam. Ada beberapa jenis pakan ayam yang bisa digunakan dalam 1 periode ayam petelur yaitu *starter*, *grower, pre-layer, layer, developer,* dan *finisher.* Hal yang membedakan jenis pakan ayam tersebut adalah umur ayam. Jadi pemberian jenis pakan ayam tersebut harus disesuaikan dengan umur ayam karena setiap jenis pakannya memiliki perbedaan kandungan nutrisi yang cukup signifikan untuk pertumbuhan dan perkembangan ayam. Selain itu, di setiap jenis pakan ayam tersebut juga memiliki banyak kriteria pembedanya seperti starter memiliki beberapa jenis yaitu *HY-LINE W-36 Starter 1 ( 0-3 wks), HY-LINE W-36 Grower ( 6-12 wks), HY-LINE W-36 Developer ( 12-15 wks), HY-LINE W-36 Pre-Lay ( 15-17 wks),* dan sebagainya.

### 3.3. Desain Sistem

Desain sistem membahas alur terbentuknya input data, proses dan output yang diharapkan oleh program ketika dijalankan. Juga disertakan user interface berbentuk *GUI* untuk program optimasi pakan ayam yang dapat digunakan user untuk memilih jenis pakan ayam, memilih bahan makanan yang dipakai, dan hasil formulasi pakan ayam dari jenis dan bahan makanan yang sudah dipilih. Berikut adalah *flowchart* alur keseluruhan sistem pada Gambar dibawah ini.

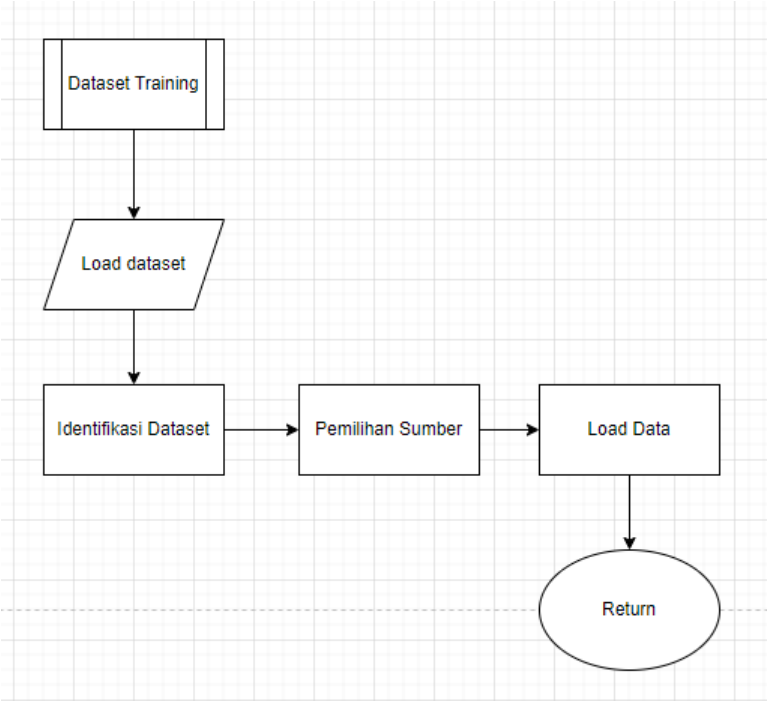


Gambar 3.1. Gambar flowchart program keseluruhan

Alur yang dilalui untuk proses program optimasi pakan ayam ini yang pertama adalah menyiapkan data-data kandungan nutrisi yang ada di setiap jenis pakan ayam dan bahan makanan beserta dengan kandungan nutrisinya. Setelah itu, dilakukan *dataset* t*raining* yang digunakan mengelompokan data-data tersebut ke dalam kelompoknya masing-masing. Misal nya, kandungan nutrisi yang ada dalam bahan makanan akan dikelompokan sesuai dengan kandungan nutrisi yang paling tinggi (contohnya energi, protein, serat, lemak, kalsium, fosfor, dan lain-lain). Selanjutnya data bahan makanan maupun parameter nutrisi akan diproses untuk menghindari adanya salah ketik atau kembar. Kemudian data tersebut akan dimasukan kedalam database lokal untuk memudahkan user untuk mengelola data. Jika *database* sudah dibuat, database tersebut akan dihubungkan ke dalam file *Python* menggunakan *RESTful API*.

Data yang sudah masuk ke dalam *database*, akan dipanggil di dalam *Python* untuk dibuat persamaan berdasarkan kandungan nutrisi yang ada. Setelah itu, data-data tersebut akan dimasukan hitung menggunakan metode Linear Programming untuk mencari solusi terbaik dari jenis pakan ayam dan bahan-bahan yang sudah dipilih dengan harga yang paling minimum.

### 3.3.1. Dataset Training




Gambar 3.2. Gambar flowchart data set training

Langkah 1: Identifikasi Dataset

Dataset dalam perhitungan kali ini berjumlah 2 file dengan format *CSV*. *Dataset* tersebut nantinya akan diubah menjadi format *SQL* untuk mempermudah dalam proses *import* kedalam *database* yang akan dibuat untuk program optimasi pakan ayam ini.

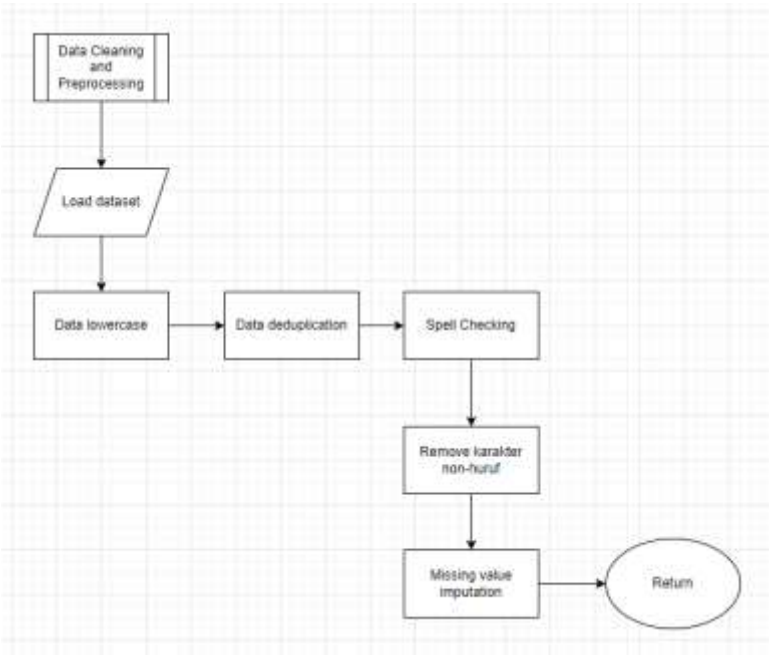
Langkah 2: Pemilihan Sumber

*Dataset* yang akan digunakan untuk proses perhitungan optimasi akan disimpan di database. *Database* untuk program optimasi pakan ayam memiliki 2 tabel penting yaitu tabel bahan makanan dan tabel parameter nutrisi untuk setiap jenis pakan ayam yang ada. Dengan menggunakan *database*, data yang dikelola akan lebih dengan efisien, seperti pencarian, penyortiran, dan penghapusan, sehingga memungkinkan untuk mendapatkan data yang dibutuhkan dengan cepat. Selain itu, *database* juga memiliki ketersediaan data yang tinggi, jadi bisa dipastikan bahwa data selalu dapat diakses dan digunakan dimanapun dan kapanpun saja. *Database* juga memiliki keamanan yang sangat tinggi yaitu dengan fitur-fitur seperti otentikasi pengguna, otorisasi akses, dan enkripsi data untuk melindungi data dari akses pihak lain.

Langkah 3: Load data

*Database* akan dihubungkan ke dalam program *Python* menggunakan *API*. Nantinya, beberapa tabel yang ada dalam database tersebut akan bisa dipanggil dengan menggunakan *library* seperti *React*. Dengan menggunakan *React* untuk menghubungkan antara *database* dan *API*, data dapat dengan mudah untuk mengambil data dan berinteraksi dengan data dari *database*, serta melakukan manipulasi data yang diperlukan. *React* juga mendukung penggunaan *RESTful* dalam membangun *API (RESTful API),* yang sesuai dengan prinsip-prinsip *REST* dan memungkinkan pembuatan *API* yang jelas, terstruktur, dan mudah dimengerti. *React* juga merupakan *framework website* yang ringan dan sederhana, yang memungkinkan pengembang untuk membuat *API* dengan cepat tanpa *overhead* yang berlebihan.

### 3.3.2. Data Cleaning and Preprocessing



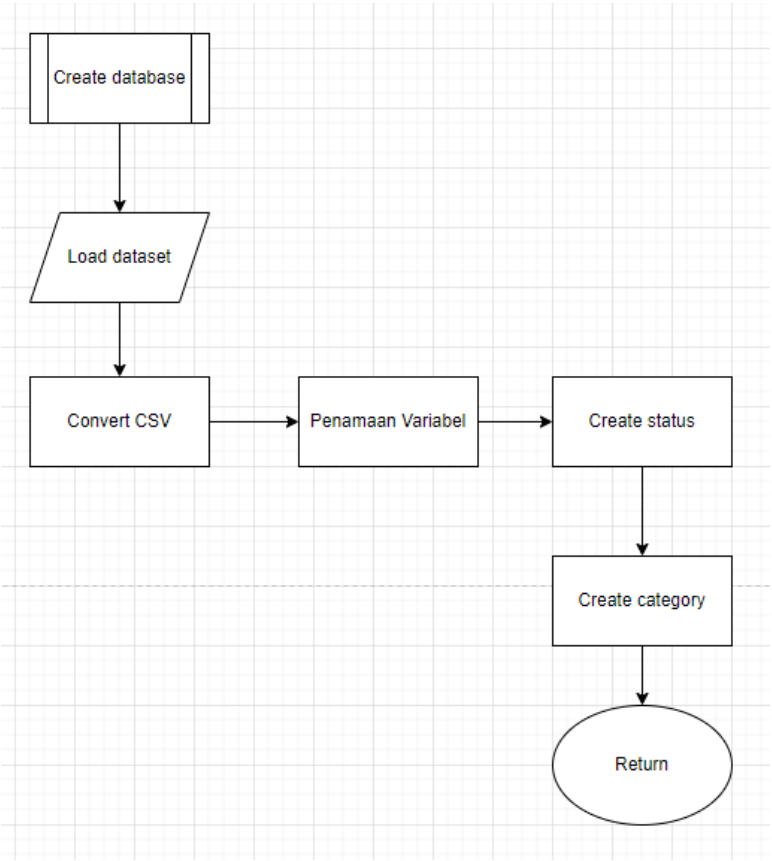

Gambar 3.3. Gambar flowchart data cleaning and preprocessing

Data cleaning dan preprocessing adalah tahapan kritis dalam penelitian data yang bertujuan untuk menghasilkan dataset yang berkualitas tinggi sebelum dilakukan analisis lebih lanjut. Proses ini mencakup serangkaian langkah untuk membersihkan, mengorganisir, dan mengubah data mentah menjadi bentuk yang lebih sesuai untuk pembuatan model. Tujuannya agar model yang dibuat akurat, benar, dan relevan dengan tujuan penelitian.

1. *Data Uppercase*: Ubah semua nama bahan makanan dan jenis pakan ayam menjadi huruf besar untuk memastikan konsistensi format

2. *Data Deduplication*: Hapus semua data yang kembar atau lebih dari 1 untuk mempermudahkan perhitungan dalam pembuatan persamaan
3. *Spell Checking*: Memeriksa dan memperbaiki kesalahan ejaan dalam nama bahan makanan jika untuk memastikan konsistensi dan kejelasan dalam data.
4. *Remove karakter non-huruf*: Hapus karakter-karakter non-huruf yang mungkin muncul, seperti angka atau simbol khusus, dari nama produk dan variabel
5. *Missing value imputation*: Masukan nilai 0 ke dalam *cell excel* yang kosong atau tidak ada nilainya baik secara manual, atau menggunakan Rumus *IF*: (Misalnya, *=IF(ISBLANK(A1), 0, A1)* akan menggantikan nilai kosong di *cell* A1 dengan 0.)

### 3.3.3. Create Database




Gambar 3.4. Gambar flowchart create database

Langkah 1: Convert CSV

Data yang berupa file *CSV* akan diubah menjadi file bertipe *SQL* untuk memudahkan dalam proses *import* kedalam *database* yang sudah disediakan. Kedua file *CSV* yang akan digunakan dimasukan kedalam *database* dan *primary key* yang ada didalam *database* tersebut

akan ditandai dengan kolom yang mempunyai label id. *Primary key* akan bersifat *auto increment* setiap kali ada penambahan data.

Langkah 2: Penamaan Variabel

Data bahan makanan yang sudah di-*import* kedalam *database* akan diberi tambahan kolom yang nantinya akan berisi variabel yang digunakan untuk mempermudah dalam pengambilan data. Misal Nya *CORN [SID]* menjadi *crn*, *MEAT & BONE MEAL. CP<45% [SID]* menjadi *mba*, *SOYBEAN MEAL (44%CP) [SID]* menjadi *sba*.

Langkah 3: Create status

Selain penambahan kolom variabel, setiap bahan makanan juga ditambahkan 1 data lagi yaitu kolom status yang bertujuan untuk mengatur bahan-bahan yang akan digunakan dalam proses optimasi. Jadi status akan memiliki 2 nilai yaitu 0 dan 1. Jika status bernilai 0, bahan makan tersebut tidak akan masuk dalam proses perhitungan. Dan jika status bernilai 1, bahan tersebut bisa masuk ke dalam proses perhitungan. Hal ini digunakan untuk meminimalisir hasil persamaan yang nantinya akan dibuat dalam proses optimasi.

Langkah 4: Create category

Selain penambahan kolom variabel dan status pada tabel makanan, tabel kategori juga merupakan salah satu hal yang penting untuk mempermudah proses pembuatan persamaan. Jadi tabel kategori digunakan untuk membagi bahan-bahan makanan sesuai dengan kategori yang sudah disediakan. Ada 19 kategori yaitu *ME, Crude Protein, True Protein, EE, CF, Total P, Avail_P, CaP, Na, Cl, Choline, Folate, dLYS, dMET, dTSAA, dTHR, dTRP, dARG,* dan *dVAL* yang nantinya akan berisi bahan-bahan yang akan dikelompokan kedalam kategori tersebut. Nantinya, di dalam tabel makanan akan diberi tambahan 1 kolom lagi yang berisi id pada tabel kategori yang nantinya nilai itu akan digunakan sebagai *primary key* pada tabel kategori untuk mengelompokkan setiap bahan makanan yang ada.

### 3.3.4. Build RESTful API

*Representational State Transfer (REST)* adalah pendekatan arsitektur perangkat lunak yang memungkinkan untuk merancang layanan web dengan cara yang terstruktur dan efisien. *RESTful API* adalah antarmuka pemrograman aplikasi *(API)* yang dibangun sesuai dengan prinsip-prinsip *REST*. Dalam hal ini, ada beberapa metode utama yang dapat dilakukan oleh *RESTful API,* yaitu:

- *GET*: Metode ini digunakan untuk meminta data dari *database* yang ditentukan. Permintaan *GET* tidak boleh mengubah status atau data pada server.
- *POST*: Metode ini digunakan untuk membuat tabel didalam *database*. *POST* digunakan untuk mengirim data ke server untuk diproses.
- *PUT*: Metode ini digunakan untuk memperbarui atau mengganti data yang ada atau membuat data yang baru jika tidak ada. Dalam konteks *RESTful API*, *PUT* sering digunakan untuk memperbarui keseluruhan data.
- *DELETE*: Metode ini digunakan untuk menghapus data yang ada dan ditentukan dari server.

### 3.3.5. Build Model Persamaan

Langkah 1: Inisialisasi kategori

Pada tahap ini, pembagian setiap bahan makanan ke dalam setiap kategori akan mempermudah dalam proses perhitungan. semua bahan akan dibagi ke dalam 19 kategori yaitu *ME, Crude Protein, True Protein, EE, CF, Total P, Avail_P, CaP, Na, Cl, Choline, Folate, dLYS, dMET, dTSAA, dTHR, dTRP, dARG,* dan *dVAL.* Jadi didalam kategori-kategori diatas akan berisi beberapa set item atau bahan makanan yang akan digunakan.

Langkah 2: Input bahan

Bahan-bahan yang akan digunakan akan dikelompokan kedalam 19 kategori diatas untuk mempermudah proses perhitungan. Untuk setiap kategori (i = kategori 1, kategori 2, kategori 3, … , kategori 19) memiliki nilai yaitu kandungan nutrisi yang ada. Misalnya kategori 1 yaitu energi, berisi jagung (*crn*), kedelai (*syb*), dan lain sebagainya, terdapat nilai dari setiap bahan tersebut yaitu nilai kandungan energi pada jagung, kedelai, dan seterusnya.

Langkah 3: Perhitungan persamaan

Persamaan untuk setiap kombinasi bahan dalam kategori-kategori yang diberikan adalah hasil penjumlahan dari produk nilai dengan nama bahan untuk setiap kategori yang sesuai.

$$Persamaan = \sum_{i=1}^{n} nilai\ numerik\ item\ i\ \times nama\ item\ i \tag{3.1}$$

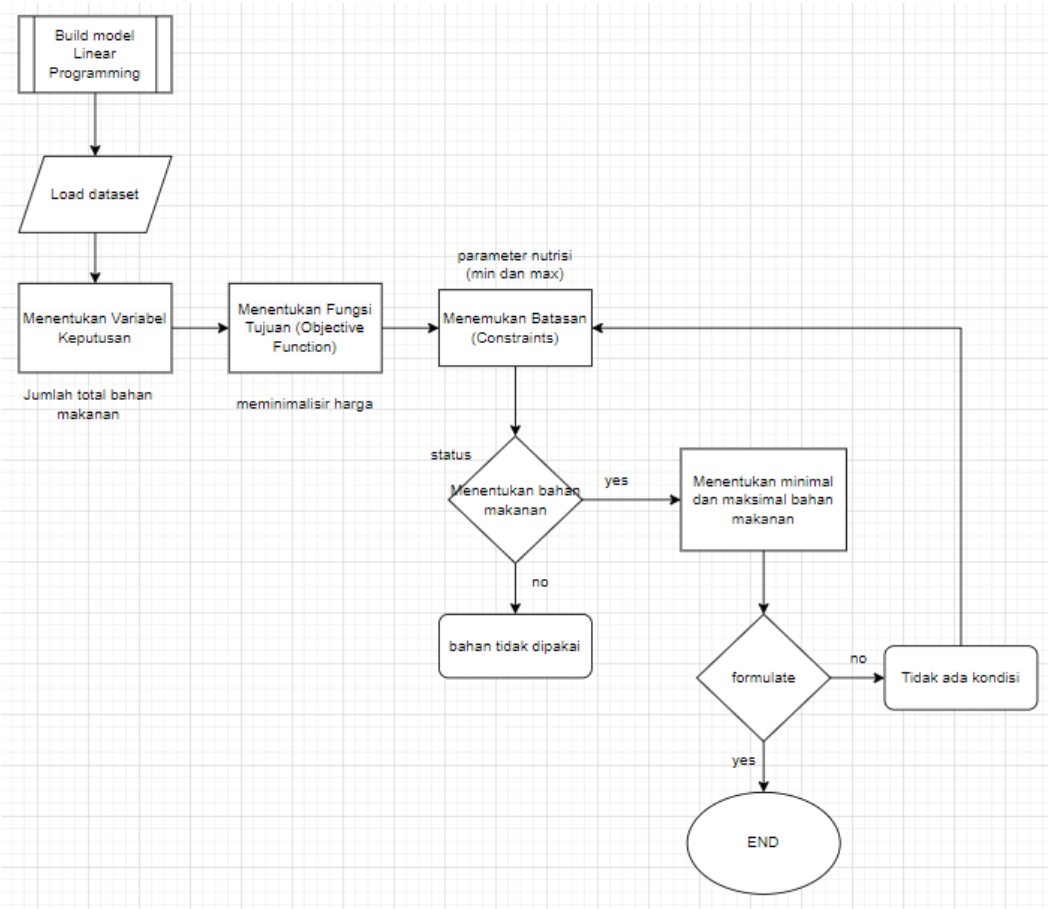
dimana:

- $n$adalah jumlah kategori.

- nilai numerik item_$i$ adalah nilai numerik dari item ke-$i$ dalam kategori ke-$i$.
- nama item_ $i$ adalah nama item ke-$i$ dalam kategori ke-$i$.

Proses ini diulangi untuk setiap kombinasi item dalam kategori-kategori yang ada.

### 3.3.6. Perhitangan Linear Programming




Gambar 3.5. Gambar flowchart model Linear Programming

Langkah 1: Menentukan Variabel Keputusan

Variabel keputusan adalah jumlah dari setiap *feed ingredient* yang digunakan dalam campuran pakan. Dalam program optimasi pakan ayam yang akan dibuat, terdapat fitur yang bernama status. Status ini berfungsi sebagai fitur untuk membuat bahan makanan yang kita ubah status nya akan masuk dalam perhitungan atau tidak. Bahan makanan yang memiliki status bernilai 0, bahan makanan tersebut tidak akan masuk dalam perhitungan. Jadi, bahan makanan tersebut tidak akan ditulis dalam variabel keputusan. Namun begitupun sebaliknya, jika bahan makanan memiliki status yang bernilai 1, maka bahan makanan tersebut akan dinyatakan dalam variabel keputusan untuk digunakan dalam perhitungan. Hal ini digunakan untuk meminimalisir

kesalahan dalam perhitungan (semakin banyak bahan, semakin banyak juga jumlah persamaan yang ada).

Langkah pertama dalam perhitungan ini adalah menentukan bahan-bahan apa saja yang akan masuk dalam perhitungan (status = 1), kemudian bahan-bahan makanan tersebut akan di masukan kedalam perhitungan $X_j$ yaitu jumlah feed ingredients $j$ yang digunakan dalam campuran pakan (dalam kg). Selanjutnya, bahan-bahan tersebut ketika dijumlahkan harus sama dengan total bahan makanan dengan mengunakan Rumus 2.3 yaitu menghitung *demand requirement* atau total bahan makanan.

Dalam hal ini, variabel keputusan $X_j$ yang mewakili jumlah feed ingredient $j$ yang digunakan dalam campuran pakan, di mana j berkisar dari 1 hingga n, dan n adalah jumlah total *feed ingredients*. Misalnya jagung, kedelai, gandum, dan katul yang akan digunakan dalam membuat komposisi bahan makanan yang optimal. Jadi, variabel keputusannya adalah $X_1$: jagung, $X_2$: kedelai, $X_3$: gandum, dan $X_4$: katul. Dimana semua bahan tersebut jika ditotal harus sesuai dengan jumlah total bahan makanan yang diminta *user*.

Langkah 2: Menentukan minimal dan maksimal bahan makanan

Selain status, terdapat juga fitur minimum dan maksimum yang digunakan dalam perhitungan formulasi pakan ayam ini. Kegunaan minimum adalah membuat bahan makan tersebut langsung masuk dalam perhitungan sesuai dengan jumlah yang dituliskan. Sedangkan maksimum, memiliki kegunaan untuk membatasi jumlah makanan yang akan digunakan dalam perhitungan formulasi pakan ayam. Kedua hal ini memiliki fungsi untuk mempermudah dalam perhitungan stok bahan makanan dan bisa memasukan atau membatasi jumlah bahan makanan sesuai dengan permintaan *user*.

Jika $Min\ j\ \neq 0, Max\ j\ \neq 0$

$$\frac{Min\ j}{100} \times t \leq X_j \leq \frac{Max\ j}{100} \times t \qquad (3.2)$$

Dimana:

- $X_j$ adalah adalah jumlah feed ingredient $j$ yang digunakan dalam campuran pakan (dalam kg)
- Min $j$ adalah nilai yang dimasukan *user* dalam kolom minimum (dalam persen)

- Max $j$ adalah nilai yang dimasukan *user* dalam kolom maksimal (dalam persen)
- $j$ adalah adalah kebutuhan total pakan yang harus diproduksi agar memenuhi permintaan atau *demand requirement*, diukur dalam kg.

Jadi dengan menggunakan Rumus 3.1, *user* dapat memberikan minimal dan maksimal jumlah pemakaian dari bahan makanan yang masuk kedalam variabel keputusan. Nilai yang dimasukan kedalam kolom min dan max adalam nilai dalam bentuk persentase. Misalnya jangung memiliki min 10% dan kedelai memiliki max 20% dengan total bahan makanan adalah 1500 kg, maka perhitungannya dapat dinyatakan sebagai:

$$X_{jagung} \geq \frac{10}{100} \times 1500 = 150\ kg$$

$$X_{Kedelai} \leq \frac{20}{100} \times 1500 = 300\ kg$$

Maka, pemakaian jagung minimal harus 150 kg dalam komposisi bahan makanan dan kedelai maksimal hanya boleh digunakan sebesar 300 kg dalam kompoisi bahan makanan. Dalam Segmen Program 4.12 Menambahkan batasan untuk minimum dan maksimum untuk setiap bahan makanan, *food_min_expr = solver.Constraint(min_value_food, solver.infinity())* dan *food_max_expr = solver.Constraint(-solver.infinity(), max_value_food)* digunakan untuk Membuat batasan bahwa jumlah bahan pakan sesuai dengan hasil perhitungan diatas.

Langkah 3: Menentukan Fungsi Tujuan (*Objective Function*)

*Objective Function* atau fungsi tujuan adalah fungsi untuk memaksimalkan atau meminimalkan. Dalam hal ini, tujuan dari program optimasi pakan ayam salah satunya adalah untuk membuat formulasi pakan ayam yang murah. Dalam hal ini, jika ingin meminimalkan total biaya dari *feed ingredients* yang digunakan dalam formulasi pakan. *Objective Function* dapat dihitung menggunakan Rumus 2.1 Menghitung *Objective Function* yaitu untuk meminimalisir harga, persamaan tersebut dapat dinyatakan sebagai:

$$Z = C_1X_1 + C_2X_2 + \cdots C_jX_j \tag{3.3}$$

Dimana:

- $Z$ adalah total biaya *feed ingredients* yang digunakan,
- $C_j$ adalah harga per kg *feed ingredient* $j$ dalam rupiah,

- $X_j$ adalah jumlah *feed ingredient* $j$ yang digunakan dalam campuran pakan (dalam kg)
- $n$ adalah jumlah total *feed ingredients* yang tersedia untuk digunakan dalam formulasi pakan.

Dengan menjumlahkan perkalian dari harga per kg setiap *feed ingredient* ($C_j$) dengan jumlah kg dari masing-masing feed ingredient ($X_j$) seperti pada Rumus 2.2, akan menghasilkan total biaya ($Z$) dari feed ingredients yang digunakan dalam formulasi pakan. Dengan menggunakan *objective = solver.Objective()* dan *objective.SetMinimization()* pada Segmen Program 4.9. Membuat function untuk meminimalisir harga, kode ini menunjukan bahawa tujuan dari linear programming adalah untuk meminimalkan nilai $Z$ (harga formulasi pakan ayam), sehingga akan menghasilkan solusi optimal yang memberikan formulasi pakan terbaik dengan biaya minimal.

Langkah 4: Menemukan Parameter (Constraints)

Parameter dalam perhitungan ini adalah batasan nutrisi yang harus dipenuhi dalam perhitungan pada proses optimasi. Dalam hal ini, ada beberapa parameter yang harus dibuat dalam persamaan, yaitu:

a. Demand Requirement:

Parameter kali ini bertujuan untuk mengharuskan total jumlah pakan yang diproduksi memenuhi permintaan user. Setelah menentukan variabel keputusan pada langkah pertama, diperlukan untuk memastikan bahwa jumlah xj pada variabel keputusan, ketika dijumlahkan harus sesuai dengan total bahan makanan. Perhitungan yang digunakan harus menjelaskan bahwa Xt = Xj, sehingga dapat dinyatakan sebagai:

$$X_t = X_1 + X_2 + X_3 + \cdots X_j \tag{3.4}$$

Dimana

- $X_t$ adalah total jumlah pakan yang diproduksi, diukur dalam kg. Ini merupakan jumlah dari semua feed ingredients ($X_j$) yang digunakan dalam campuran pakan.
- $t$ adalah kebutuhan total pakan yang harus diproduksi agar memenuhi permintaan atau *demand requirement*, diukur dalam kg.

Parameter ini menyatakan bahwa total jumlah pakan yang diproduksi harus setidaknya sama dengan kebutuhan total pakan ($t$) atau produksi pakan harus

memenuhi atau melebihi jumlah pakan yang dibutuhkan oleh permintaan *user*. Langkah ini hampir sama dengan langkah pertama, yang membedakan dengan langkah pertama adalah di tahap kali ini hanya perlu membuat persamaan dimana total pakan yang dihasilkan harus sama dengan total bahan-bahan yang sudah diinputkan oleh *user*. Kode *total_food_expr = solver.Constraint(1500, 1500)* pada Segmen Program 4.13. Menambahkan batasan total bahan makanan maksimum 1500 kg adalah *demand requirement* yang diminta *user*. Jadi, total jumlah pakan yang diproduksi harus sama persis 1500 kg.

b. Minimum Requirement:

Parameter ini memastikan bahwa jumlah nutrisi dalam hasil formulasi pakan ayam memenuhi parameter minimum sesuai dengan kebutuhan nutrisi ayam. Rumus 2.3 menjelaskan bahwa jumlah nutrisi $i$ dari semua bahan pakan $j$ memenuhi atau melebihi kebutuhan minimum $Bi$. Hal ini dilakukan dengan mengiterasi setiap nutrisi $i$, menetapkan batas bawah $Bi$ , dan kemudian menetapkan koefisien untuk setiap variabel keputusan $X_j$, dimana:

- $a_{ij}\ X_j$ adalah total jumlah nutrisi $i$ dalam pakan, diukur dalam satuan yang relevan. Ini merupakan jumlah dari semua nutrisi ($a_{ij}$) yang terkandung dalam feed ingredients ($X_j$) yang digunakan dalam campuran pakan.
- $Bi$ adalah adalah kebutuhan minimum nutrisi $i$ yang harus dipenuhi oleh pakan untuk ayam, diukur dalam satuan yang relevan.

Perhitungan ini menyatakan bahwa total jumlah nutrisi dalam pakan harus setidaknya sama dengan kebutuhan minimum nutrisi ($Bi$) untuk setiap jenis nutrisi $i$ yang dibutuhkan oleh ayam. Jadi, formula pakan yang dihasilkan oleh perhitungan harus mengandung jumlah nutrisi yang cukup untuk memenuhi kebutuhan minimum nutrisi sesuai dengan jenis pakan tertentu.

Dalam Segmen Program 4.10 Menambahkan batasan untuk minimum parameter nutrisi, untuk setiap kebutuhan nutrisi ($Bi$) yang ada minimum nutrisinya (*min_value != 0*), maka diperlukan perhitungan denagn menggunakan Rumus 2.3 untuk memastikan bahwa total nutrisi yang dihasilkan oleh bahan makanan ($X_j$) harus lebih dari paramter nutrisi yang sudah yang ada (*constraints_min*).

c. Maximum Requirement:

Selain *Minumnun Requirement*, Parameter ini memastikan bahwa jumlah nutrisi dalam hasil formulasi pakan ayam tidak melebihi parameter sesuai dengan kebutuhan nutrisi ayam. Rumus 2.4 menjelaskan bahwa jumlah nutrisi $i$ dari semua bahan pakan $j$ tidak melebihi kebutuhan maksimumum $Bi$. Hal ini dilakukan dengan mengiterasi setiap nutrisi i, menetapkan batas bawah $Bi$ , dan kemudian menetapkan koefisien untuk setiap variabel keputusan $X_j$, dimana:

- $a_{ij}\ X_j$ adalah total jumlah nutrisi $i$ dalam pakan, diukur dalam satuan yang relevan. Ini merupakan jumlah dari semua nutrisi (aij) yang terkandung dalam feed ingredients ($X_j$) yang digunakan dalam campuran pakan.
- $Bi$ adalah kebutuhan maksimum nutrisi $i$ yang dapat diterima oleh ayam, diukur dalam satuan yang relevan.

Perhitungan ini menyatakan bahwa total jumlah nutrisi dalam pakan tidak mekebihi kebutuhan maksimum nutrisi ($Bi$) untuk setiap jenis nutrisi $i$ yang dibutuhkan oleh ayam. Jadi, formula pakan yang dihasilkan oleh perhitungan harus mengandung jumlah nutrisi yang cukup untuk tidak melebihi kebutuhan maksimum nutrisi sesuai dengan jenis pakan tertentu.

Dalam Segmen Program 4.11 Menambahkan batasan untuk maksimum parameter nutrisi, untuk setiap kebutuhan nutrisi ($Bi$) yang ada maksimum nutrisinya (*max_value != 0*), maka diperlukan perhitungan denagn menggunakan Rumus 2.4 untuk memastikan bahwa total nutrisi yang dihasilkan oleh bahan makanan ($X_j$) tidak boleh melebihi paramter nutrisi yang sudah yang ada (*constraints_max*).

d. Non-negativity Constraints:

Parameter ini memastikan bahwa setiap variabel keputusan harus bernilai non-negatif (tidak boleh negatif). Secara matematis, $X_j \geq 0$ menyatakan bahwa setiap variabel keputusan $X_j$harus lebih besar dari atau sama dengan nol. Hal ini dinyatakan untuk setiap feed ingredient $j$, di mana $j$ dapat berada dalam rentang nilai dari 1 hingga $n$, dimana $n$ adalah jumlah total *feed ingredients* yang dipertimbangkan dalam perumusan masalah.

Langkah 5: Formulate atau Menentukan hasil

Langkah terakhir dalam proses perhitungan *Linear Programming* adalah program akan mencari solusi terbaik untuk menghasilkan pakan ayam yang memenuhi jumlah nutrisi tertentu dengan harga yang paling minimum. Program akan mengecek bahan makanan apa yang memiliki status yang bernilai 1, kemudian bahan makanan tersebut akan diidentifikasi ke dalam variabel keputusan. Setelah itu, program akan membuat persamaan untuk mengecek apakah bahan makanan tersebut memenuhi parameter minimum dan maksimum. Kemudian program akan membuat juga persamaan agar jumlah dari bahan-bahan tersebut tidak bernilai negatif. Selanjutnya, program akan menghasilkan jumlah persamaan yang berisi komposisi bahan makanan yang sangat banyak jumlahnya (komposisi bahan makanan yang memenuhi semua parameter persamaan). Lalu komposisi terbaik yang dihasilkan oleh program akan dipilih satu yang merupakan komposisi bahan makanan yang memiliki harga paling murah. Jika tidak ada komposisi bahan makanan yang memenuhi parameter minimum maupun melebihi maksimum (harga tidak akan dihitung), maka program akan berhenti.

### 3.4. Desain User Interface

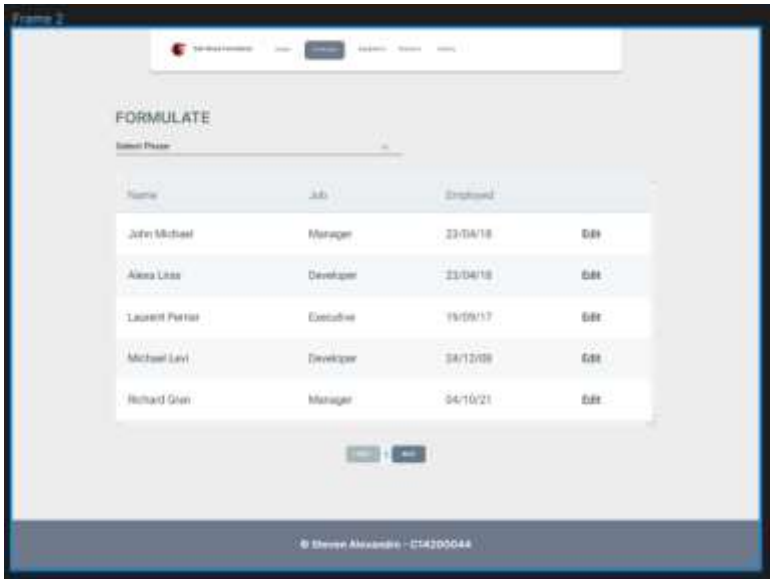
*Design User Interface* ini berisi tentang program optimasi pakan ayam yang dapat digunakan *user* untuk memilih jenis pakan ayam, memilih bahan makanan yang dipakai, dan hasil formulasi pakan ayam dari jenis dan bahan makanan yang sudah dipilih. Berikut design *user interface* pada *Graphical User Interface* yang dibuat menggunakan *figma*:

#### 3.4.1. Home


Gambar 3.6. Gambar Prototype Home Page

Pada *Home Page* akan berisi sebuah *navbar* yang terletah pada *header website* yang akan menampilkan beberapa menu yang bisa dipillih, yaitu halaman *Formulate*, *Ingredients*, *Nutritions*, dan *History*.

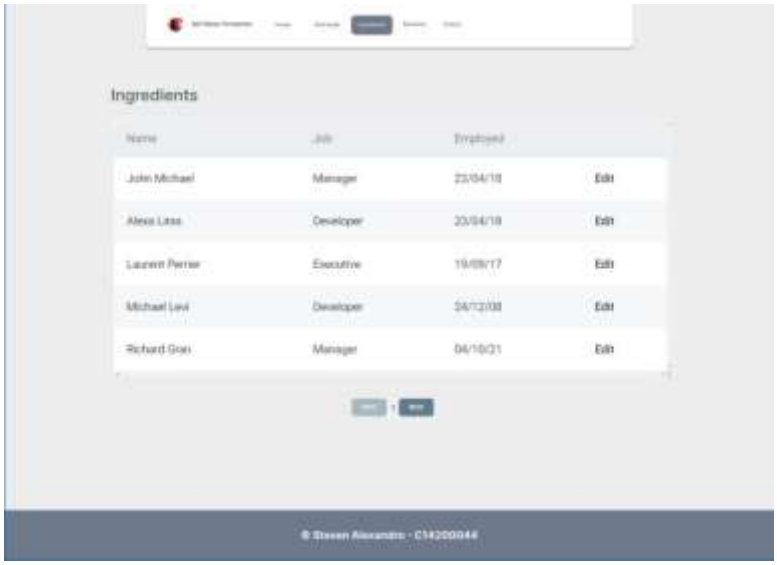
### 3.4.2. Formulate




Gambar 3.7. Gambar prototype formulate

Pada halaman *formulate*, halaman ini akan digunakan sebagai halaman untuk input *user* dan perhitungan program ini. Halaman ini akan berisi kolom yang digunakan untuk memilih jenis bahan makanan apa yang akan dibuat. Selain itu, terdapat juga tabel yang dapat digunakan *user* untuk memilih bahan makanan apa saja yang dapat digunakan *user* dalam proses perhitungan.

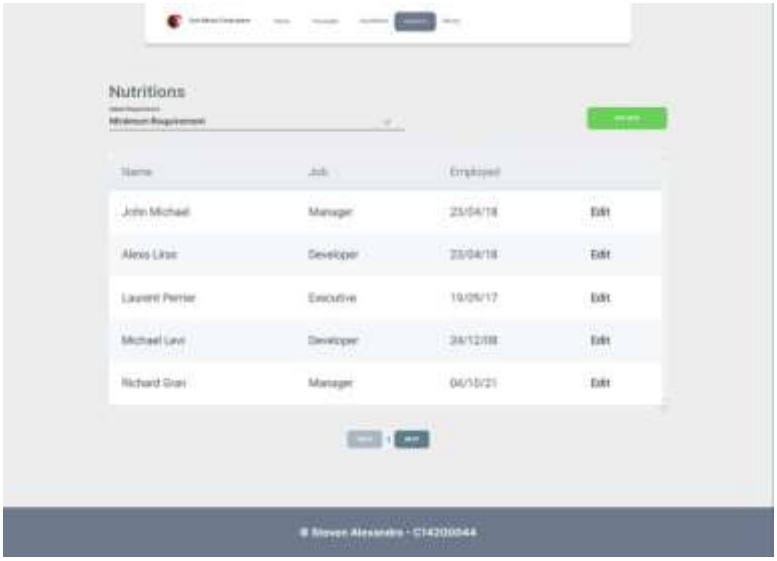
### 3.4.3. Ingredients




Gambar 3.8. Gambar Prototype Ingredients

Pada halaman *ingredients* akan berisi tabel bahan makanan beserta dengan harga dan kandungan nutrisi setiap bahan.

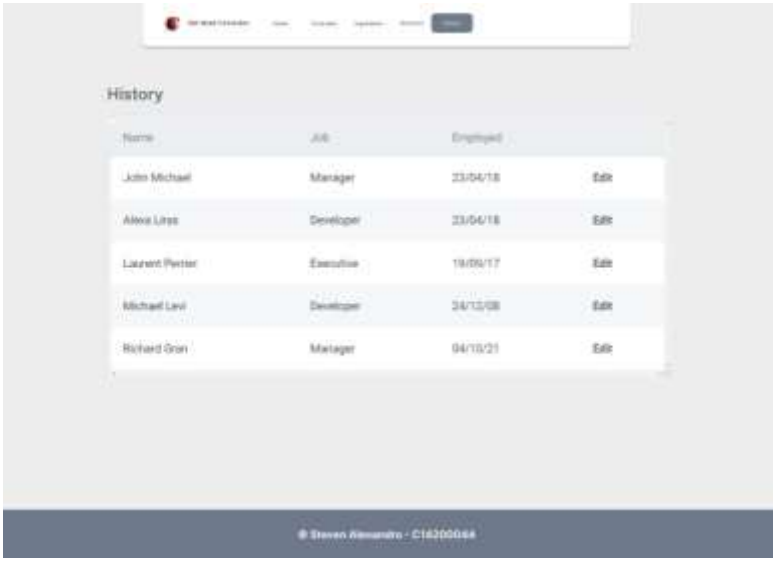
### 3.4.4. Nutritions




Gambar 3.9. Gambar Prototype Nutritions

Pada halaman *nutritions* akan terdapat tabel yang berisi nama dan kandungan nutrisi dari setiap jenis pakan ayam yang ada dalam program. Terdapat juga kolom untuk melihat parameter minimum maupun maksimum dari setiap jenis pakan ayam. Selain itu, *user* juga dapat membuat jenis pakan ayam yang baru sesuai dengan keinginan *user*.

### 3.4.5. History




Gambar 3.10. Gambar Prototype History

Pada halaman *history* terdapat tabel untuk melihat riwayat dari hasil perhitungan program ini. Selain dapat melihat tanggal, *user* dapat melihat jenis pakan ayam dan hasil bahan makanan apa yang sudah program buat komposisi nya.