

BAB 3 METODOLOGI PENELITIAN

3.1 Metode Penelitian

Pada penelitian ini, terdapat beberapa tahapan penelitian yang digunakan antara lain, telaah literatur, perancangan, implementasi, pengujian dan evaluasi, dan dokumentasi. Penjelasan tahapan penelitian dapat dilihat pada penjelasan berikut:

- Telaah literatur adalah tahapan untuk mencari referensi terkait masalah yang diteliti. Literatur yang digunakan dalam penelitian ini yaitu mengenai *enterprise resource planning (ERP)*, *index feeding*, *feed rate feeding*, *monte carlo simulation*, *end user computing satisfaction (EUCS)*, dan skala likert.
- Perancangan sistem dibagi menjadi dua tahap yaitu, perancangan sistem dalam bentuk diagram yang meliputi *flowchart*, *data flow diagram (DFD)*, dan *database schema* serta perancangan sistem dalam bentuk *graphical user interface (GUI)*.
- Implementasi sistem berupa sistem ERP berbasis *web* yang dibuat dengan *framework ReactJs* dan *Django*.
- Pengujian sistem dilakukan pada server lokal ketika proses pembangunan sistem ERP selesai. Dalam penelitian ini, dilakukan dua jenis pengujian berdasarkan koneksi dan lokasi, yakni pengujian dengan koneksi jaringan seluler di lokasi PT Samudra dan pengujian menggunakan koneksi *Wi-Fi* di luar lokasi PT Samudra. Pengujian ini mencakup evaluasi simulasi *monte carlo* dan pengukuran kecepatan waktu proses perhitungan. Tujuan dari pengujian ini juga melibatkan identifikasi kekurangan fitur atau *bug* dalam sistem. Pada setiap *bug* yang ditemukan, sistem akan kembali ke tahap penulisan kode hingga semua masalah diatasi dan memenuhi persyaratan yang telah ditetapkan.
- Pembuatan dokumentasi dilakukan dengan tujuan untuk mencatat setiap pengujian yang dilakukan.

3.2 Analisis Sistem

Hasil analisis dari kebutuhan proses bisnis dalam sistem ERP modul pengawasan dan perhitungan pakan udang dengan algoritma monte carlo pada PT Samudra dibagi menjadi dua hal. Pertama, kebutuhan *input output* data pada sistem. Kebutuhan *input* dari sistem ERP ini antara lain, data *user* dan *role* untuk akses sistem, data induk (*master data*) yang meliputi data *suppliers*, *supplier types*, *ponds*, *clusters*, *feeding rate presets*, dan *feeding rate parameters*, data *feeding entries*, dan data *active ponds*.

Kedua, kebutuhan dari segi fungsionalitas. PT Samudra belum memiliki sistem ERP untuk menunjang pengawasan dan perhitungan pakan udang. Perhitungan pakan udang saat ini hanya menggunakan papan tulis putih dan *google spreadsheets*. Fungsionalitas sistem ERP diharapkan dapat melakukan perhitungan rekomendasi pakan udang yang lebih kompleks. Hasil dari kebutuhan yang telah dipaparkan, sebuah sistem ERP berbasis *web* dapat digunakan untuk memenuhi fungsionalitas perhitungan rekomendasi pakan yang kompleks serta menggantikan pencatatan data yang menggunakan papan tulis dan *google spreadsheets*.

Proses pembuatan sistem diputuskan menggunakan algoritma monte carlo sebagai algoritma utama untuk perhitungan rekomendasi pakan dalam sistem. Sistem ERP berbasis *web* ini dibuat dengan *framework* ReactJS untuk sisi *frontend* dan Django untuk sisi *backend*. Algoritma monte carlo digunakan untuk melakukan perhitungan rekomendasi pakan pada sisi *backend* yang kemudian dikirimkan pada sisi *frontend* melalui *Application Programming Interface* (API).

3.3 Perancangan Sistem

Dari analisis sistem yang telah dilakukan, maka dirancang sebuah sistem ERP berbasis *web* dengan beberapa fungsionalitas. Fungsionalitas yang terdapat pada sistem sebagai berikut:

- a. Proses *create*, *read*, *update*, *delete* data *user* dan *role*

Proses ini memungkinkan superadmin untuk mengelola data *user* dan *role* yang terdapat pada sistem.

b. Proses *create, read, update, delete master data*

Proses ini berguna untuk mengelola data induk (*master data*) dari data kolam, data *cluster*, data *supplier*, dan *feeding rate preset* yang digunakan untuk mendukung kegiatan budidaya udang.

c. Proses pencatatan *active ponds* dan *feeding entries*

Proses pencatatan kolam yang sedang aktif budidaya dan pemberian pakan merupakan proses utama. Proses ini memungkinkan pengguna untuk melihat pergerakan pakan serta pertumbuhan udang hari demi hari. Selain itu pencatatan ini dapat digunakan untuk menghitung rekomendasi pakan udang.

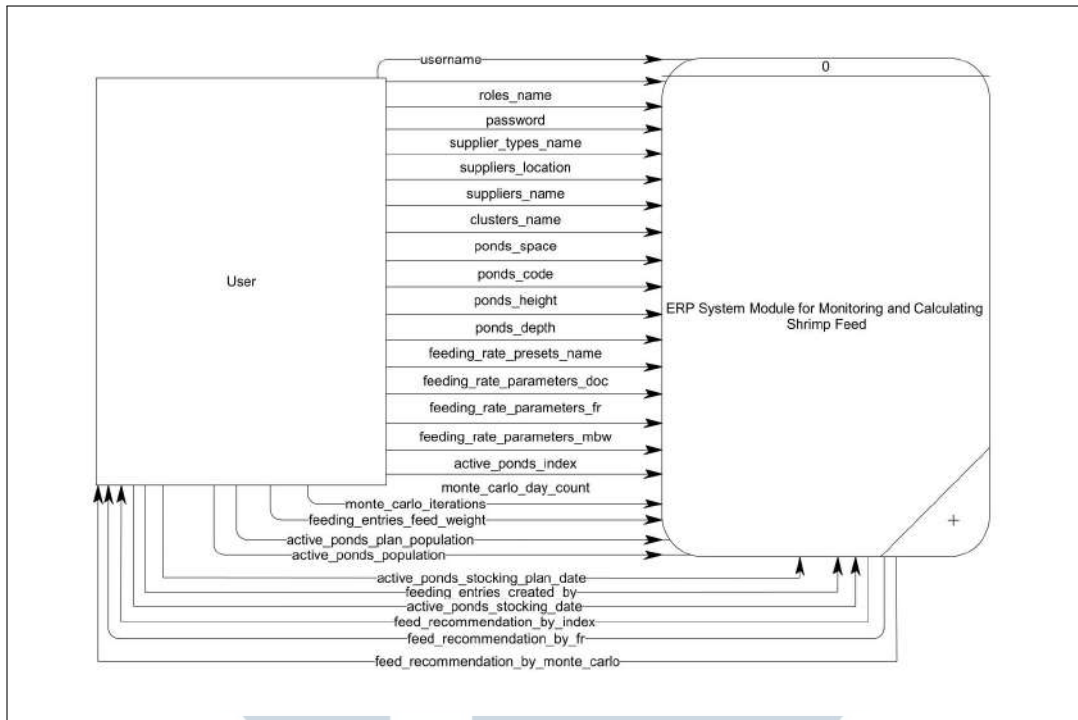
d. Proses perhitungan rekomendasi pakan

Pada proses ini, pengguna dapat menerima *output* dari sistem ERP yaitu rekomendasi pakan yang diberikan untuk hari berikutnya. Rekomendasi pakan yang diberikan oleh sistem ERP merupakan hasil perhitungan dari tiga metode perhitungan yaitu *index feeding* dengan Rumus 2.2, *FR feeding* dengan Rumus 2.4, dan *monte carlo* dengan Rumus 2.5 hingga Rumus 2.7. Ketiga metode ini menghasilkan angka rekomendasi yang berbeda-beda sehingga pengguna dapat memilih rekomendasi pakan yang tepat untuk budidaya.

3.3.1 Data Flow Diagram

Berdasarkan fungsionalitas sistem yang telah dijabarkan maka dirancanglah sebuah alur data yang terdapat pada sistem saat dijalankan. Alur data dirancang dalam bentuk *Data Flow Diagram* (DFD). DFD yang dibuat dimulai dari *Context Diagram* yang kemudian diturunkan menjadi beberapa level. DFD akan dijabarkan sebagai berikut.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

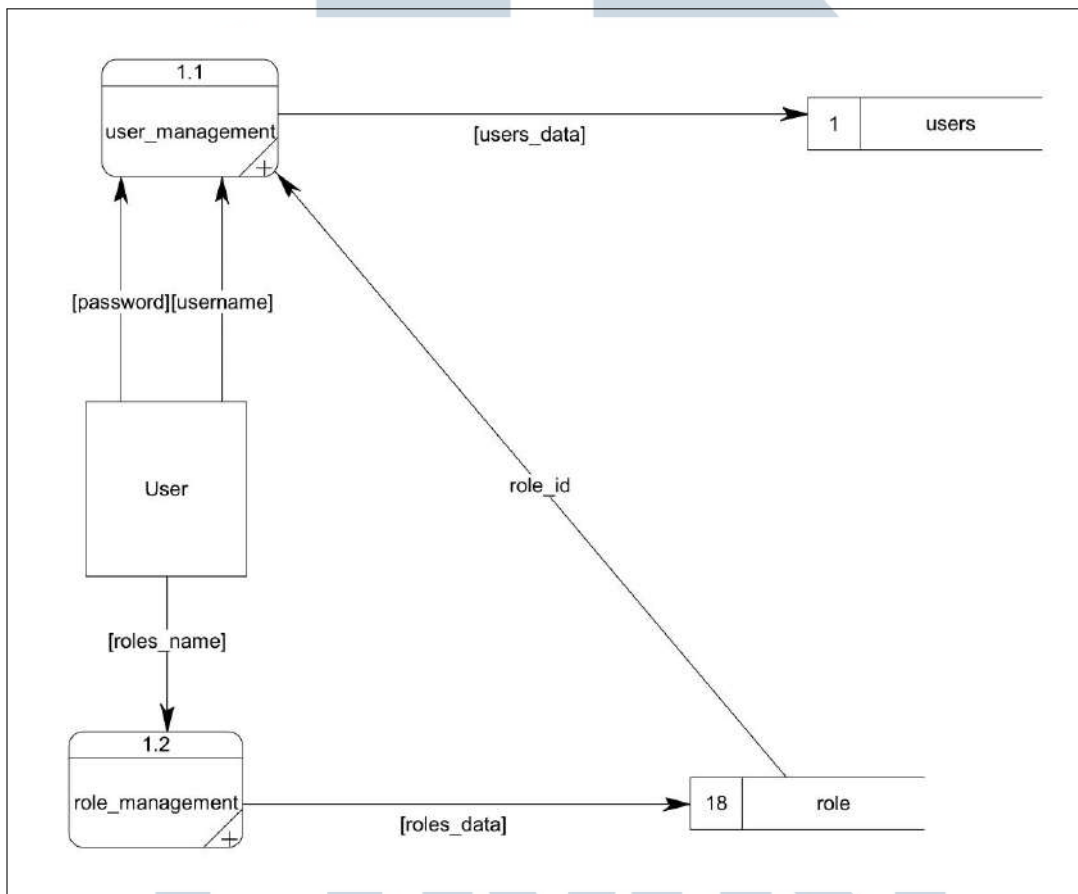


Gambar 3.1. *Context Diagram* Sistem ERP

Gambar 3.1 merupakan DFD *Context Diagram* sistem ERP menggunakan algoritma monte carlo yang menggambarkan aliran data sistem secara menyeluruh.



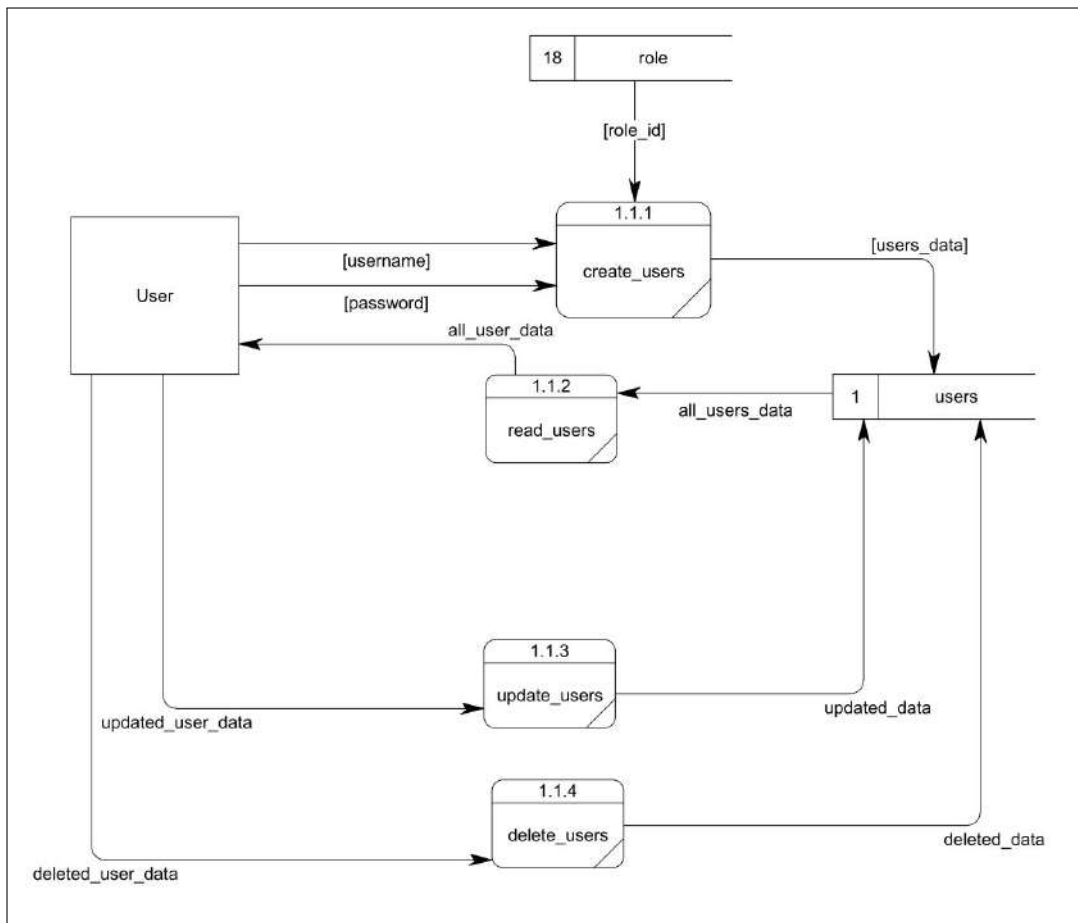
Gambar 3.2 merupakan hasil dekomposisi dari DFD *Context Diagram*. DFD *Context Diagram* dibagi ke dalam 4 proses utama sistem yaitu *User & Role Management*, *Master Data Management*, *Active Ponds Management*, dan *Feeding Entry & Recommendation*.



Gambar 3.3. DFD Level 2 *User & Role Management*

Gambar 3.3 menjelaskan lebih detail proses *User & Role Management* yang dibagi ke dalam dua subproses yaitu *User Management* dan *Role Management*.

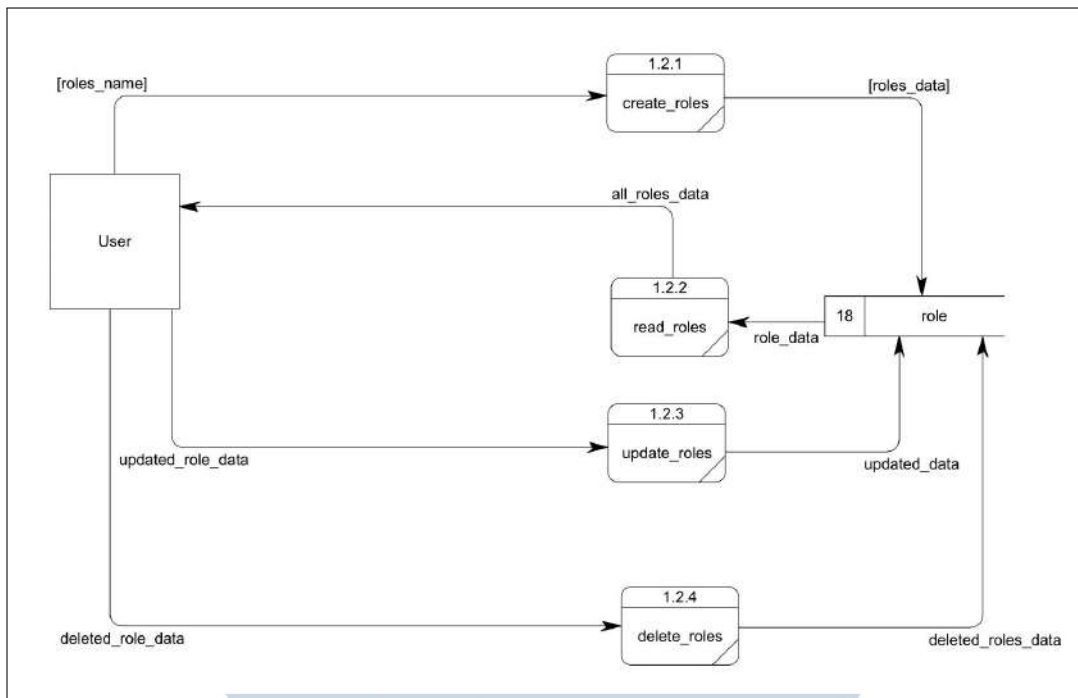
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.4. DFD Level 3 *User Management*

Gambar 3.4 merupakan subproses dari DFD Level 2 *User & Role Management*. Terdapat empat buah subproses yang dalam proses *user management* yaitu membuat *user* baru, melihat *user*, mengubah data *user*, dan menghapus *user* yang terdapat dalam sistem. Pada subproses tersebut, memerlukan data *role* yang didapatkan dari Tabel Roles.

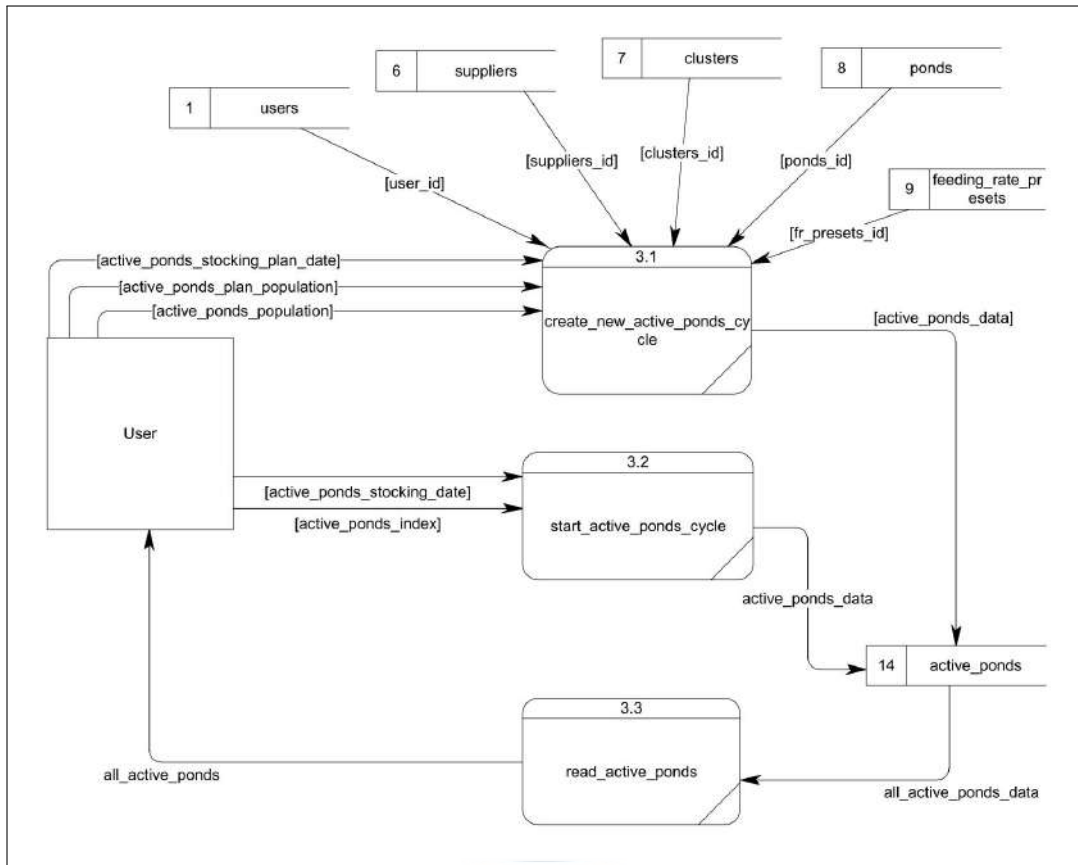
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.5. DFD Level 3 Role Management

Gambar 3.5 merupakan subproses dari DFD Level 2 *User & Role Management*. Terdapat empat buah subproses yang dalam proses *role* yaitu, membuat *role* baru, melihat *role*, mengubah data *role*, dan menghapus *role* yang terdapat dalam sistem.

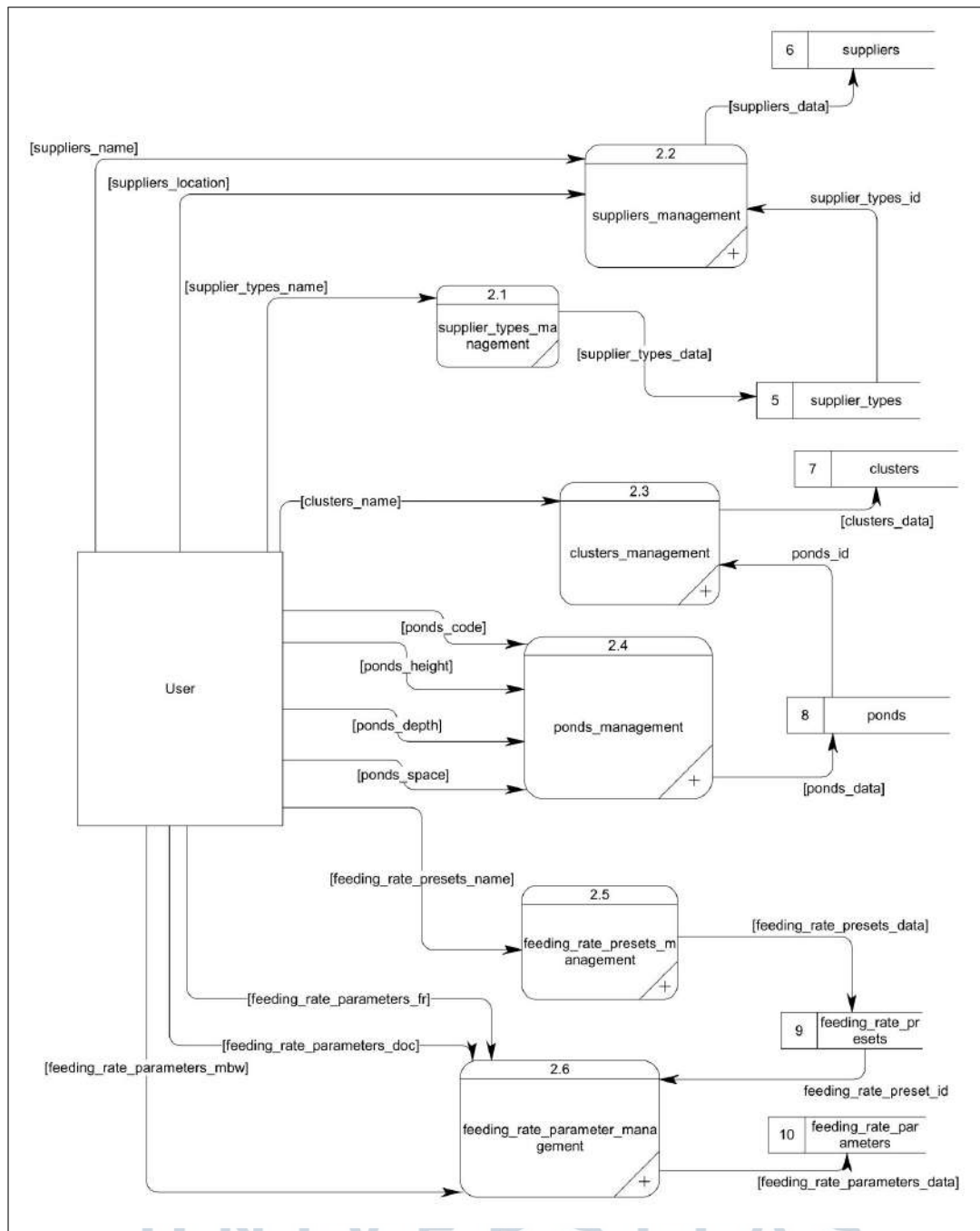




Gambar 3.6. DFD Level 2 Active Ponds Management

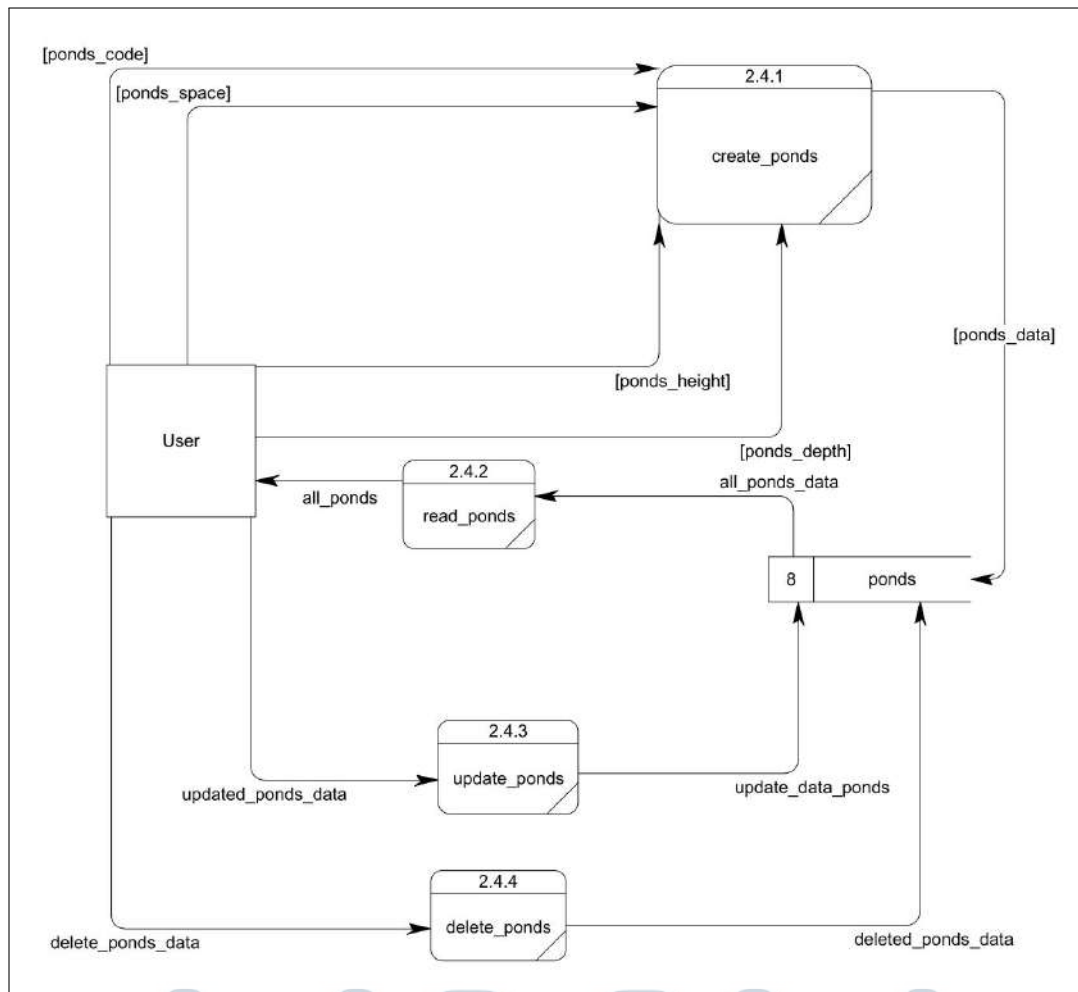
Gambar 3.6 menjelaskan proses *active ponds management* dibagi ke dalam tiga buah subproses yaitu, *start new cycle*, *confirm stocking*, dan melihat *active ponds*. *Start New Cycle* merupakan proses saat awal mula pengguna ingin menggunakan kolam tersebut untuk memulai budidaya. *Confirm Stocking* adalah proses dimana memastikan bahwa rencana budidaya sudah sesuai. Pada proses ini memerlukan data dari beberapa tabel antara lain Tabel User, Supplier, FR Preset, Clusters, dan Ponds.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.7. DFD Level 2 Master Data

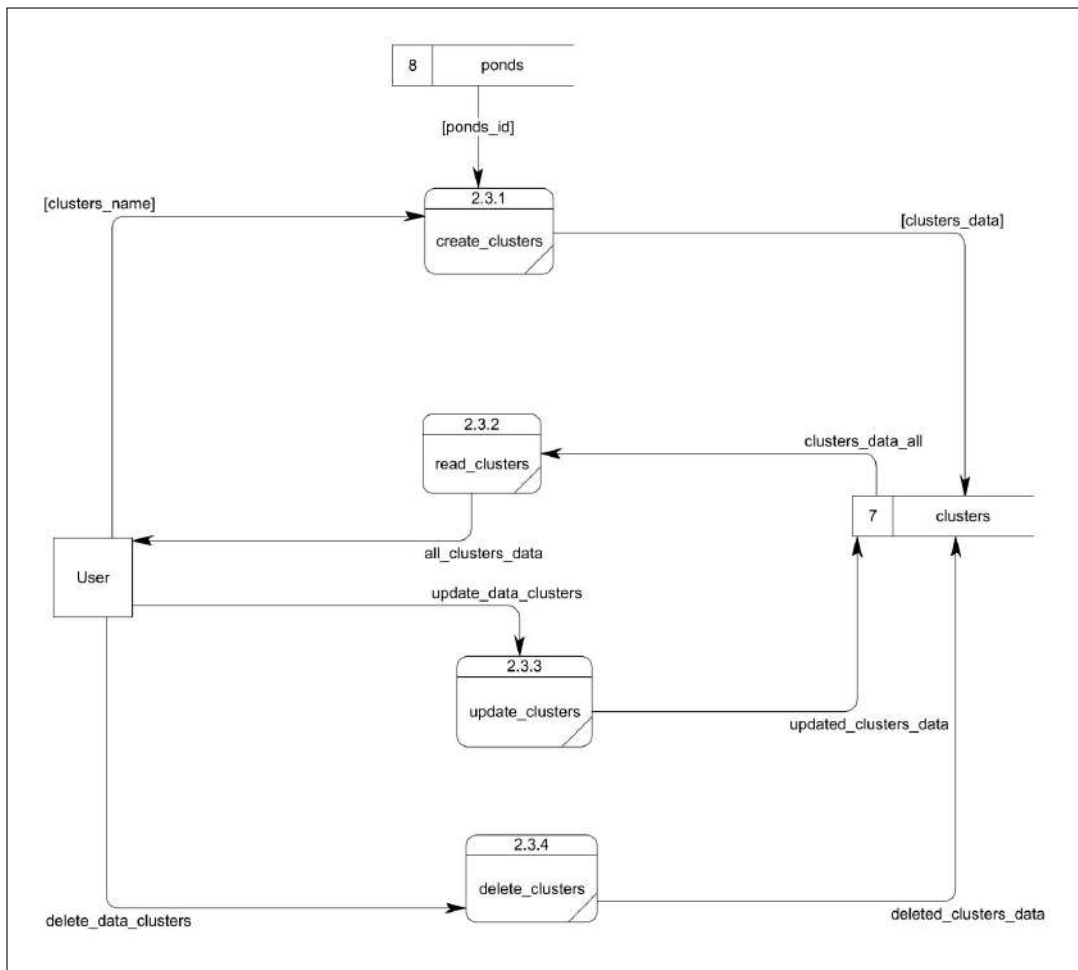
Gambar 3.7 menjabarkan proses *master data* dibagi ke dalam enam subproses yaitu *ponds*, *clusters*, *suppliers*, *suppliers type*, *FR presets*, *FR Params*.



Gambar 3.8. DFD Level 3 *Master Data Ponds Management*

Gambar 3.8 merupakan hasil dekomposisi dari DFD Level 2 *Master Data* tepatnya pada *Ponds Management*. Terdapat empat subproses pada *ponds management* yaitu, membuat *ponds* baru, melihat *ponds*, mengubah data *ponds*, dan menghapus *ponds* yang terdapat dalam sistem.

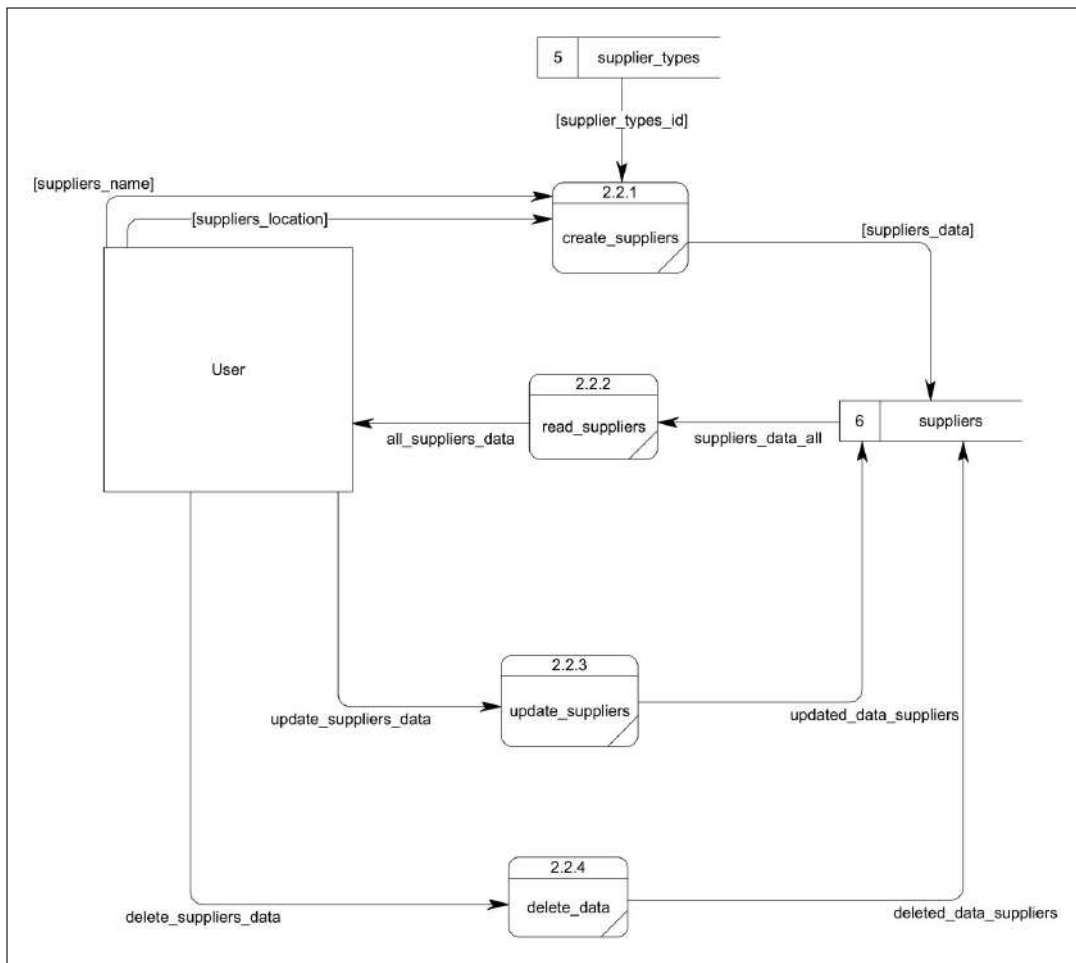
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.9. DFD Level 3 *Master Data Clusters Management*

Gambar 3.9 merupakan hasil dekomposisi dari DFD Level 2 *Master Data* tepatnya pada *Clusters Management*. Pada proses tersebut dibagi ke dalam empat subproses yaitu, membuat *clusters* baru, melihat *clusters*, mengubah data *clusters*, dan menghapus *clusters* yang terdapat dalam sistem. Pada proses ini memerlukan data *ponds* yang diambil dari Tabel Ponds.

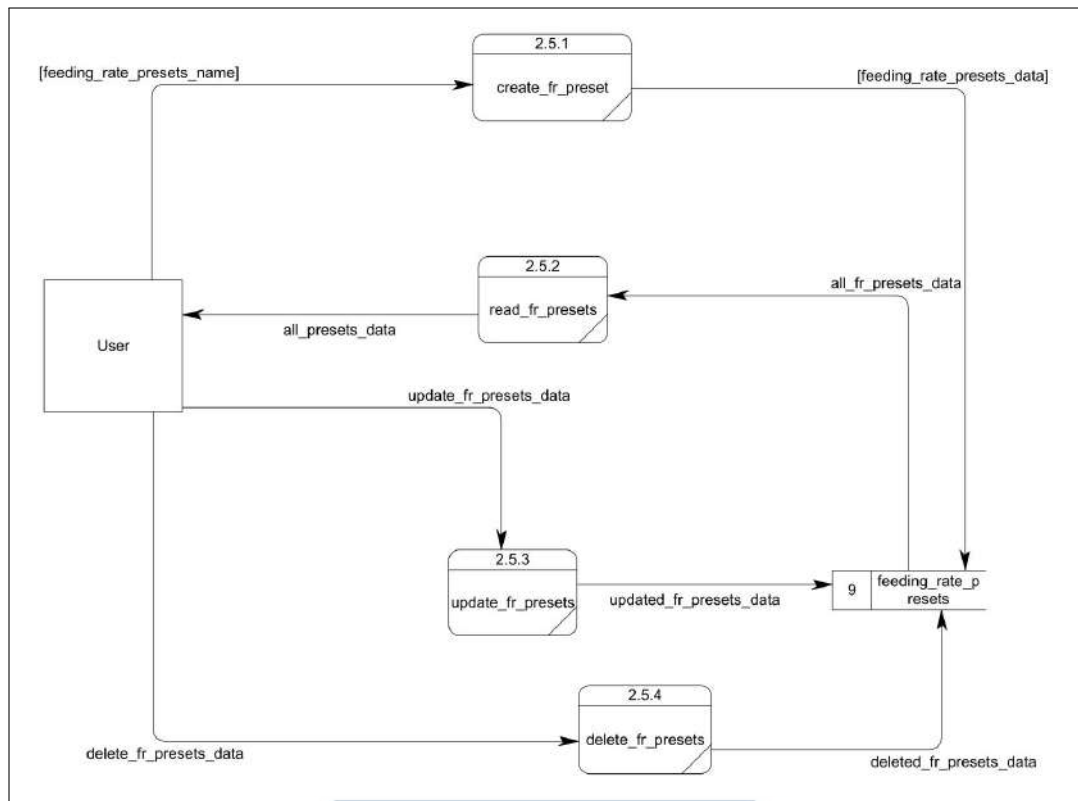
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.10. DFD Level 3 *Master Data Suppliers Management*

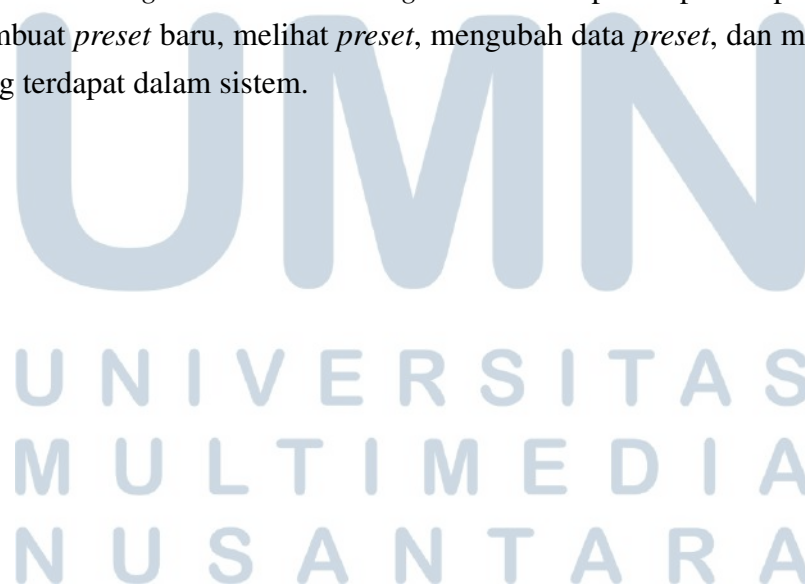
Gambar 3.10 menjelaskan lebih detail proses *suppliers management*. Proses *suppliers management* dibagi ke dalam empat subproses yaitu, membuat *suppliers* baru, melihat *suppliers*, mengubah data *suppliers*, dan menghapus *suppliers* yang terdapat dalam sistem. Pada proses ini memerlukan data *supplier types* yang diambil dari Tabel Supplier Types.

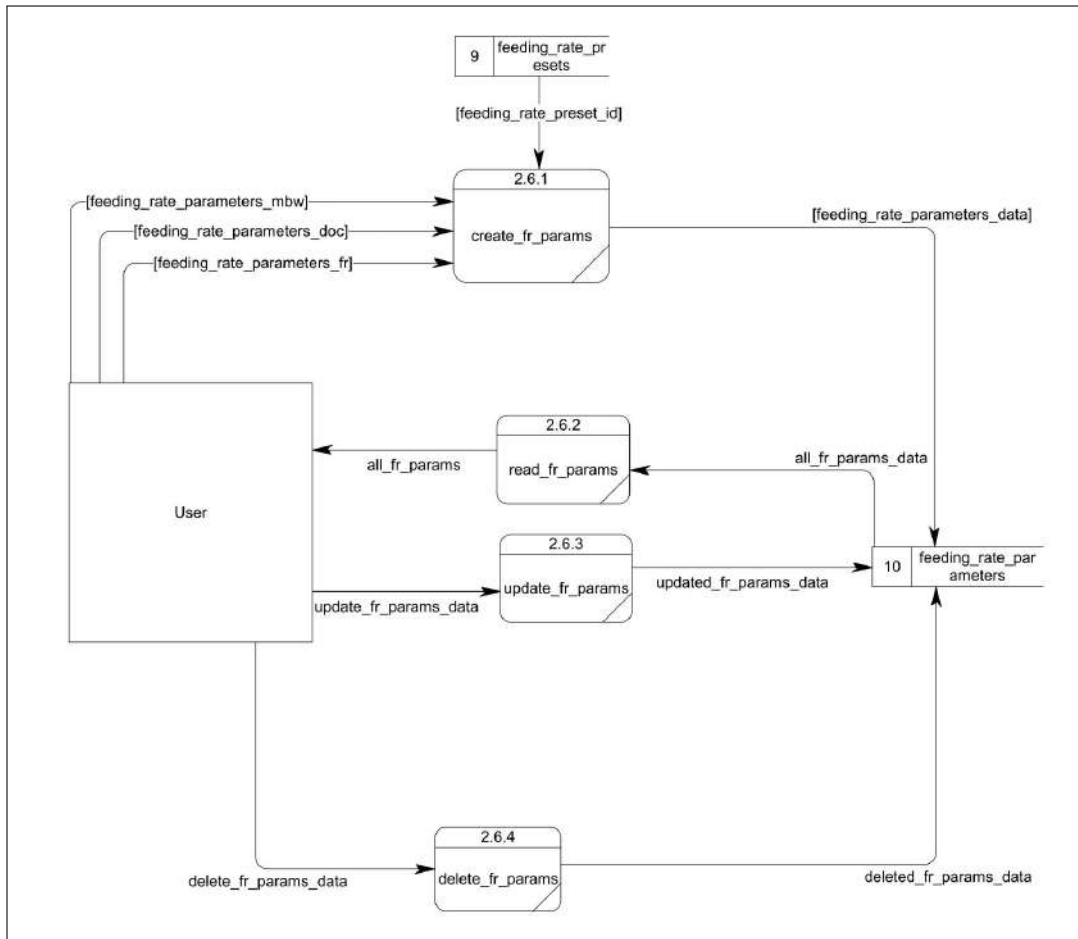
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.11. DFD Level 3 *Master Data Feeding Rate Preset Management*

Gambar 3.11 merupakan hasil dekomposisi dari DFD Level 2 *Master Data* tepatnya pada *Feeding Rate Preset Management*. Terdapat empat subproses pada yaitu, membuat *preset* baru, melihat *preset*, mengubah data *preset*, dan menghapus *preset* yang terdapat dalam sistem.

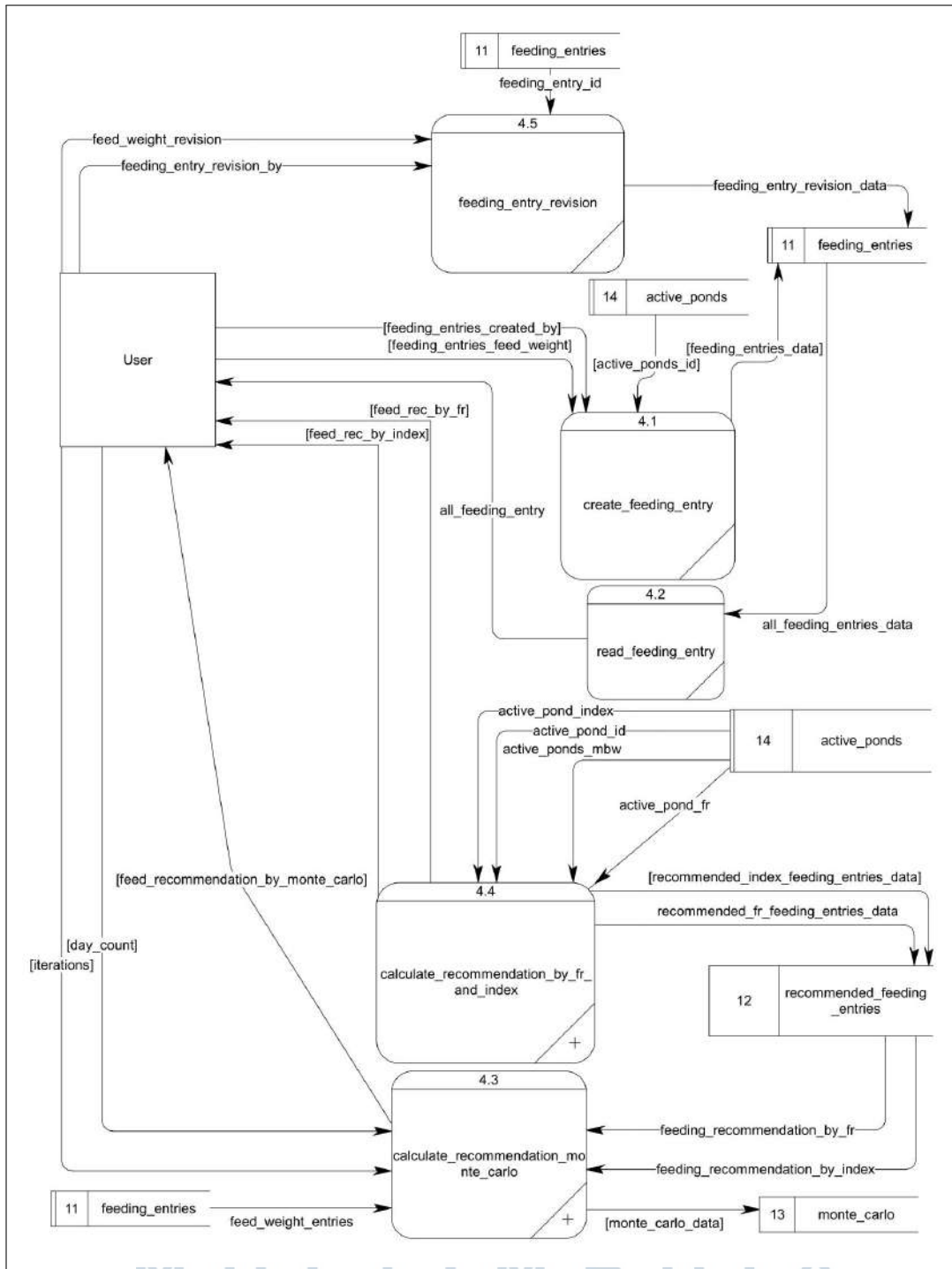




Gambar 3.12. DFD Level 3 *Master Data Feeding Rate Parameters Management*

Gambar 3.9 merupakan hasil dekomposisi dari DFD Level 2 *Master Data* tepatnya pada *Clusters Management*. Pada proses tersebut dibagi ke dalam empat subproses yaitu, membuat *parameters* baru, melihat *parameters*, mengubah data *parameters*, dan menghapus *parameters* yang terdapat dalam sistem. *Parameter* yang telah dibuat akan digunakan untuk sebuah *preset*. maka dari itu proses ini memerlukan data *FR Presets* yang diambil dari Tabel Feeding Rate Presets.

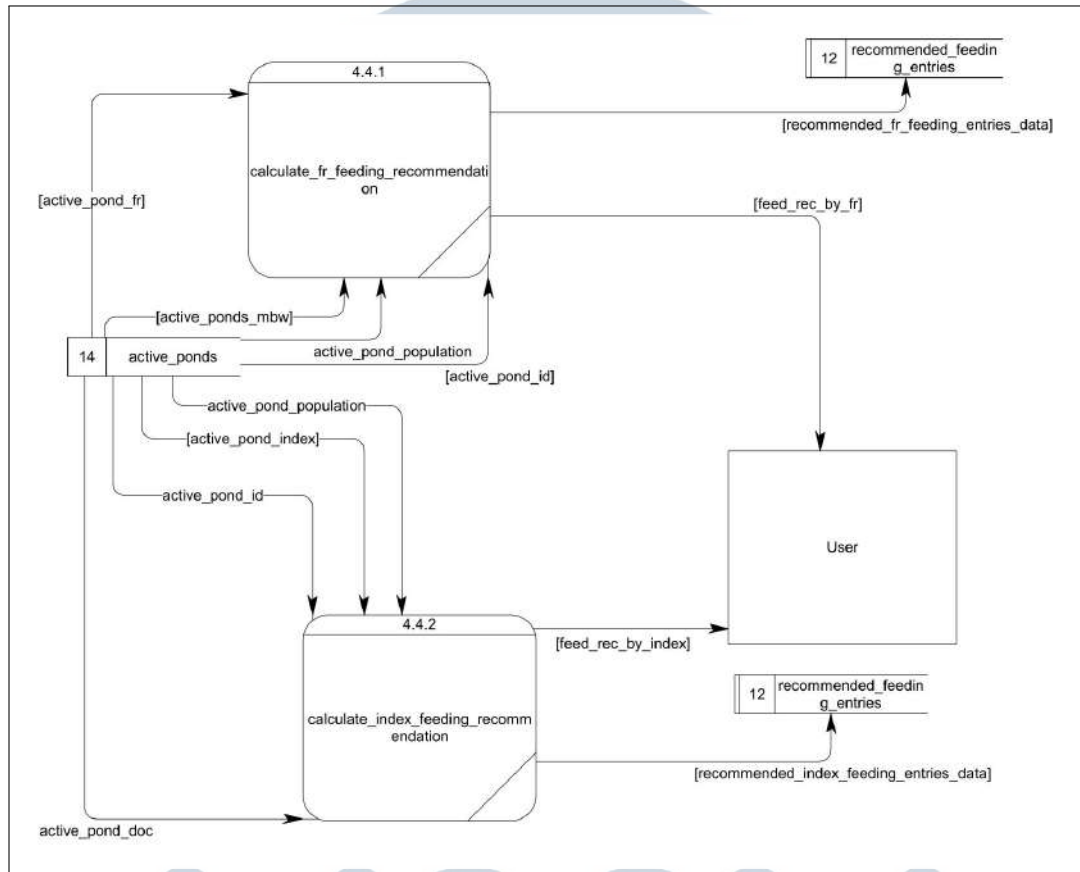
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.13. DFD Level 2 *Feeding Entry and Recommendation*

Gambar 3.13 menjelaskan proses *feeding entry and recommendation* yang dibagi menjadi lima subproses yaitu, *feeding entry revision*, *create feeding entry*, *read feeding entry*, *calculate feeding recommendation by fr and index*, dan *calculate feeding recommendation by monte carlo*. Pada proses ini mengeluarkan tiga buah

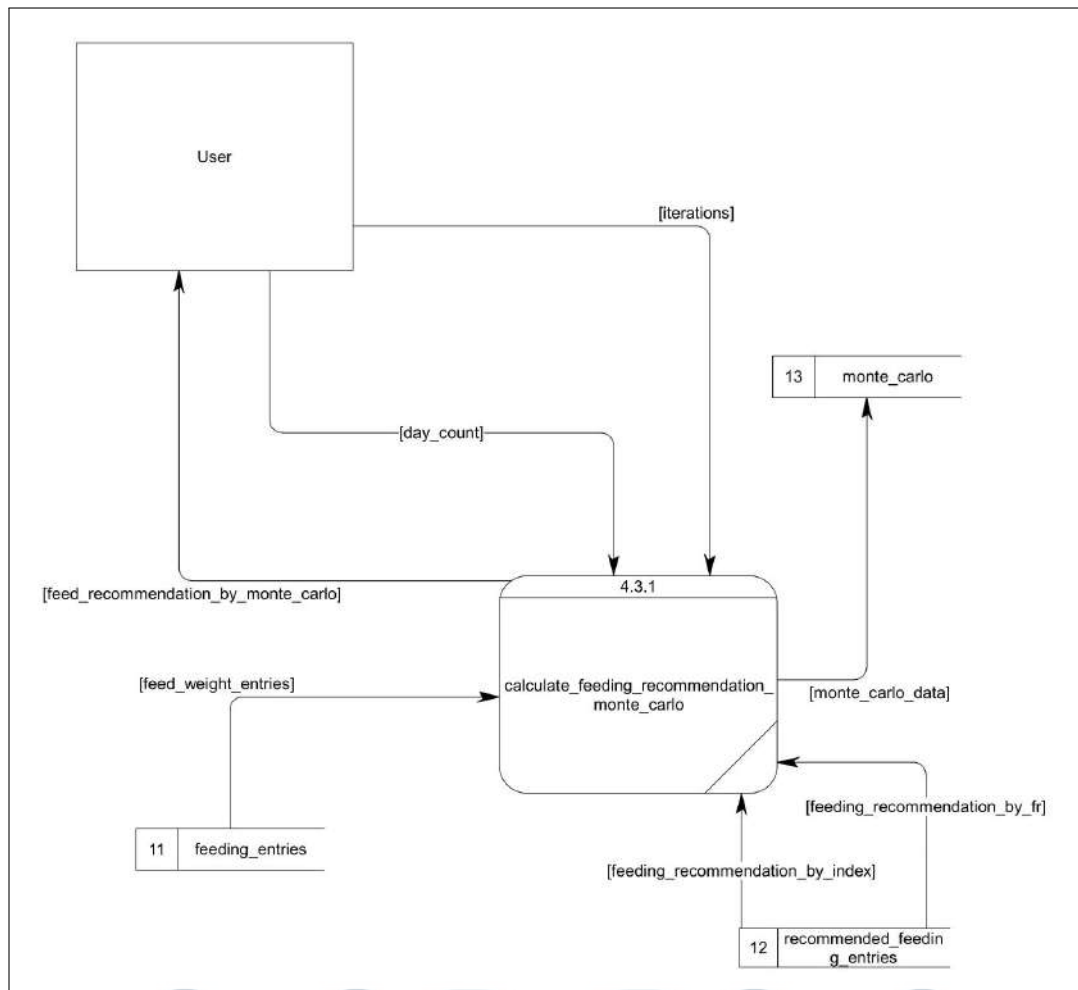
output berupa angka hasil perhitungan rekomendasi pakan berdasarkan *index*, *fr*, dan *monte carlo*.



Gambar 3.14. DFD Level 3 *Feeding Recommendation By Index and FR*

Gambar 3.14 menjabarkan proses *calculate recommendation by fr and index*. Proses tersebut dibagi menjadi dua buah subproses yaitu perhitungan rekomendasi berdasarkan *index* dan berdasarkan FR. Kedua perhitungan tersebut menggunakan Rumus 2.2 dan Rumus 2.4 yang telah dijabarkan pada Bab 2.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

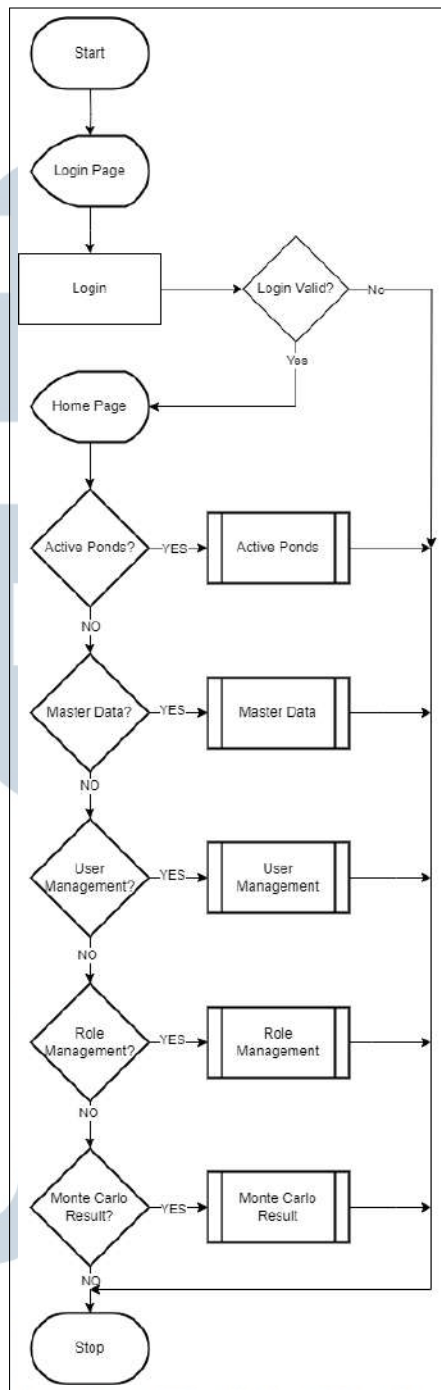


Gambar 3.15. DFD Level 3 *Feeding Recommendation By Monte Carlo*

Gambar 3.15 menjabarkan proses *calculate recommendation by monte carlo*. Pada perhitungan monte carlo digunakan beberapa data antara lain *feeding recommendation by index* dan *feeding recommendation by fr* dari Tabel Recommended Feeding Entries, *feed weight* dari Tabel Feeding Entries, serta jumlah hari dan iterasi dari pengguna. Rumus perhitungan rekomendasi berdasarkan monte carlo dijabarkan pada Bab 2 pada Rumus 2.5 hingga Rumus 2.7.

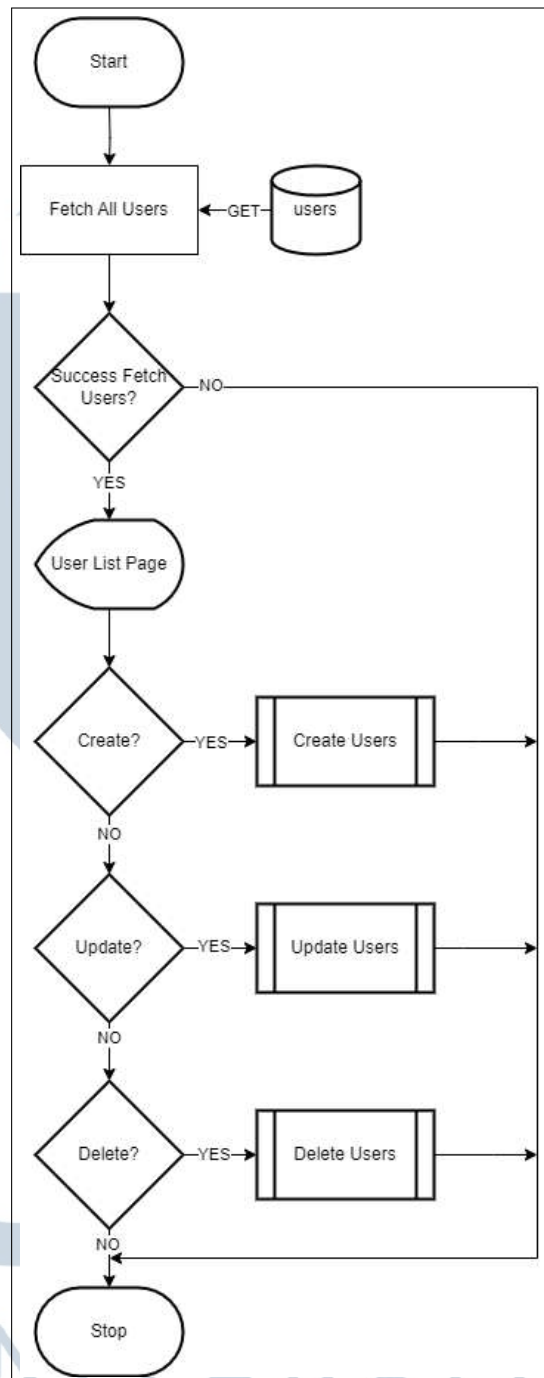
3.3.2 Flowchart Sistem

Alur kerja sistem ERP yang dibangun dibagi menjadi beberapa bagian yang meliputi *Login*, *Active Ponds*, *User Management*, *Role Management*, *Master Data*, dan *Monte Carlo*. Setiap bagian akan dijabarkan sebagai berikut.



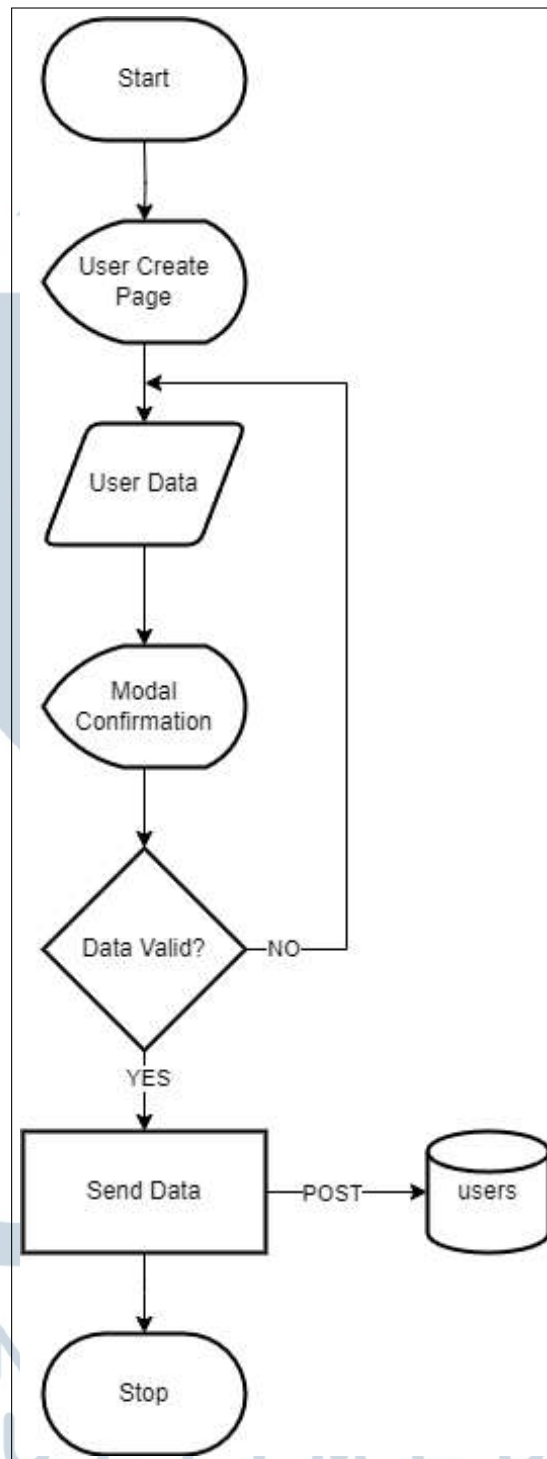
Gambar 3.16. General App Flowchart

Gambar 3.16 menggambarkan alur sistem ERP secara keseluruhan. Berdasarkan alur kerja tersebut, pengguna diharuskan untuk melakukan *login* terlebih dahulu sebelum dapat menggunakan fitur yang terdapat di dalam sistem. Selain itu pengguna juga harus menggunakan *credentials* yang valid dan sesuai dengan yang terdaftar pada *database*.



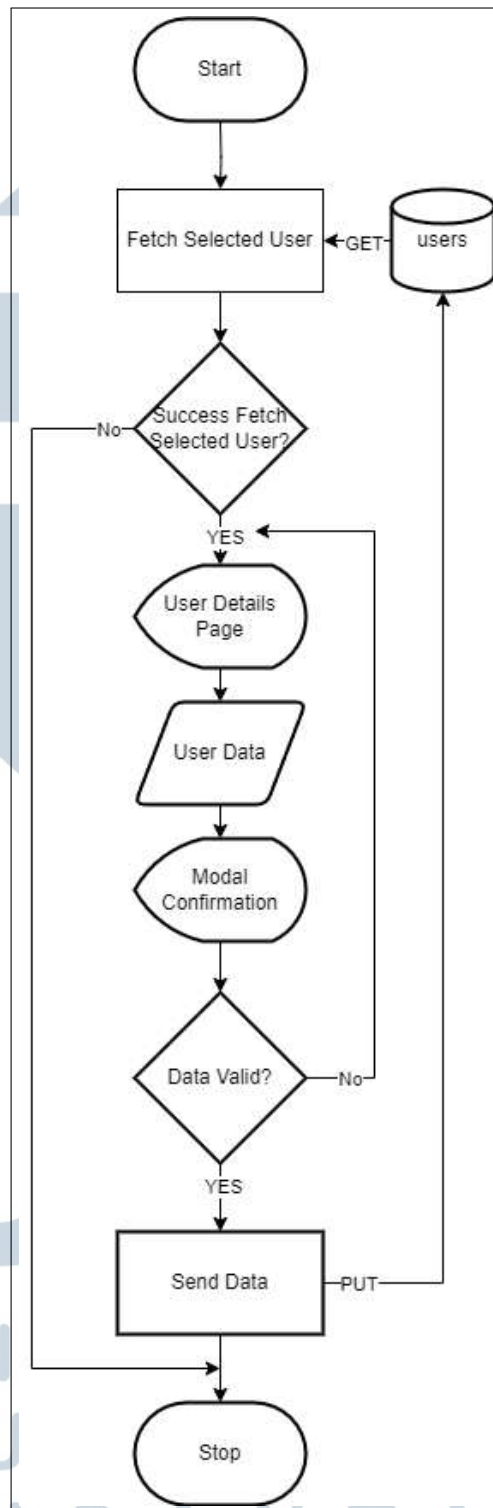
Gambar 3.17. *User Management Flowchart*

Gambar 3.17 menggambarkan alur kerja pada proses *user management*. Pada awal proses tersebut, akan ditampilkan daftar *user* yang terdapat di dalam *database*. Selain itu *user* dapat menambahkan *user* baru, menyunting *user* dari *list user* yang ada, dan menghapus *user*.



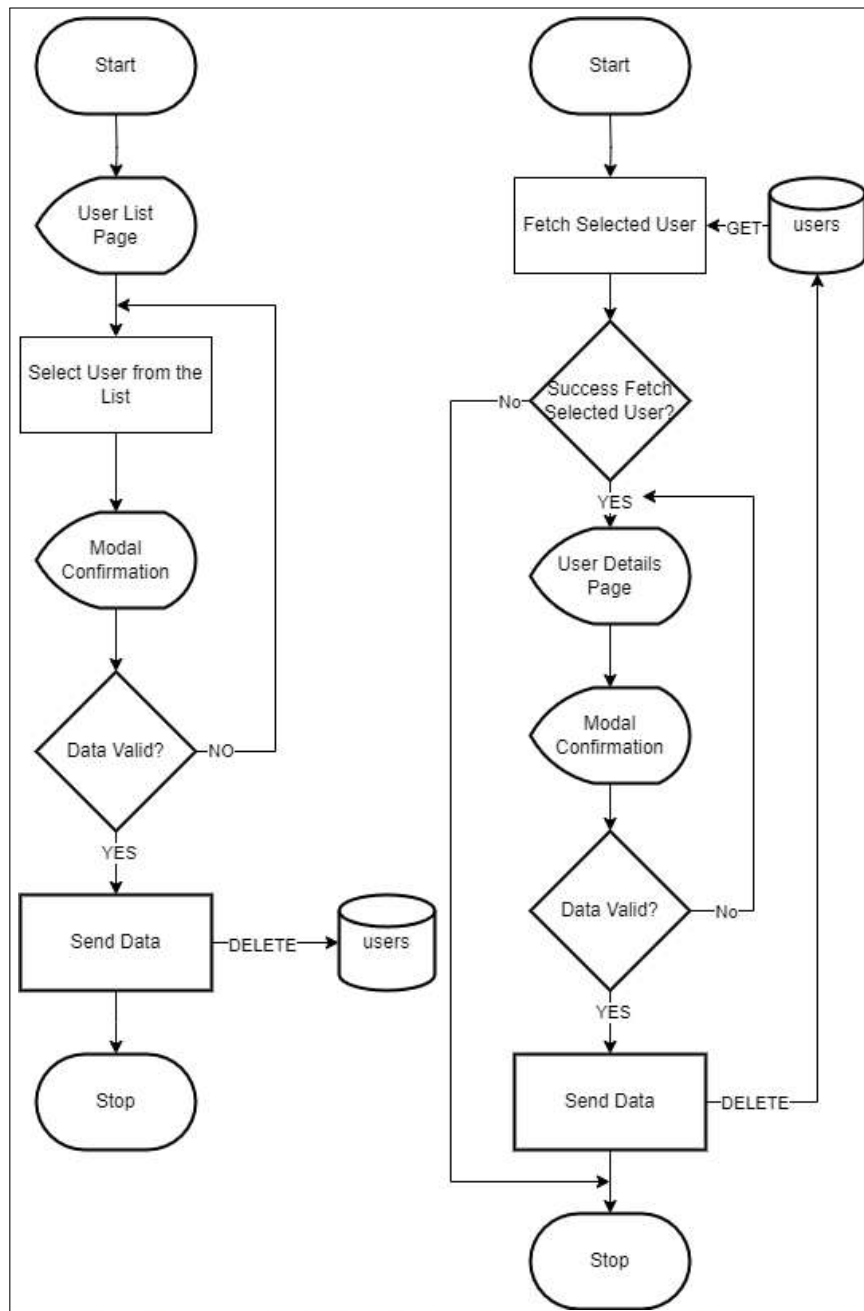
Gambar 3.18. *User Management Create Flowchart*

Gambar 3.18 menggambarkan alur kerja proses pembuatan *user* baru. Saat membuat *user* baru perlu memasukkan beberapa data yaitu *username*, *password*, dan *role*. Apabila ketiga data tersebut sudah diisi dan valid, maka akan disimpan ke dalam *database*.



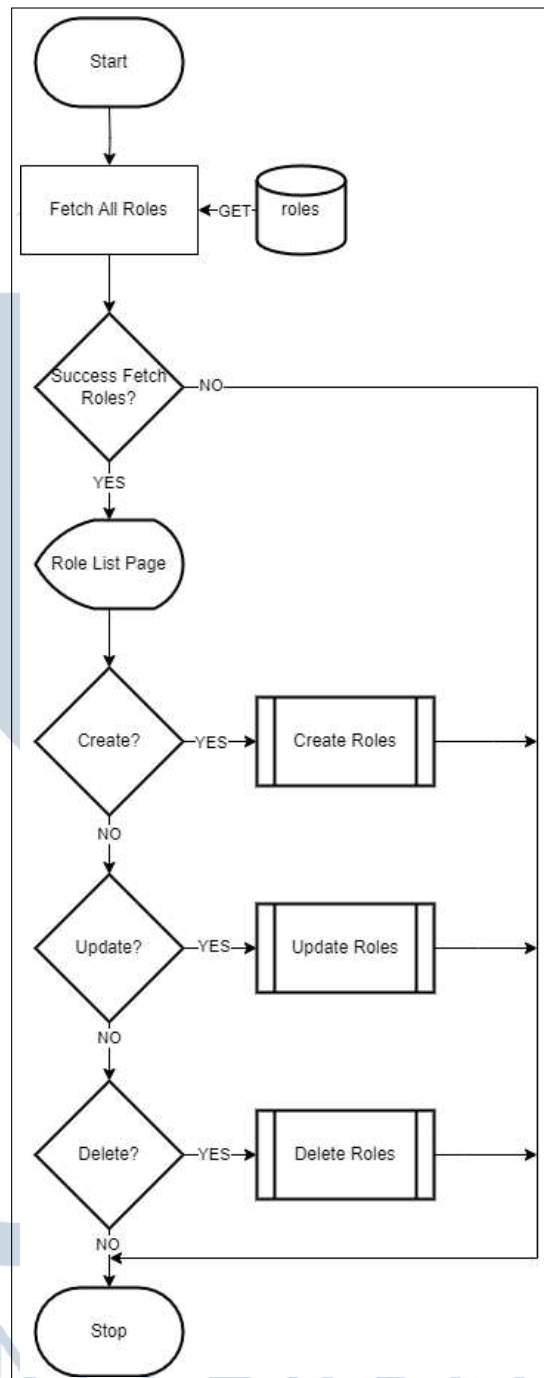
Gambar 3.19. *User Management Edit Flowchart*

Gambar 3.19 menggambarkan alur kerja proses *edit user* yang terdapat dalam *database*. Pada *edit user*, hanya dapat mengganti *role* dari *user* tersebut.



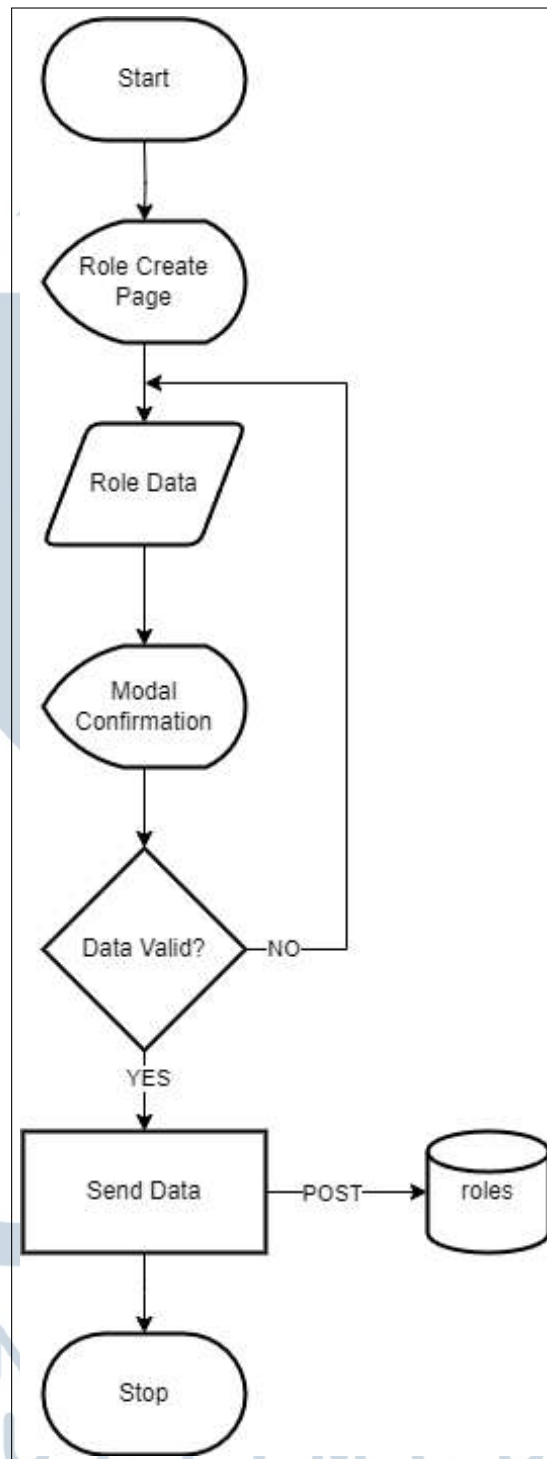
Gambar 3.20. User Management Delete Flowchart

Gambar 3.20 menggambarkan alur kerja proses menghapus *user* yang terdapat dalam *database*. Pengguna dapat menghapus data yang diinginkan dengan dua cara. Cara pertama pada *flowchart* kiri adalah dengan memilih data yang ingin dihapus dari *list data*. Sedangkan pada *flowchart* kanan adalah dengan masuk ke dalam halaman *edit* dan menghapusnya.



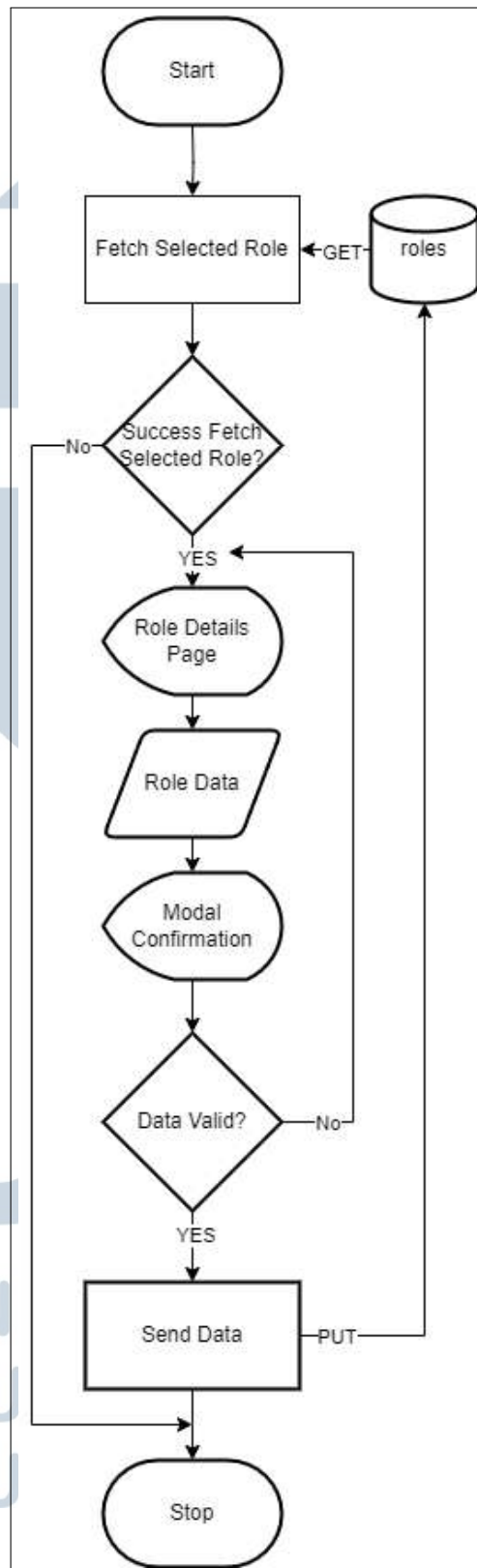
Gambar 3.21. *Role Management Flowchart*

Gambar 3.21 menggambarkan alur kerja pada proses *role management*. Pada awal proses tersebut, akan ditampilkan daftar *role* yang terdapat di dalam *database*. Selain itu *user* dapat menambahkan *role* baru, menyunting *role* dari *list role* yang ada, dan menghapus *role*.



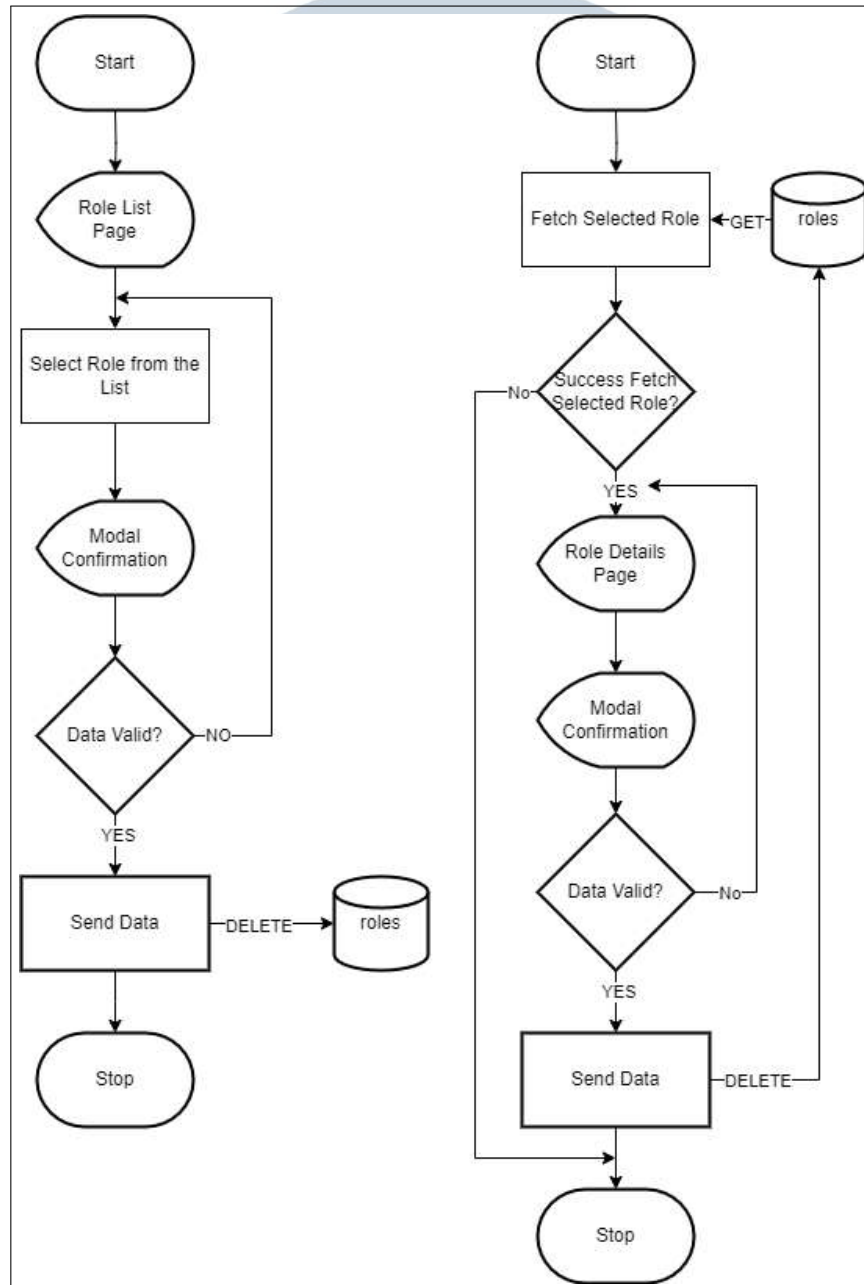
Gambar 3.22. Role Management Create Flowchart

Gambar 3.22 menggambarkan alur kerja proses pembuatan *role* baru. Saat membuat *role* baru perlu memasukkan data yaitu *role name*. Apabila data tersebut sudah diisi dan valid, maka akan disimpan ke dalam *database*.



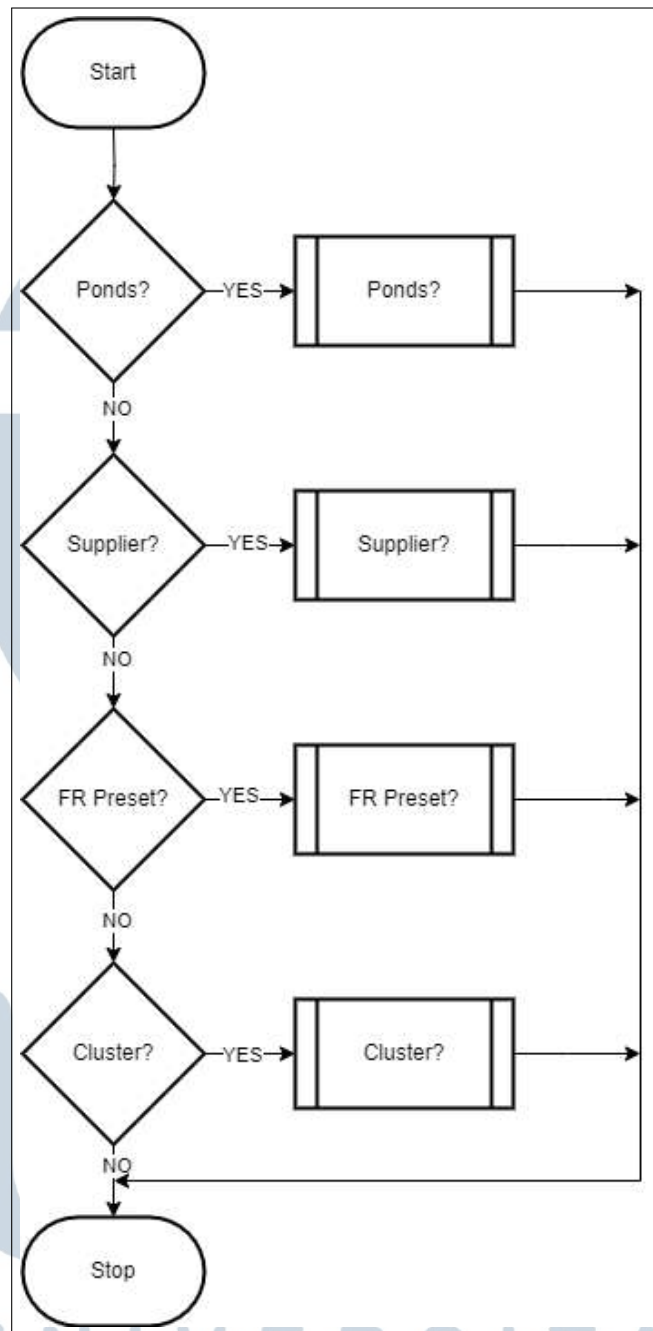
Gambar 3.23. Role Management Edit Flowchart

Gambar 3.23 menggambarkan alur kerja proses *edit role* yang terdapat dalam *database*.



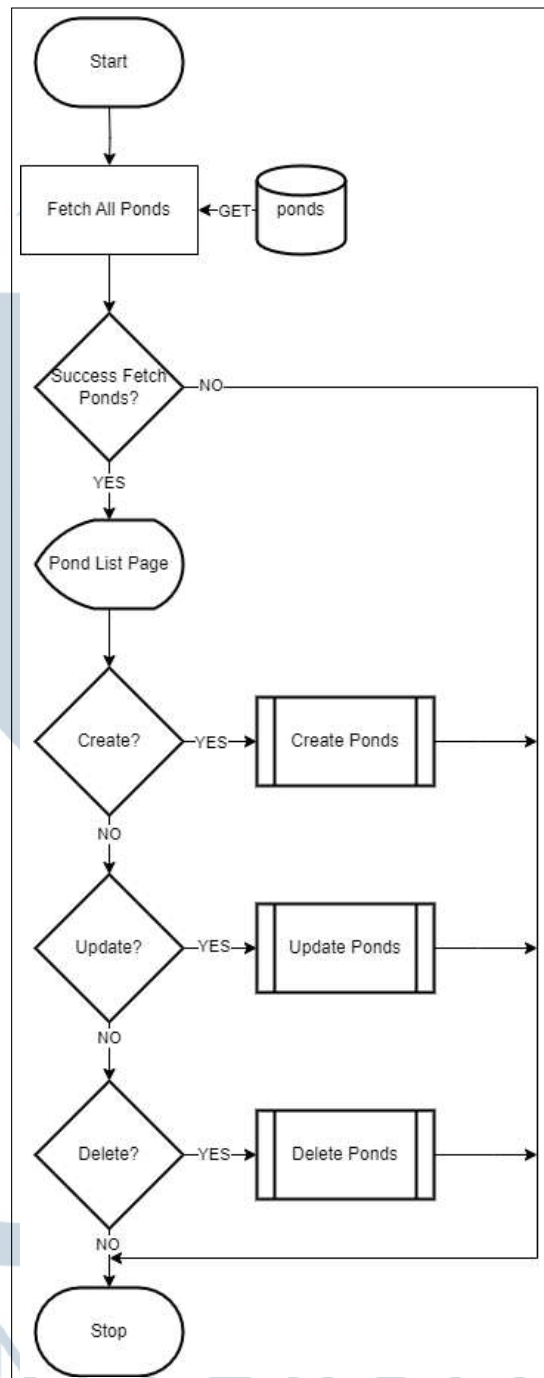
Gambar 3.24. Role Management Delete Flowchart

Gambar 3.24 menggambarkan alur kerja proses menghapus *role* yang terdapat dalam *database*. Pengguna dapat menghapus data yang diinginkan dengan dua cara. Cara pertama pada *flowchart* kiri adalah dengan memilih data yang ingin dihapus dari *list data*. Sedangkan pada *flowchart* kanan adalah dengan masuk ke dalam halaman *edit* dan menghapusnya.



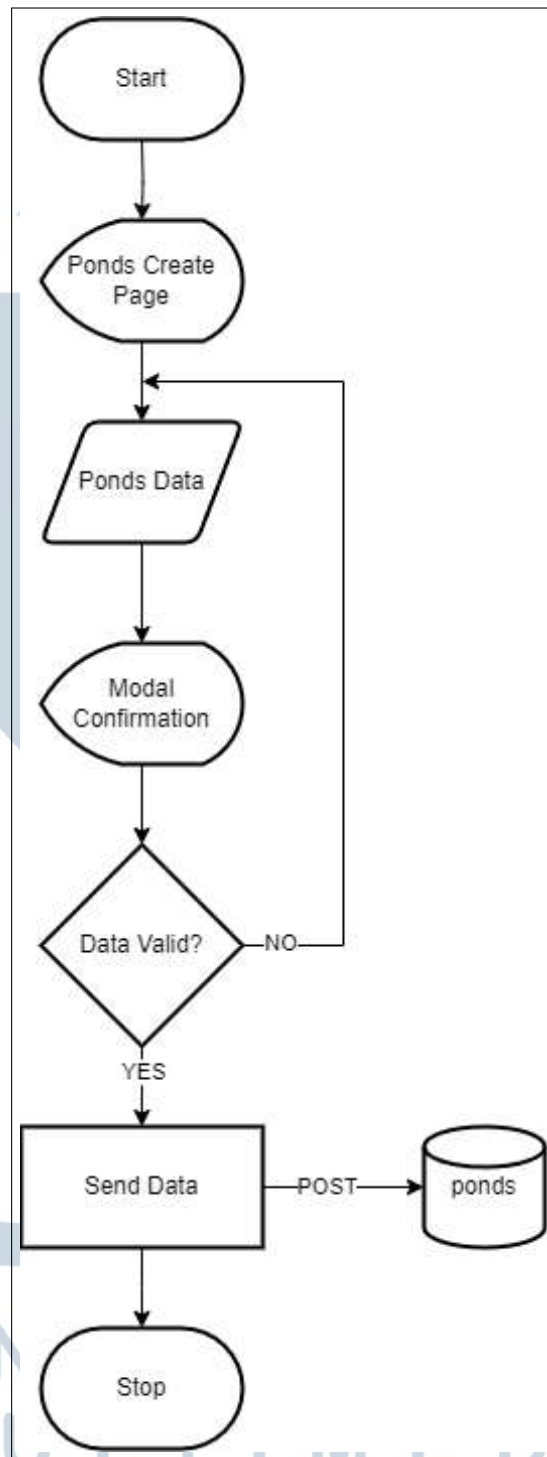
Gambar 3.25. Master Data General Flowchart

Gambar 3.25 menggambarkan garis besar alur kerja *master data*. *Master Data* dibagi menjadi empat yaitu *ponds*, *clusters*, *suppliers*, dan *FR Preset*.



Gambar 3.26. *Ponds Master Data Flowchart*

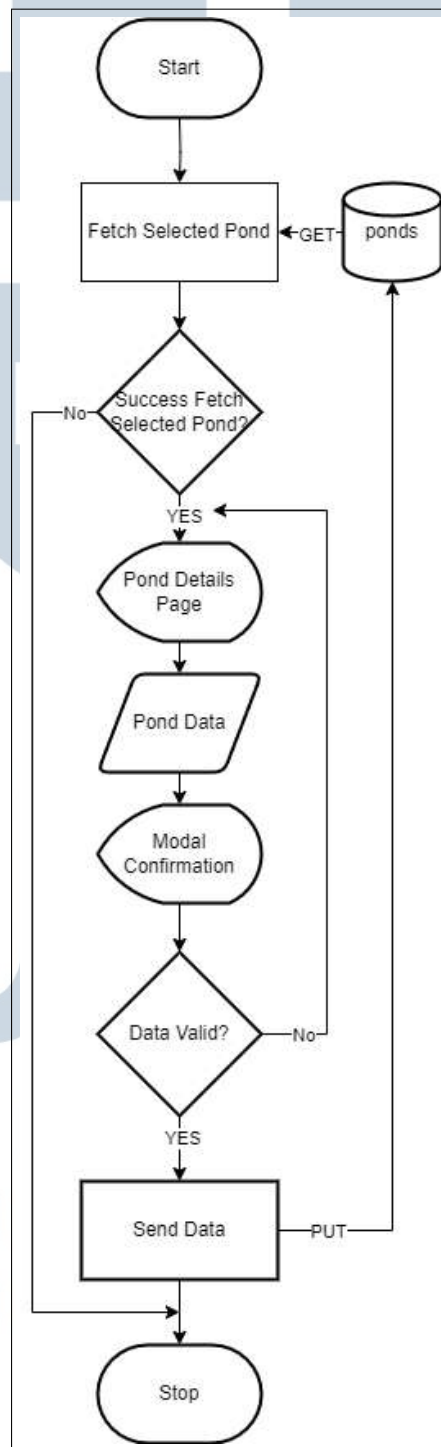
Gambar 3.26 menggambarkan alur kerja pada proses *master data ponds*. Pada awal proses tersebut, akan ditampilkan daftar *ponds* yang terdapat di dalam *database*. Selain itu *user* dapat menambahkan *ponds* baru, menyunting *ponds* dari *list ponds* yang ada, dan menghapus *ponds*.



Gambar 3.27. *Ponds Master Data Create Flowchart*

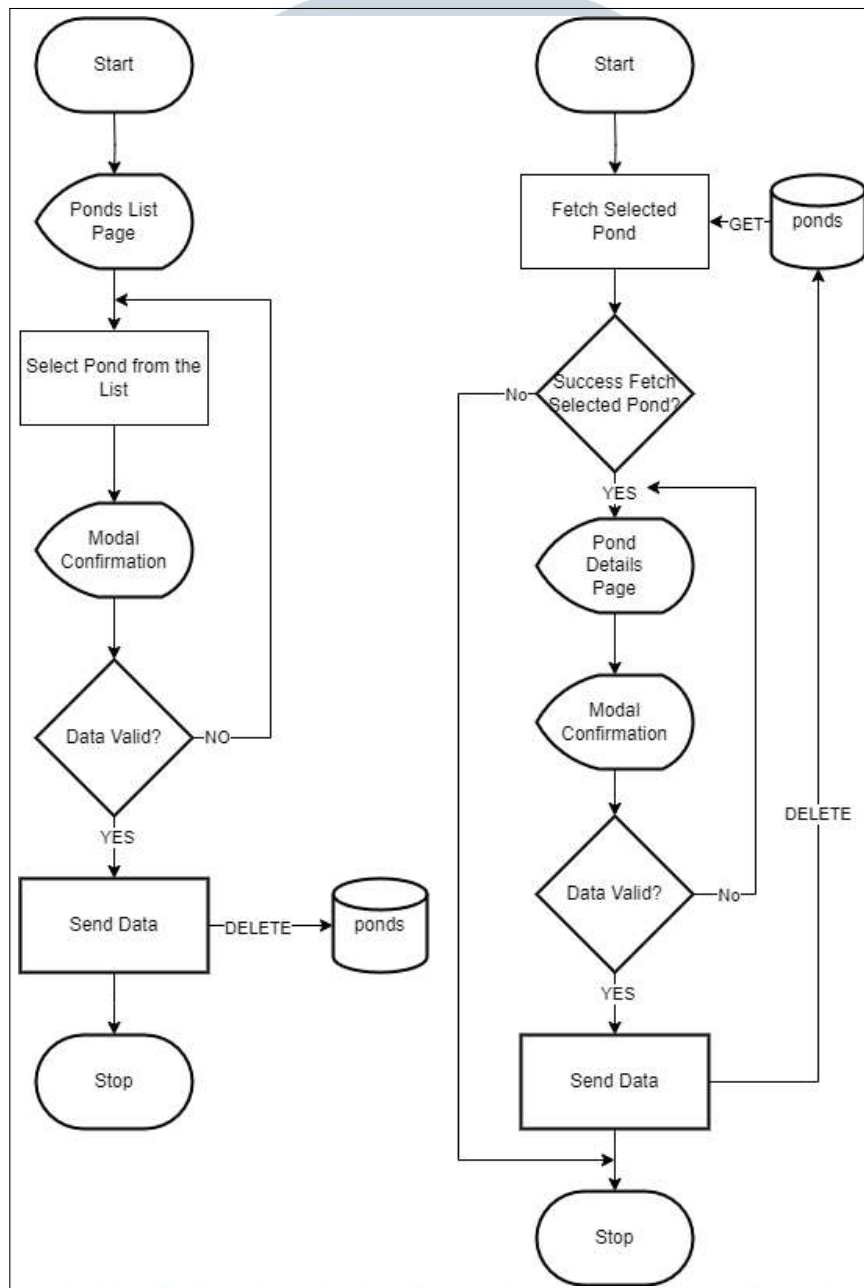
Gambar 3.27 menggambarkan alur kerja proses pembuatan *ponds* baru. Membuat *ponds* baru terdapat dua cara yaitu, *single* dan *mutliple*. Alur pembuatan dan data yang digunakan keduanya sama saja, yang membedakan adalah apabila *multiple* akan membuat beberapa *ponds* dalam satu waktu yang sama. Saat

membuat *ponds* baru perlu memasukkan beberapa data yaitu *pond code*, *pond space*, *pond depth*, *pond water height*. Untuk *pond volume* merupakan hitungan dari *pond space* dikali dengan *pond water height*. Apabila data tersebut sudah diisi dan valid, maka akan disimpan ke dalam *database*.



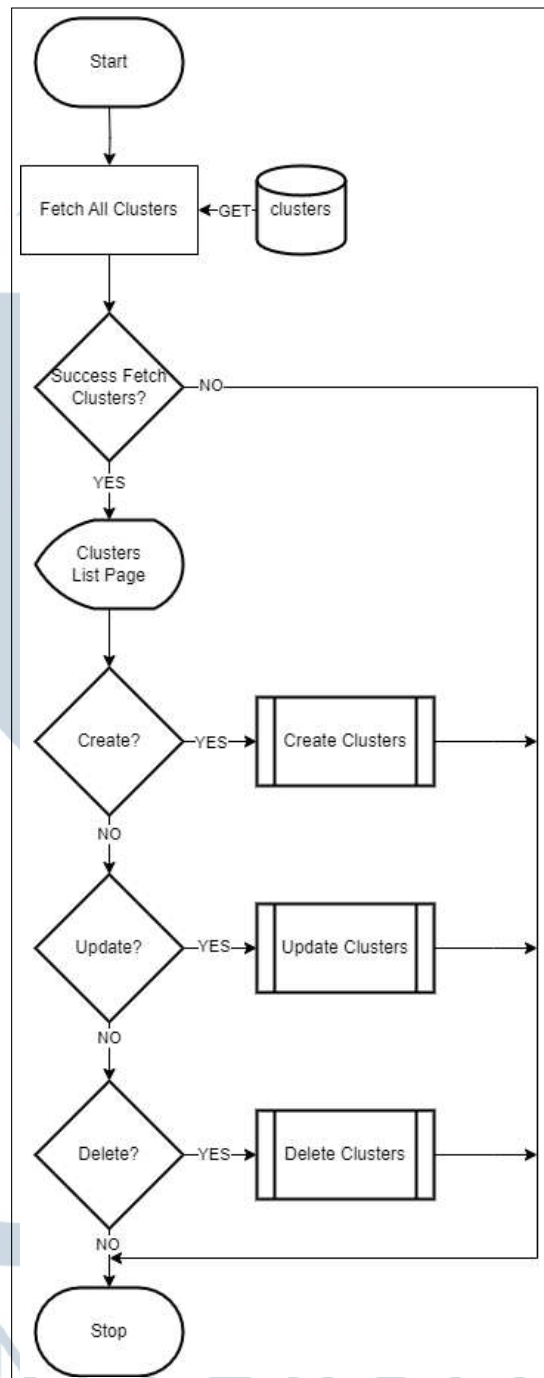
Gambar 3.28. *Ponds Master Data Edit Flowchart*

Gambar 3.28 menggambarkan alur kerja proses *edit pond* yang terdapat dalam *database*. *Edit pond* hanya dapat dilakukan satu per satu untuk setiap *pond*.



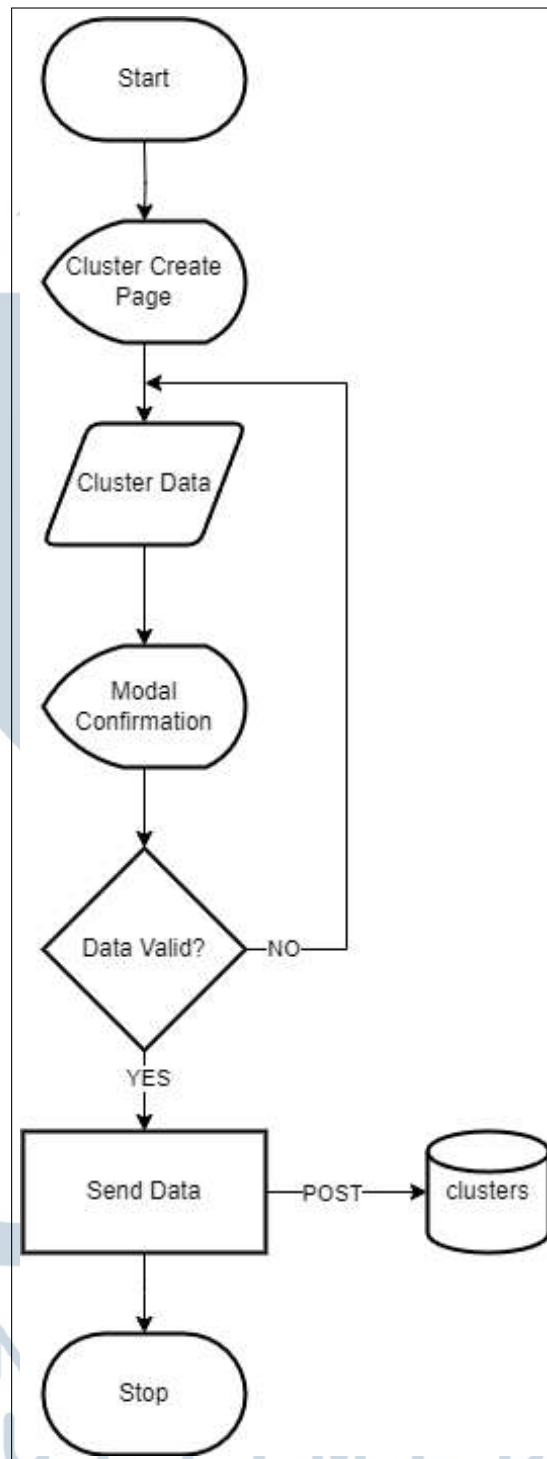
Gambar 3.29. *Ponds Master Data Delete Flowchart*

Gambar 3.29 menggambarkan alur kerja proses menghapus *ponds* yang terdapat dalam *database*. Pengguna dapat menghapus data yang diinginkan dengan dua cara. Cara pertama pada *flowchart* kiri adalah dengan memilih data yang ingin dihapus dari *list data*. Sedangkan pada *flowchart* kanan adalah dengan masuk ke dalam halaman *edit* dan menghapusnya.



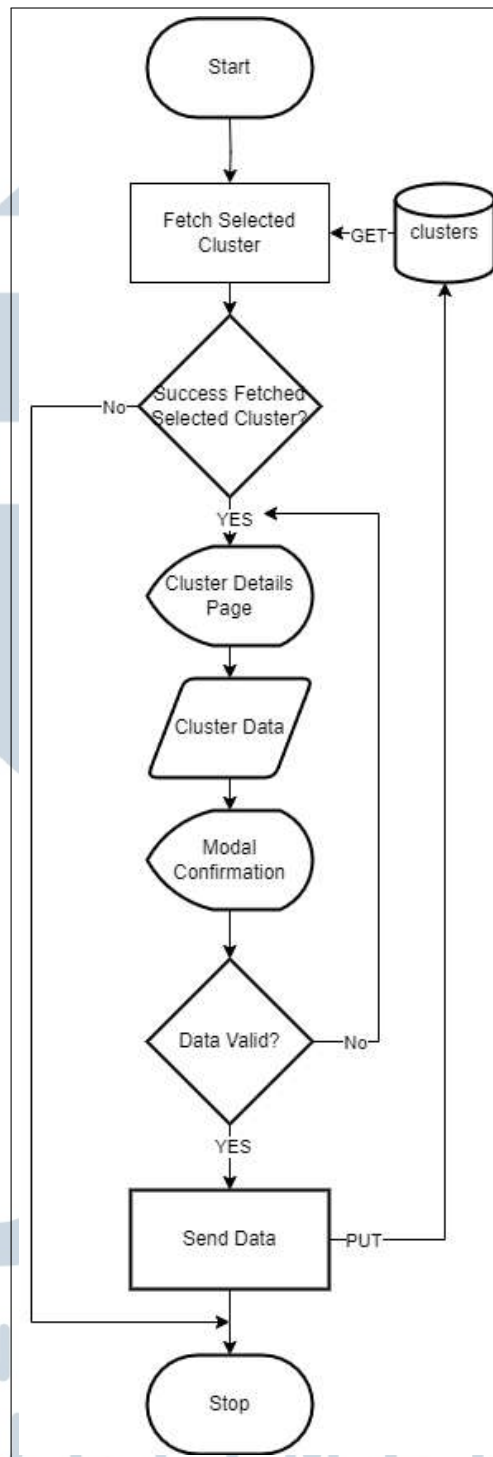
Gambar 3.30. *Clusters Master Data Flowchart*

Gambar 3.30 menggambarkan alur kerja pada proses *master data cluster*. Pada awal proses tersebut, akan ditampilkan daftar *cluster* yang terdapat di dalam *database*. Selain itu *user* dapat menambahkan *cluster* baru, menyunting *cluster* dari *list cluster* yang ada, dan menghapus *cluster*.



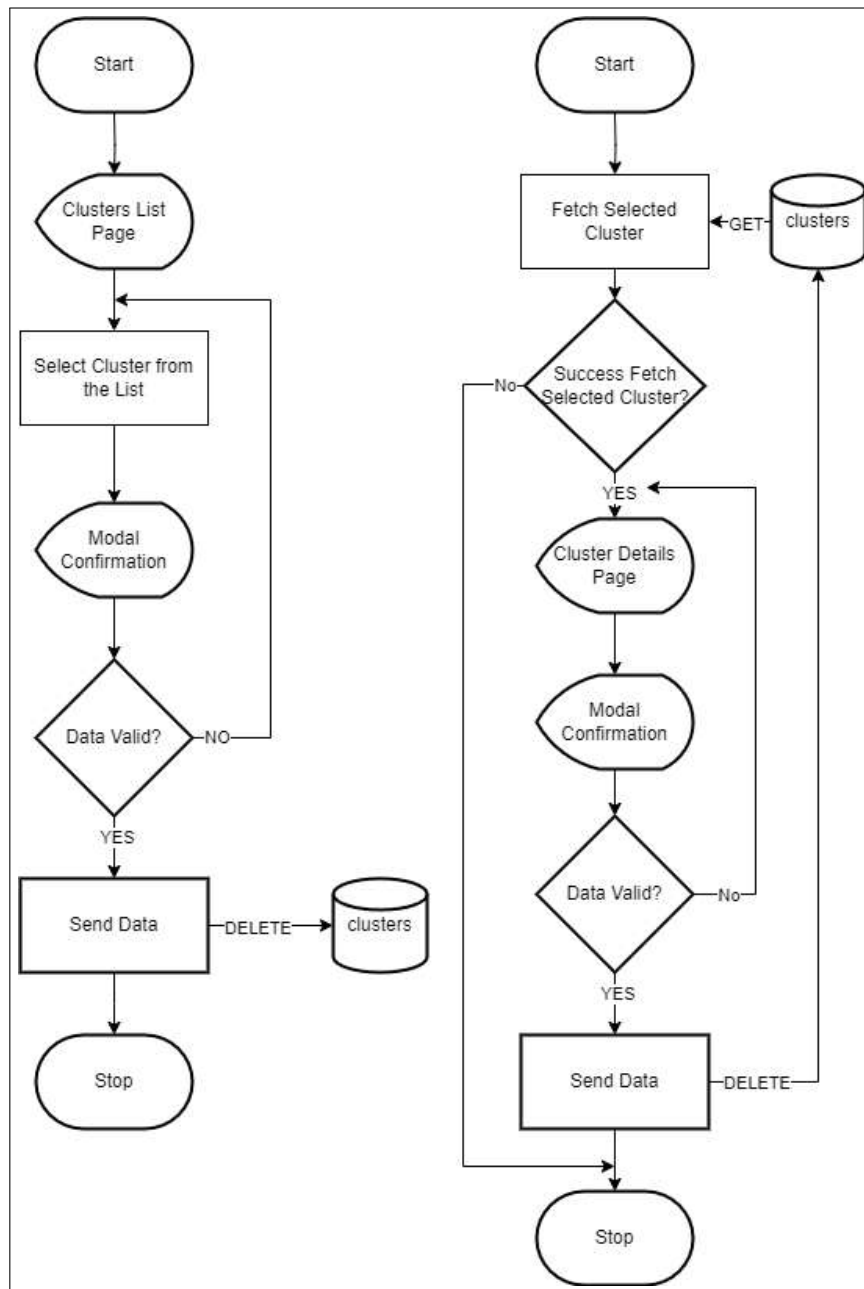
Gambar 3.31. *Clusters Master Data Create Flowchart*

Gambar 3.31 menggambarkan alur kerja proses pembuatan *clusters* baru. Saat membuat *clusters* baru perlu memasukkan beberapa data yaitu *cluster name* dan *ponds* yang diinginkan. Apabila data tersebut sudah diisi dan valid, maka akan disimpan ke dalam *database*.



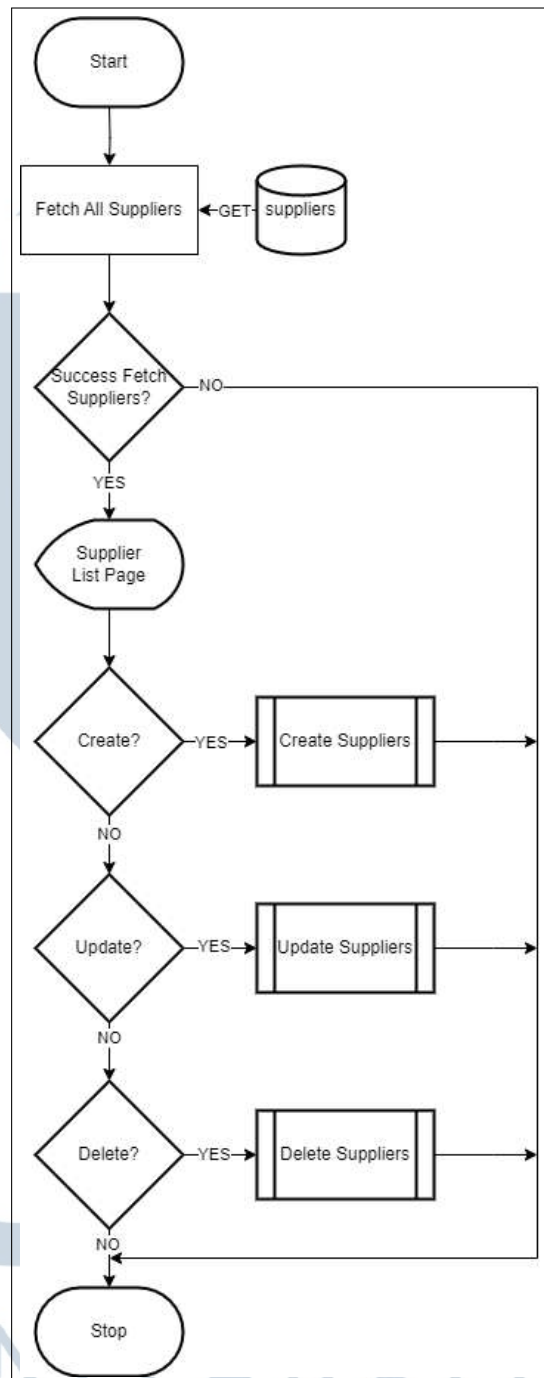
Gambar 3.32. Clusters Master Data Edit Flowchart

Gambar 3.32 menggambarkan alur kerja proses *edit clusters* yang terdapat dalam *database*. Data yang dapat di-*edit* sama dengan data yang dimasukkan saat pembuatan *clusters* baru.



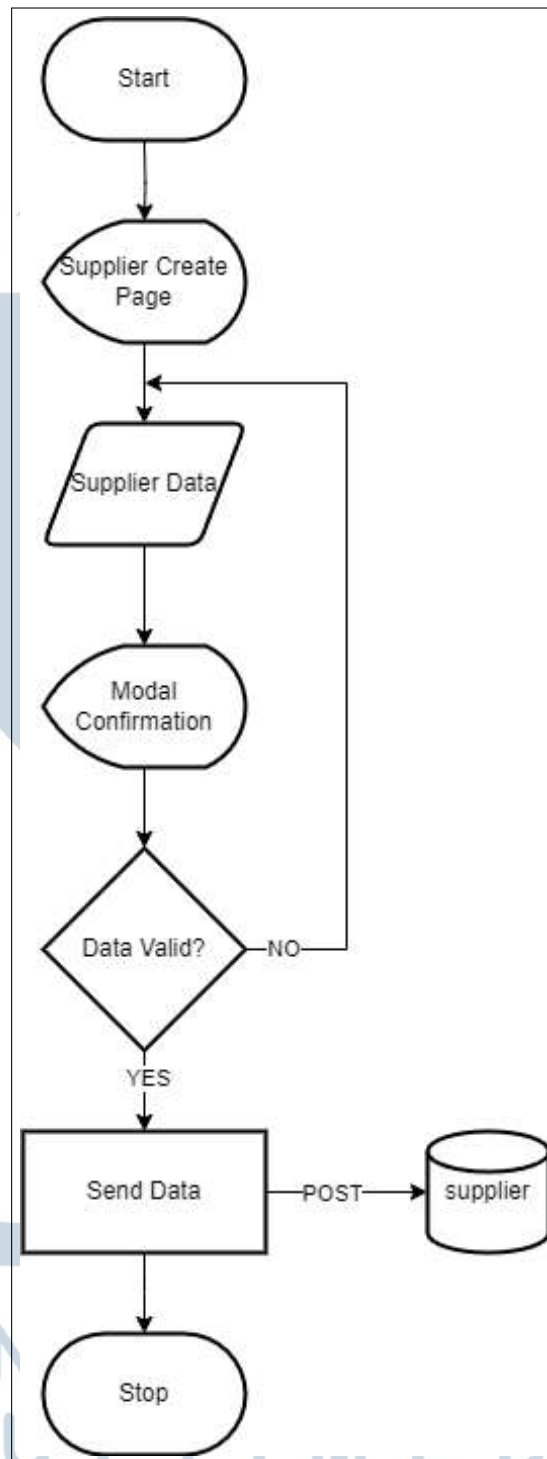
Gambar 3.33. Clusters Master Data Delete Flowchart

Gambar 3.33 menggambarkan alur kerja proses menghapus *clusters* yang terdapat dalam *database*. Pengguna dapat menghapus data yang diinginkan dengan dua cara. Cara pertama pada *flowchart* kiri adalah dengan memilih data yang ingin dihapus dari *list data*. Sedangkan pada *flowchart* kanan adalah dengan masuk ke dalam halaman *edit* dan menghapusnya.



Gambar 3.34. *Suppliers Master Data Flowchart*

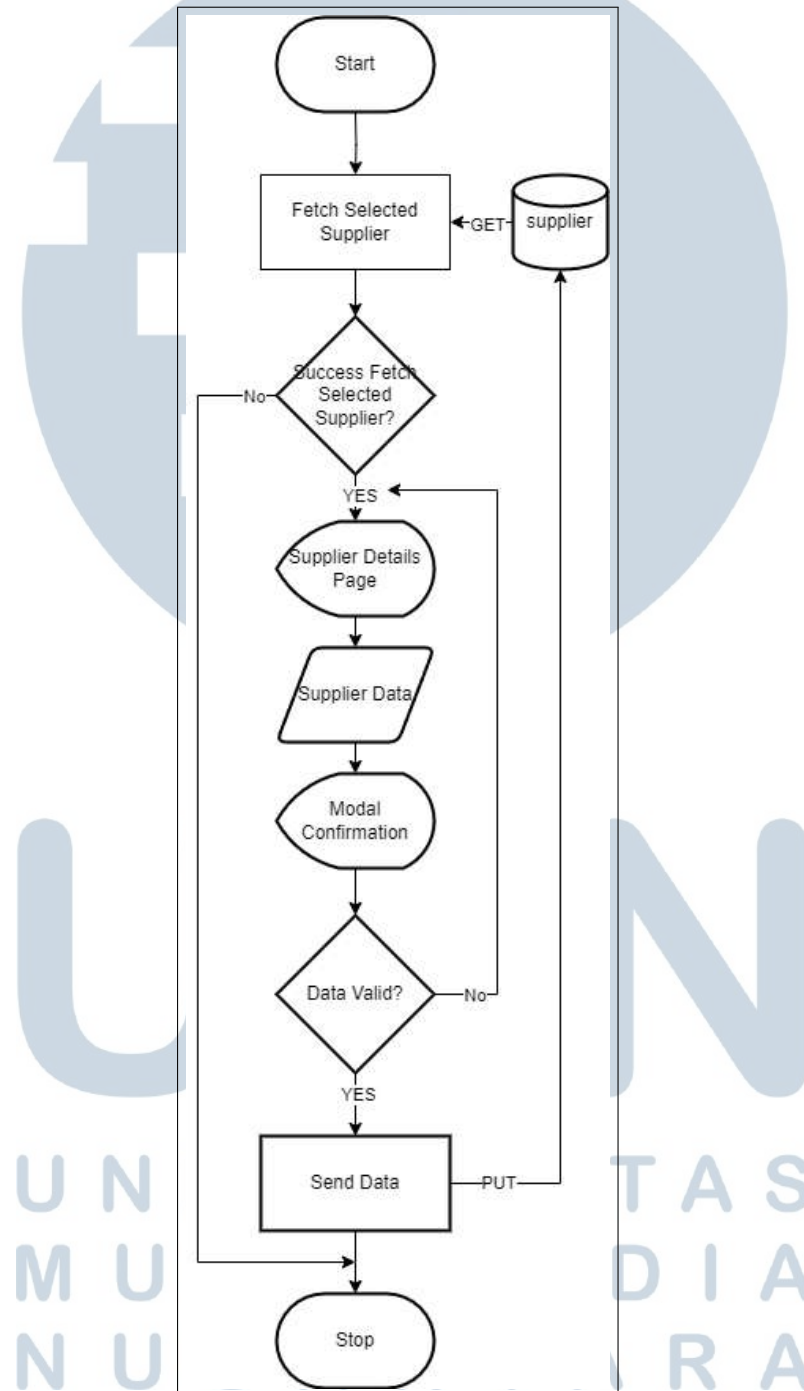
Gambar 3.34 menggambarkan alur kerja pada proses *master data suppliers*. Pada awal proses tersebut, akan ditampilkan daftar *suppliers* yang terdapat di dalam *database*. Selain itu *user* dapat menambahkan *suppliers* baru, menyunting *suppliers* dari *list suppliers* yang ada, dan menghapus *suppliers*.



Gambar 3.35. *Suppliers Master Data Create Flowchart*

Gambar 3.35 menggambarkan alur kerja proses pembuatan *suppliers* baru. Membuat *suppliers* baru terdapat dua cara yaitu, *single* dan *multiple*. Alur pembuatan dan data yang digunakan keduanya sama saja, yang membedakan adalah apabila *multiple* akan membuat beberapa *suppliers* dalam satu waktu yang sama.

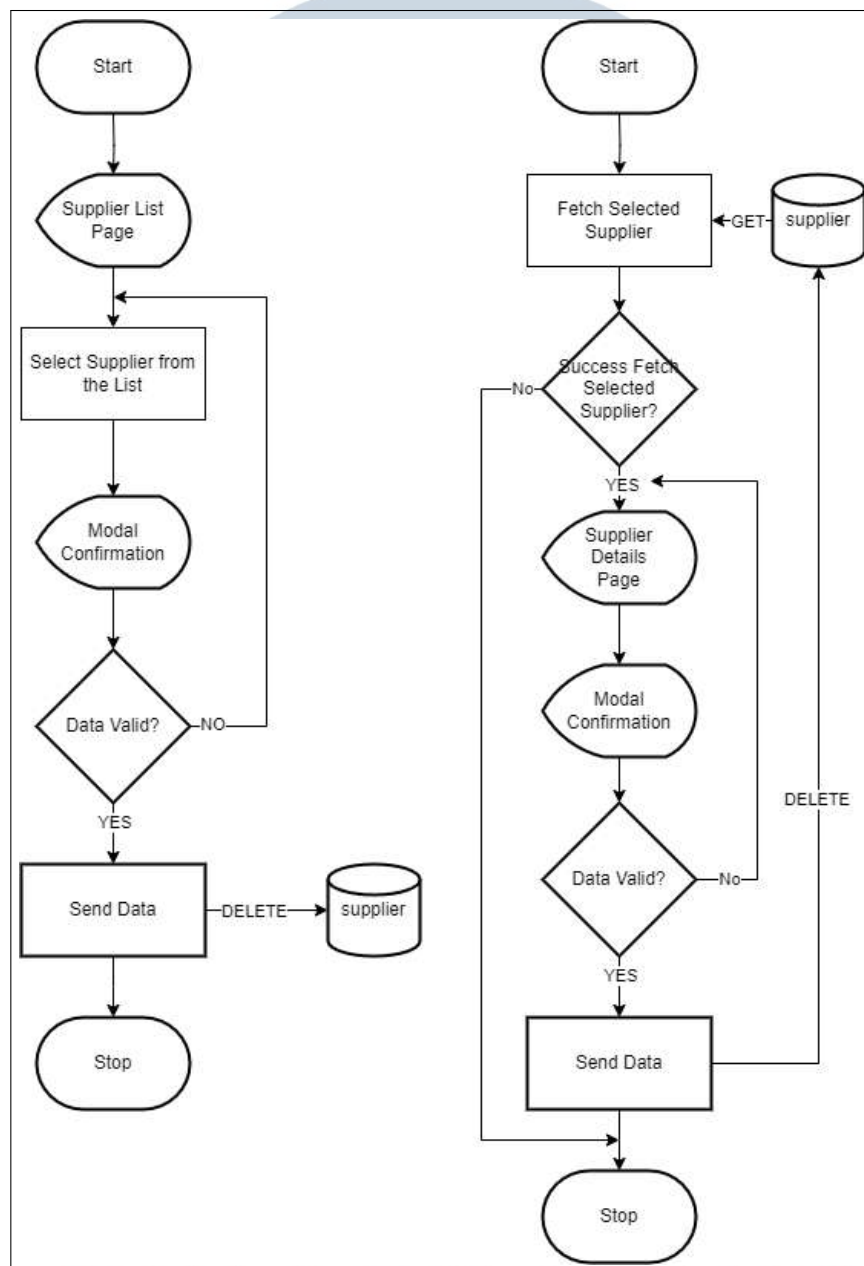
Saat membuat *suppliers* baru perlu memasukkan beberapa data yaitu *supplier name*, *location*, dan *supplier type* yang diinginkan. Apabila data tersebut sudah diisi dan valid, maka akan disimpan ke dalam *database*.



Gambar 3.36. *Suppliers Master Data Edit Flowchart*

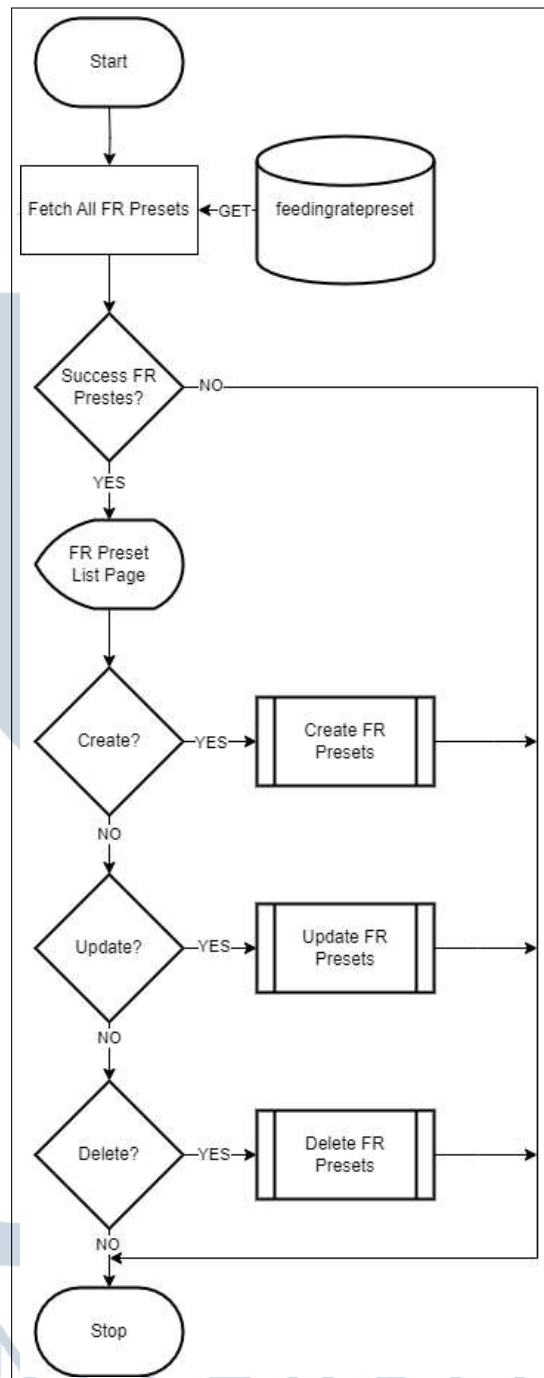
Gambar 3.36 menggambarkan alur kerja proses *edit suppliers* yang terdapat dalam *database*. Data yang dapat di-*edit* sama dengan data yang dimasukkan saat

pembuatan *suppliers* baru. *Edit supplier* hanya dapat dilakukan satu per satu untuk setiap *supplier*.



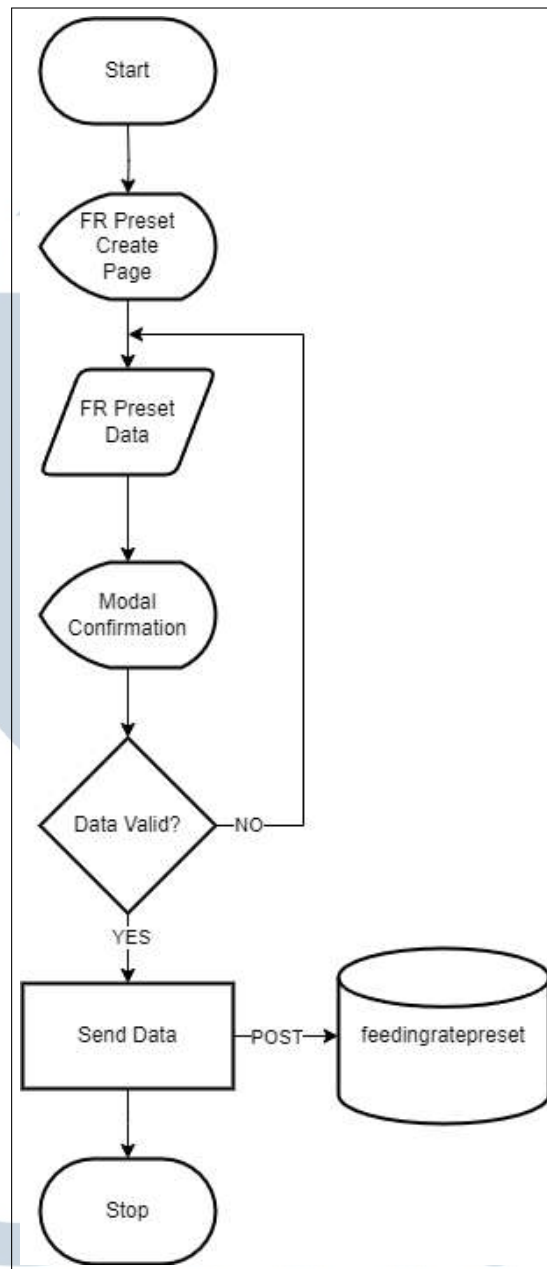
Gambar 3.37. *Suppliers Master Data Delete Flowchart*

Gambar 3.37 menggambarkan alur kerja proses menghapus *suppliers* yang terdapat dalam *database*. Pengguna dapat menghapus data yang diinginkan dengan dua cara. Cara pertama pada *flowchart* kiri adalah dengan memilih data yang ingin dihapus dari *list data*. Sedangkan pada *flowchart* kanan adalah dengan masuk ke dalam halaman *edit* dan menghapusnya.



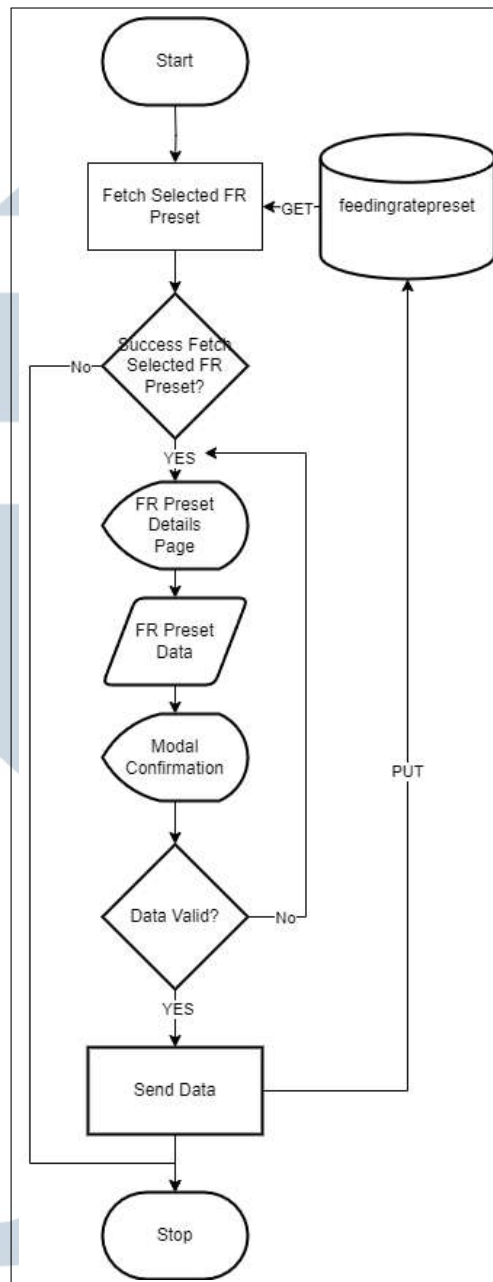
Gambar 3.38. *FR Preset Master Data Flowchart*

Gambar 3.38 menggambarkan alur kerja pada proses *master data FR preset*. Pada awal proses tersebut, akan ditampilkan daftar *FR preset* yang terdapat di dalam *database*. Selain itu *user* dapat menambahkan *FR preset* baru, menyunting *FR preset* dari *list FR preset* yang ada, dan menghapus *FR preset*.



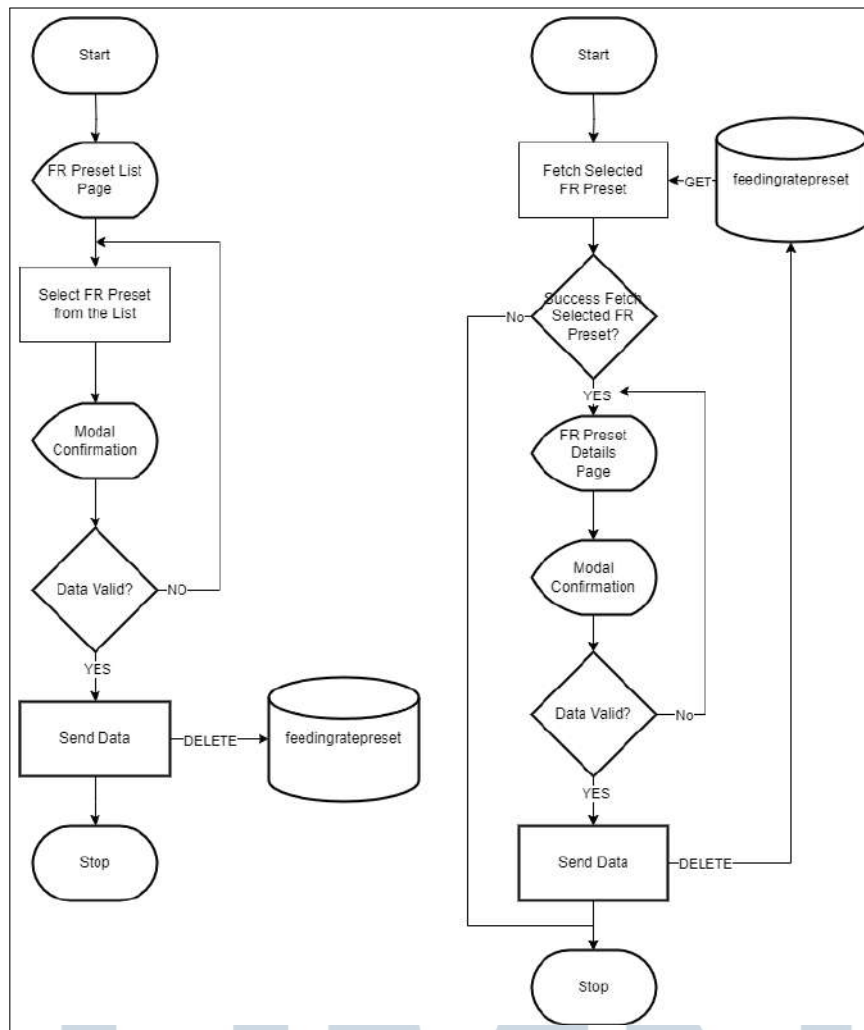
Gambar 3.39. *FR Preset Master Data Create Flowchart*

Gambar 3.39 menggambarkan alur kerja proses pembuatan *FR presets* baru. Saat membuat *FR Presets* baru perlu memasukkan beberapa data yaitu *preset name* dan *FR Parameters* yang diinginkan. *FR Parameters* terdiri dari beberapa data yaitu DoC, MBW, dan *FR Percentage*. Apabila data tersebut sudah diisi dan valid, maka akan disimpan ke dalam *database*.



Gambar 3.40. *FR Preset Master Data Edit Flowchart*

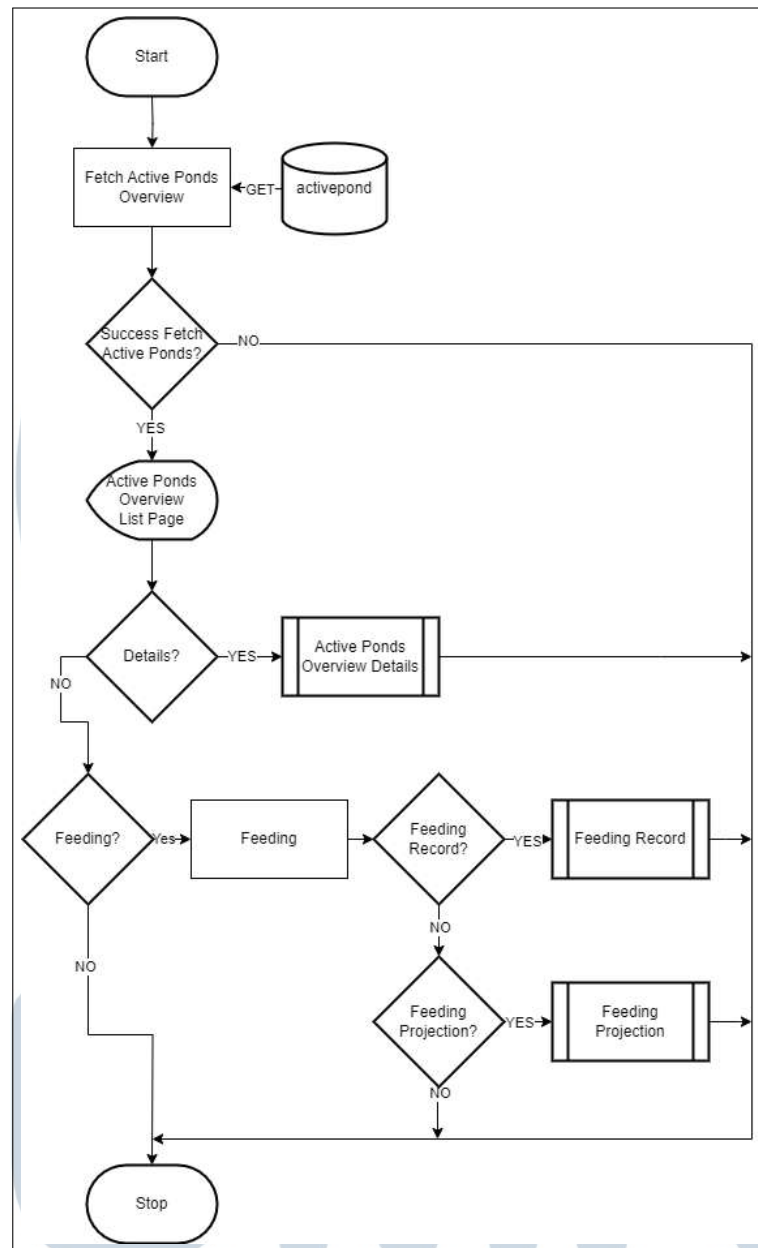
Gambar 3.40 menggambarkan alur kerja proses *edit FR presets* yang terdapat dalam *database*. Data yang dapat di-*edit* sama dengan data yang dimasukkan saat pembuatan *FR presets* baru.



Gambar 3.41. *FR Preset Master Data Delete Flowchart*

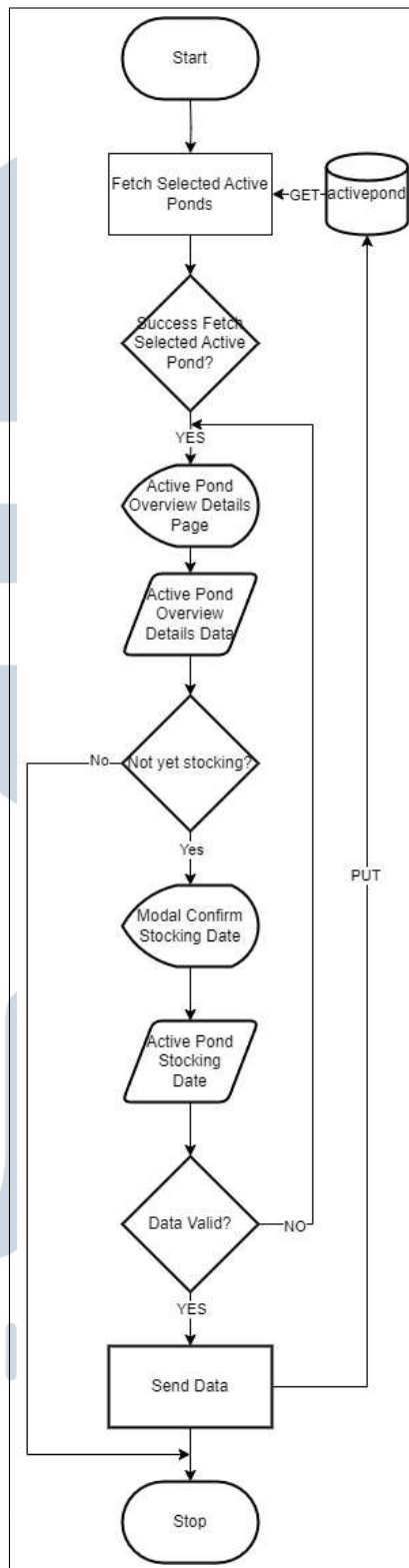
Gambar 3.41 menggambarkan alur kerja proses menghapus *FR Presets* yang terdapat dalam *database*. Pengguna dapat menghapus data yang diinginkan dengan dua cara. Cara pertama pada *flowchart* kiri adalah dengan memilih data yang ingin dihapus dari *list data*. Sedangkan pada *flowchart* kanan adalah dengan masuk ke dalam halaman *edit* dan menghapusnya.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



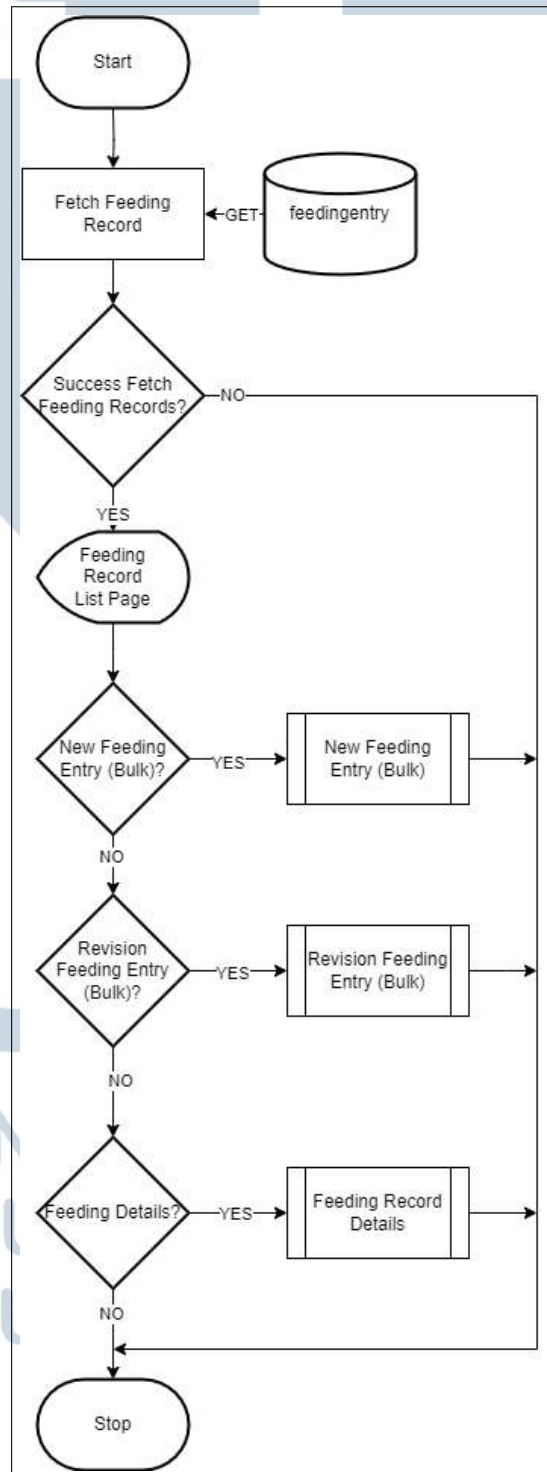
Gambar 3.42. *Active Ponds Flowchart*

Gambar 3.42 menggambarkan alur kerja pada proses *active ponds*. Pada awal proses tersebut, akan ditampilkan *list active ponds* yang terdapat di dalam *database*. Selain itu *user* dapat melihat detail dari salah satu *active ponds*, *feeding record*, dan *feeding recommendation*.



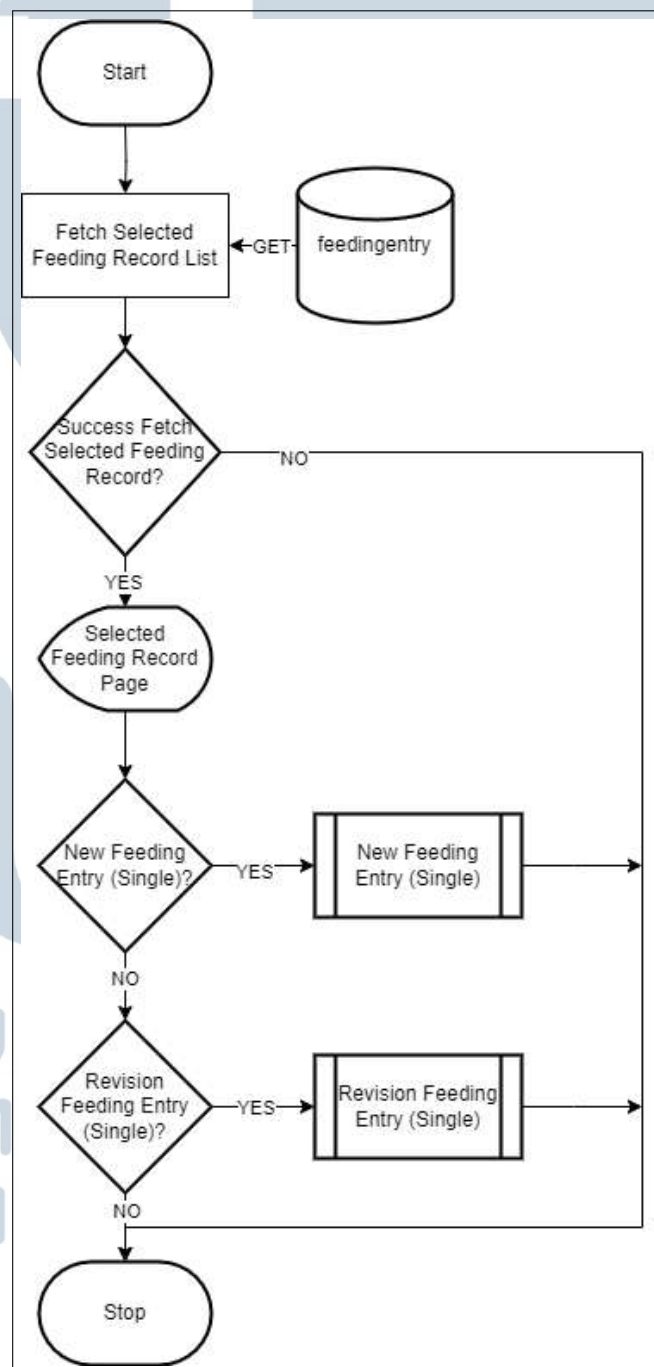
Gambar 3.43. Active Ponds Overview Details Flowchart

Gambar 3.43 menggambarkan alur kerja pada proses *active ponds details*. Pada proses ini *user* dapat melihat data detail terkait *active pond* yang dipilih. Selain itu *user* juga dapat melakukan *confirm stocking* apabila ingin memulai budidaya udang pada kolam tersebut.



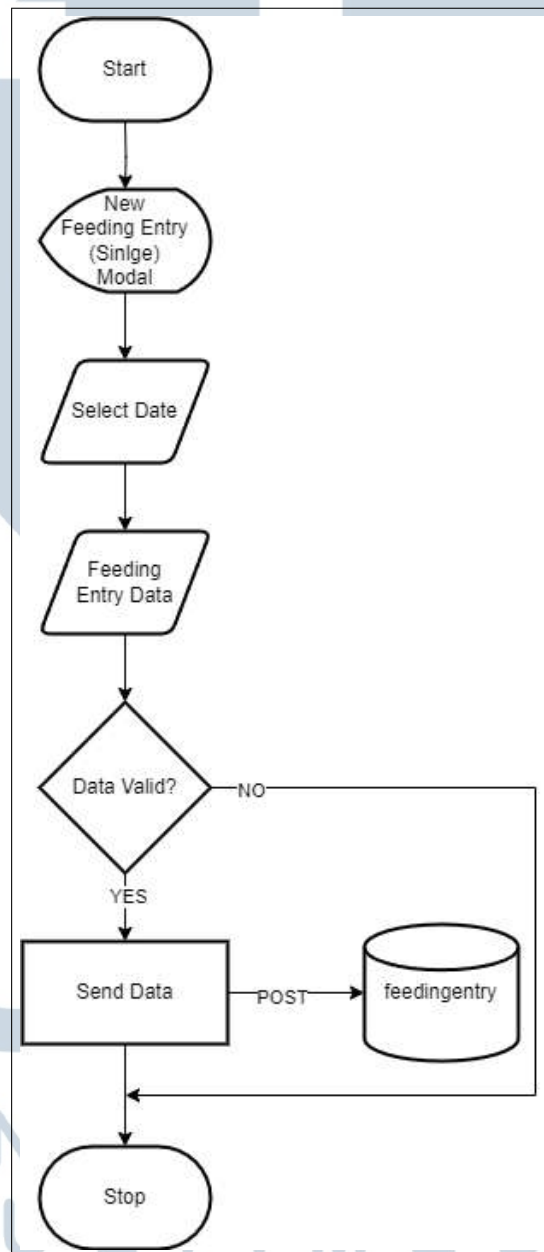
Gambar 3.44. *Feeding Record Flowchart*

Gambar 3.44 menggambarkan alur kerja pada proses *feeding record*. Pada proses ini *user* dapat melihat data pakan yang telah diberikan kepada udang dan rekomendasi pakan. Data yang dapat dilihat pada proses ini hanya data hari ini dan dua hari ke belakang. Apabila ingin melihat data yang lebih lengkap terdapat pada proses *feeding record details*. Pada proses ini terdapat pilihan untuk *input* data pakan secara sekaligus untuk beberapa kolam serta revisi data pakan.



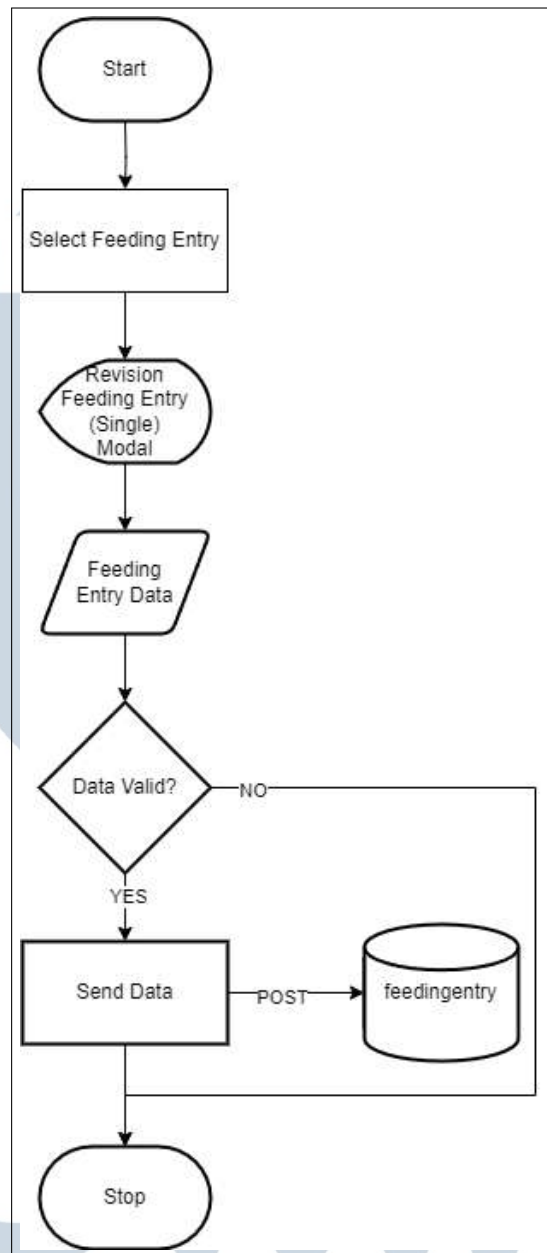
Gambar 3.45. *Feeding Record Details Flowchart*

Gambar 3.45 menggambarkan alur kerja pada proses *feeding record details*. Pada proses ini *user* dapat melihat keseluruhan data pakan yang telah diberikan beserta dengan rekomendasi pakan. Pada proses ini *user* juga diberikan pilihan untuk *input* data pakan baru dan revisi data pakan pada kolam yang diinginkan.



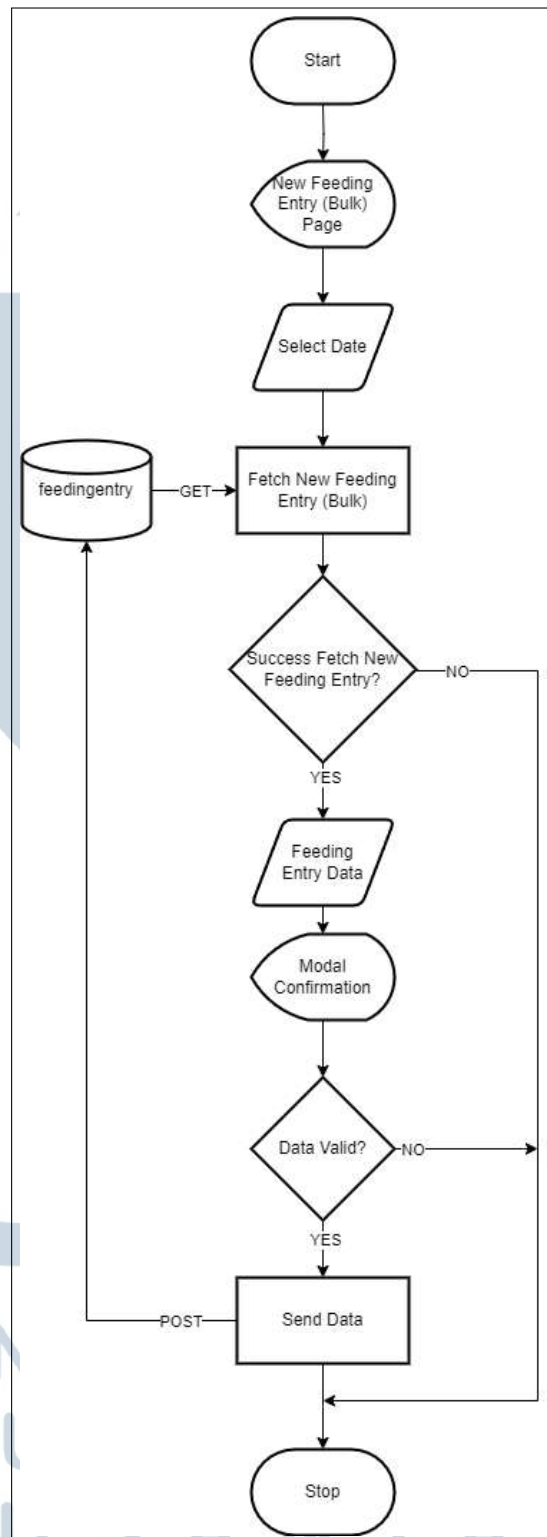
Gambar 3.46. *New Feeding Entry (Single) Flowchart*

Gambar 3.46 menggambarkan alur kerja pada proses *new feeding entry (single)*. Pada proses ini *user* dapat memasukkan data pakan pada kolam yang diinginkan. Data yang diperlukan adalah jumlah pakan (*feed weight*) dan tanggal pakan. Hanya dapat memasukkan data pada hari yang belum ada data pakan.



Gambar 3.47. *Revision Feeding Entry (Single) Flowchart*

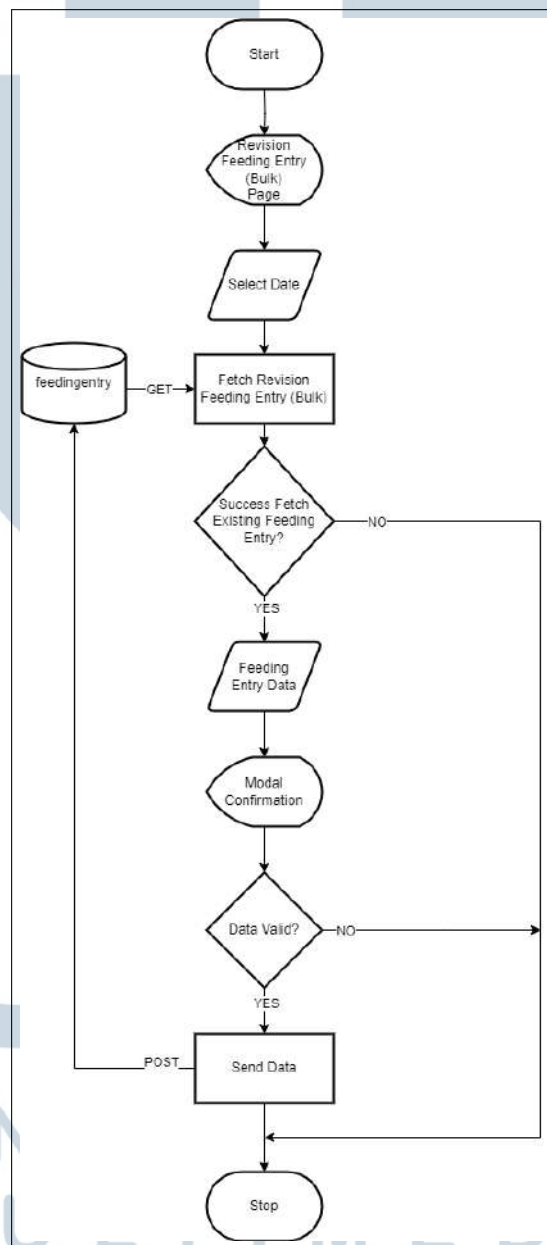
Gambar 3.47 menggambarkan alur kerja pada proses *revision feeding entry (single)*. Pada proses ini *user* dapat memasukkan data pakan pada kolam yang diinginkan. Data yang diperlukan adalah jumlah pakan (*feed weight*) dan tanggal pakan. Revisi hanya dapat dilakukan pada tanggal yang sudah memiliki data pakan sebelumnya.



Gambar 3.48. *New Feeding Entry (Bulk) Flowchart*

Gambar 3.48 menggambarkan alur kerja pada proses *new feeding entry (bulk)*. Pada proses ini *user* dapat memasukkan data pakan pada beberapa kolom

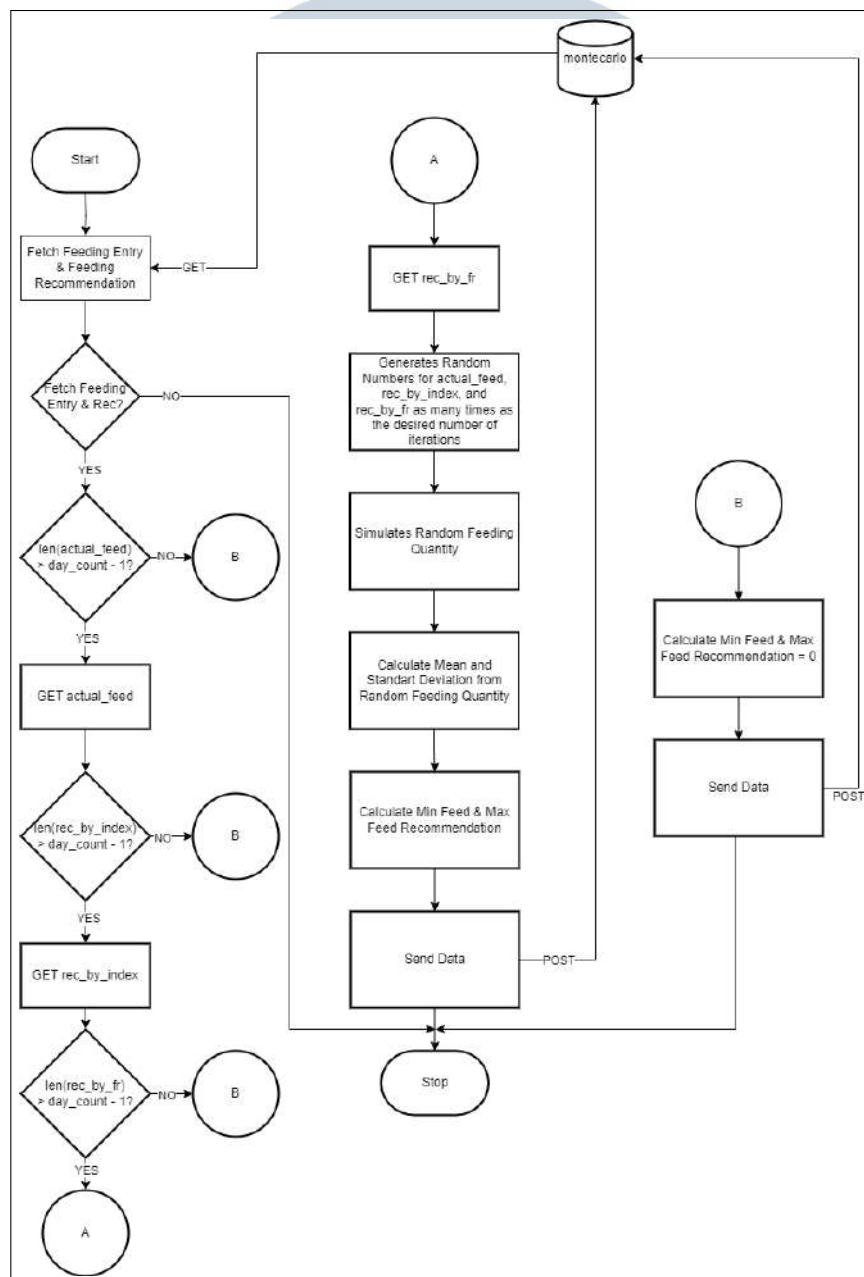
yang diinginkan sekaligus. Data yang diperlukan adalah jumlah pakan (*feed weight*), tanggal pakan, dan *ponds*. *User* hanya dapat memasukkan data pakan pada tanggal yang belum memiliki data pakan. Apabila sudah memiliki data pakan pada tanggal tersebut maka *user* hanya bisa revisi data pakan saja.



Gambar 3.49. *Revision Feeding Entry (Bulk) Flowchart*

Gambar 3.49 menggambarkan alur kerja pada proses *revision feeding entry (bulk)*. Pada proses ini *user* dapat memasukkan data pakan pada beberapa kolom yang diinginkan sekaligus. Data yang diperlukan adalah jumlah pakan (*feed weight*), tanggal pakan, dan *ponds*. Revisi hanya dapat dilakukan pada tanggal yang

sudah memiliki data pakan sebelumnya. Apabila belum memiliki data pakan pada tanggal tersebut, *user* harus *input new feeding*.



Gambar 3.50. Monte Carlo Simulation Flowchart

Gambar 3.50 menggambarkan alur kerja proses algoritma simulasi monte carlo. Sebelum melakukan simulasi monte carlo, program akan mengecek apakah sudah ada rekomendasi pakan maupun pakan sebenarnya dalam sistem. Apabila menemukan data tersebut, maka selanjutnya akan dicek banyaknya data yang terdapat dalam sistem. Apabila banyaknya data sudah sesuai dengan banyaknya

hari yang diinginkan, selanjutnya program akan menghasilkan angka acak dari banyaknya iterasi simulasi yang diinginkan dan program akan menghitung *mean*, standar deviasi, dan rekomendasi pakan dengan *confidence interval*. Apabila banyaknya data tidak sesuai dengan banyaknya hari yang diinginkan, maka program akan memberikan rekomendasi pakan nol yang berarti terdapat ketidaksesuaian pada data yang ada.

3.3.3 Entity Relationship Diagram

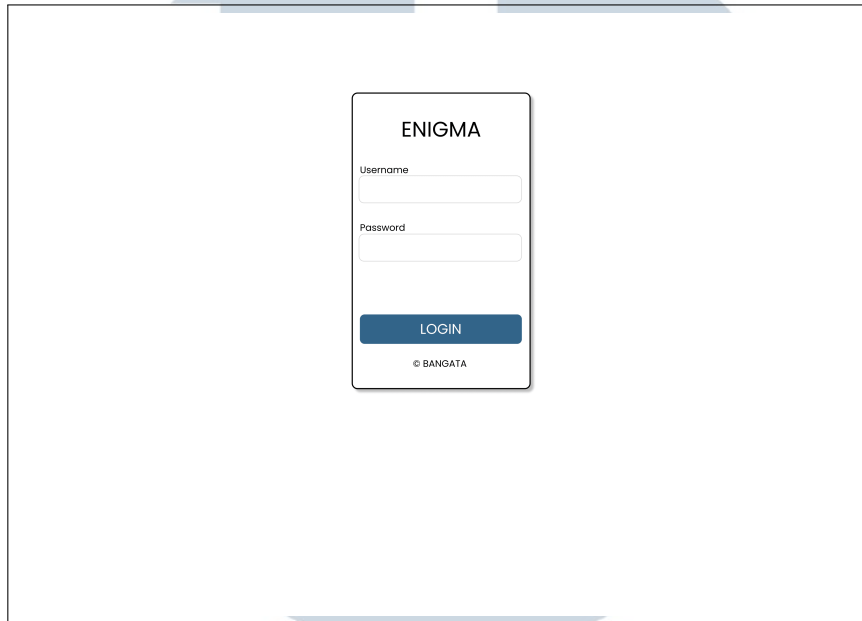
Struktur tabel yang telah dijelaskan saling terhubung membentuk satu basis data yang terintegrasi. Entity Relationship Diagram (ERD) digunakan sebagai model untuk menjelaskan hubungan data dalam basis data, berfokus pada objek-objek dasar data yang memiliki relasi tabel. ERD sering kali menjadi dasar persyaratan dalam pengembangan aplikasi atau sistem. Gambar 3.51 menampilkan ERD dari *database* sistem ERP yang digunakan.



Gambar 3.51. Entity Relationship Diagram sistem ERP

3.3.4 Antar Muka Sistem

Setelah perancangan sistem selesai dilakukan, selanjutnya adalah membuat antar muka sistem untuk memudahkan pengguna.

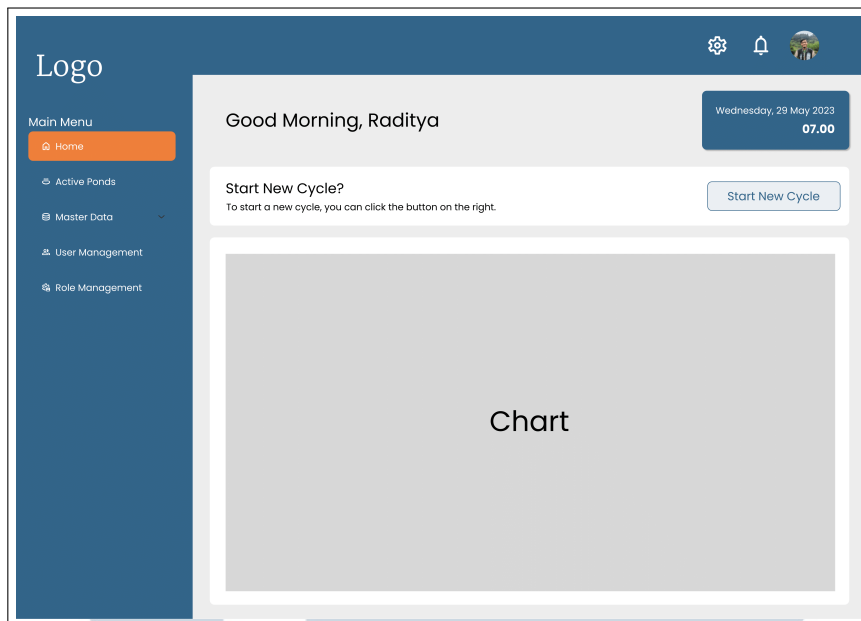


The image shows a login page for a system named 'ENIGMA'. The page is centered on a white background. At the top of the central box is the title 'ENIGMA'. Below the title are two input fields: 'Username' and 'Password'. Below the password field is a blue button with the text 'LOGIN'. At the bottom of the central box is the copyright notice '© BANGATA'.

Gambar 3.52. *Login Page*

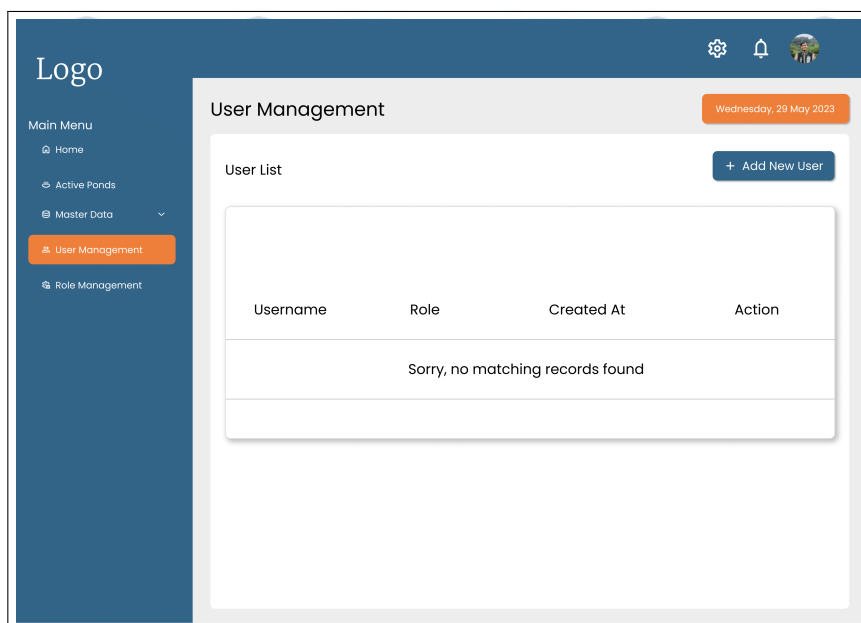
Gambar 3.52 merupakan tampilan awal ketika pengguna mengakses sistem ERP berbasis *web*. Pengguna harus login terlebih dahulu sebelum dapat menggunakan fitur yang terdapat dalam sistem ERP. *Credentials* pengguna didapatkan ketika awal membuat *user*.

UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



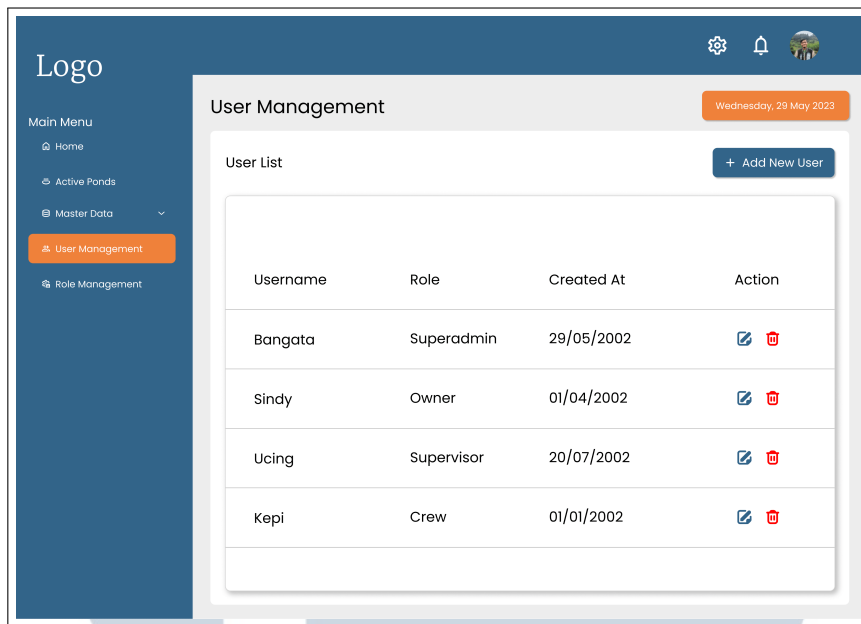
Gambar 3.53. Home Page

Gambar 3.53 merupakan tampilan dari *home page* setelah *user* berhasil masuk ke dalam sistem. Pada sidebar terdapat beberapa navigasi yang dapat mengantarkan pengguna ke dalam halaman/fitur yang diinginkan. Selain itu juga terdapat tombol *start new cycle* untuk memulai budidaya dan *chart* untuk menunjukkan pergerakan pakan.



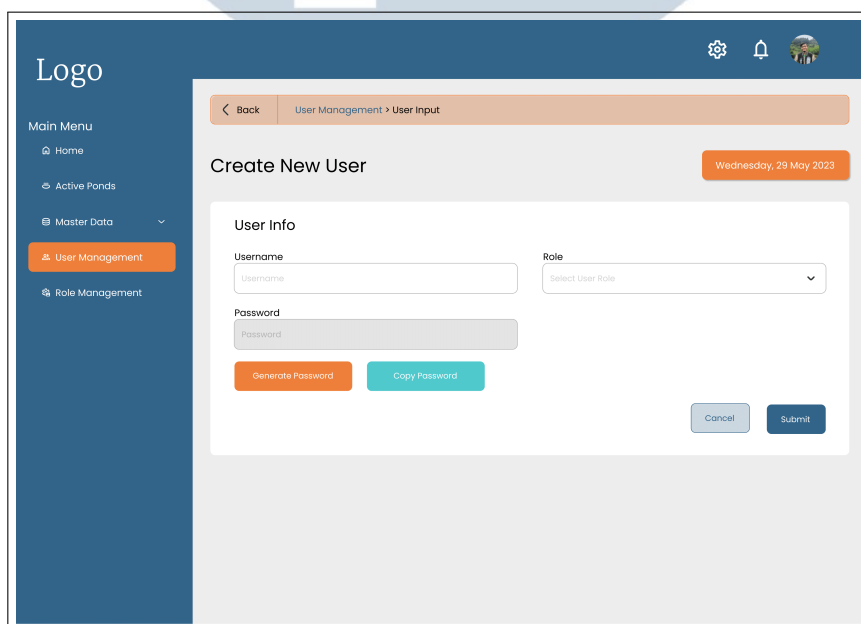
Gambar 3.54. User List Page (Empty)

Gambar 3.54 merupakan tampilan dari *user management* saat table kosong.



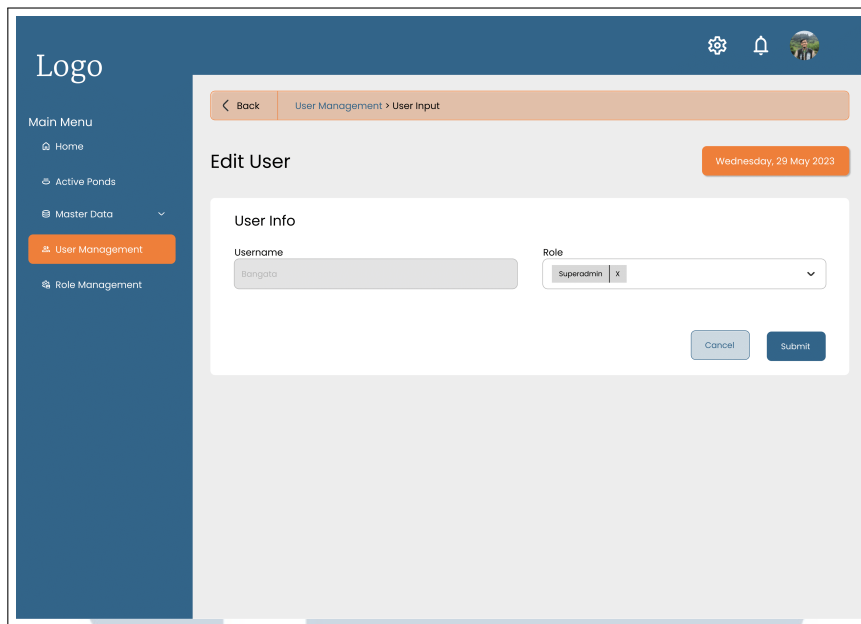
Gambar 3.55. User List Page (Filled)

Gambar 3.55 adalah tampilan tabel apabila terdapat data di dalamnya.



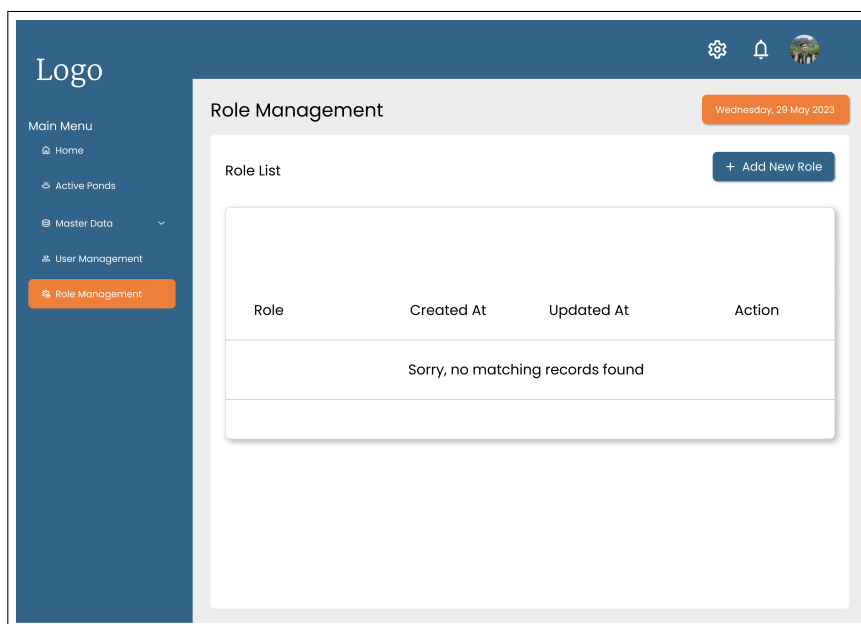
Gambar 3.56. User Create Page

Gambar 3.56 merupakan tampilan saat ingin menambahkan data pengguna baru. Terdapat tiga buah *formfield* yang wajib diisi sebelum dapat mengirimkan data ke dalam *database*. Untuk *formfield password* terdapat dua buah tombol untuk membuat karakter acak dan tombol menyalin *password* tersebut.



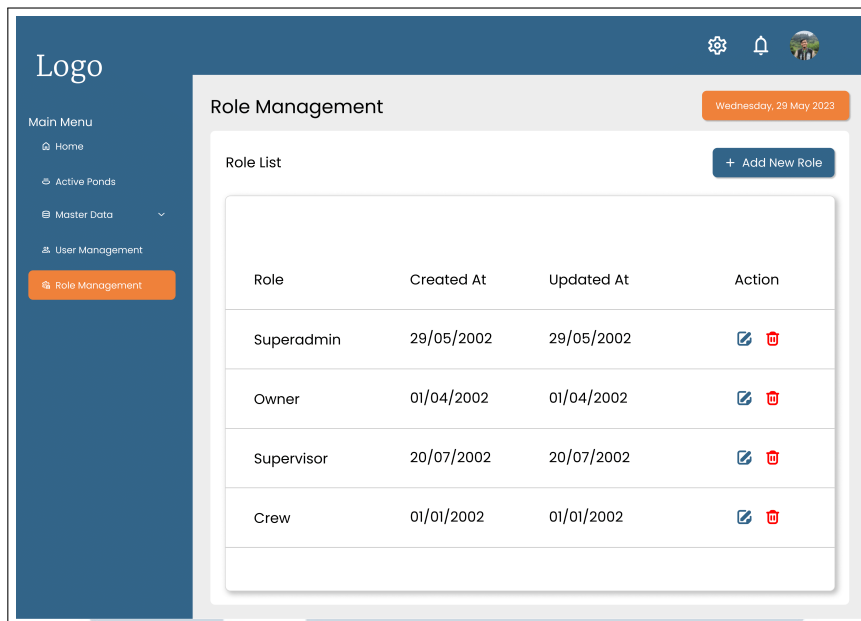
Gambar 3.57. *User Edit Page*

Gambar 3.57 merupakan tampilan apabila ingin menyunting data pengguna. Yang membedakan dari keduanya adalah saat menyunting *username* tidak dapat diubah dan tidak ada *formfield password*.



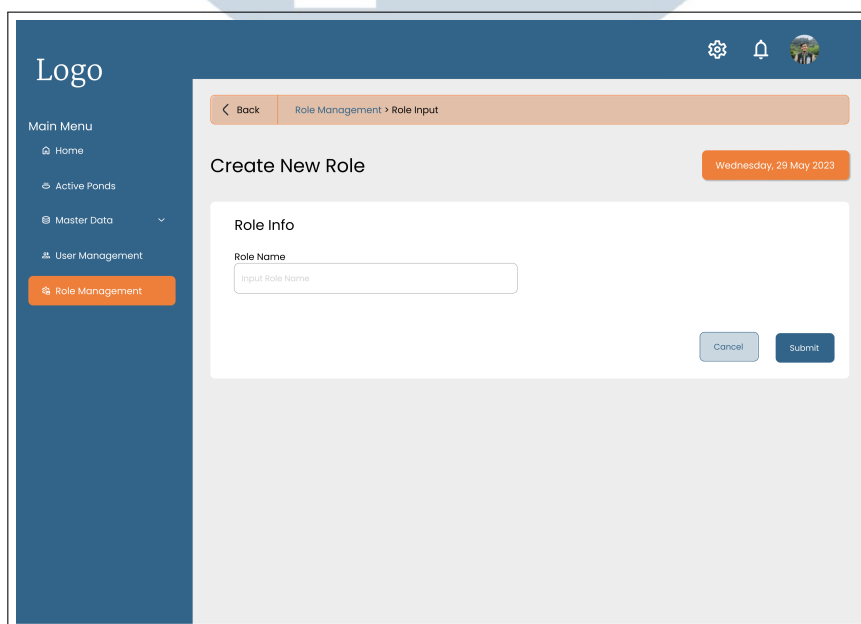
Gambar 3.58. *Role List Page (Empty)*

Gambar 3.58 merupakan tampilan dari *role management* saat tabel kosong.



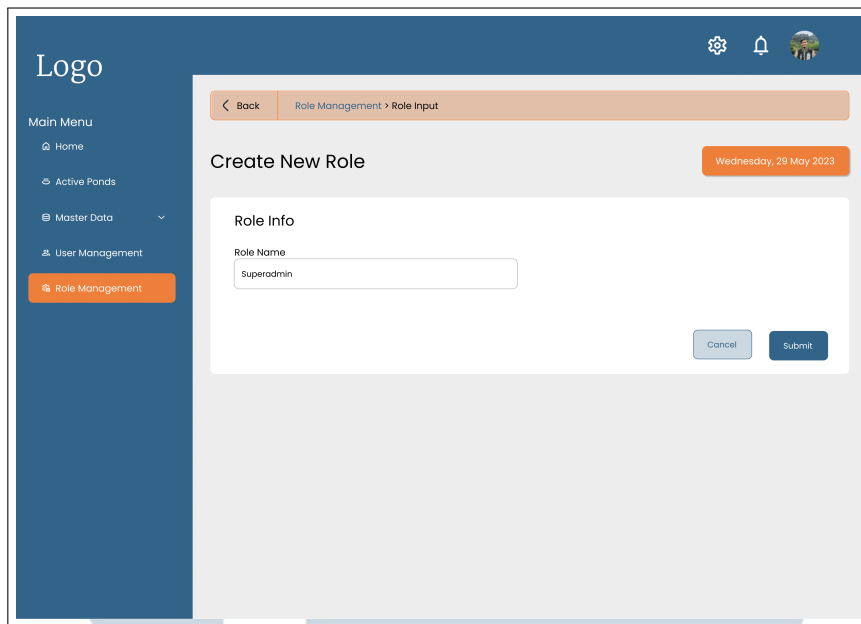
Gambar 3.59. Role List Page (Filled)

Gambar 3.59 adalah tampilan tabel apabila terdapat data di dalamnya.



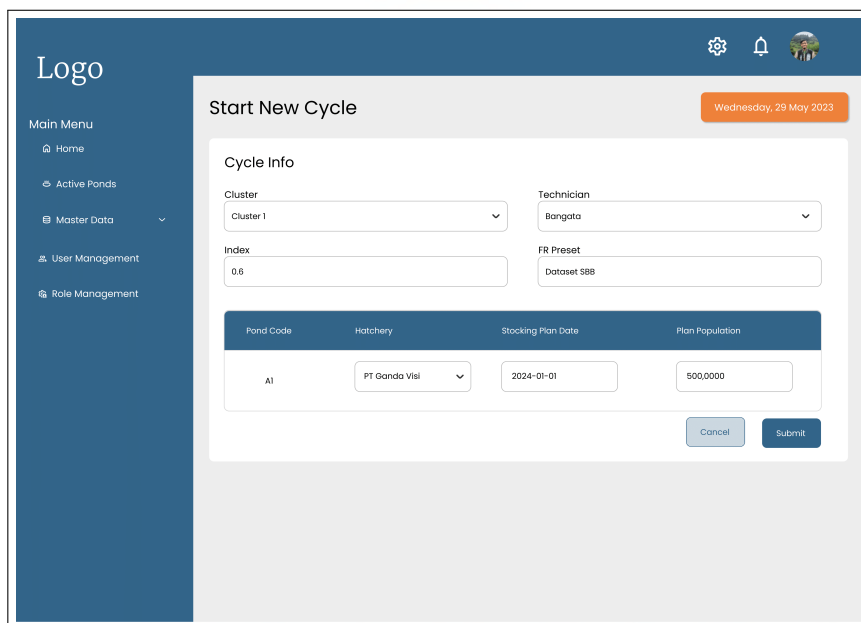
Gambar 3.60. Role Create Page

Gambar 3.60 merupakan tampilan saat ingin menambahkan data *role* baru. Terdapat sebuah *formfield* yang wajib diisi sebelum dapat mengirimkan data ke dalam *database*.



Gambar 3.61. *Role Edit Page*

Gambar 3.61 merupakan tampilan apabila ingin menyunting data *role*. Yang membedakan dari keduanya adalah saat menyunting perlu mendapatkan data yang sudah ada terlebih dahulu.



Gambar 3.62. *Start New Cycle Page*

Gambar 3.62 merupakan tampilan halaman saat ingin memulai atau mengaktifasi kolam untuk budidaya. Pada halaman tersebut terdapat beberapa *formfield* yang perlu diisi sebelum dikirimkan ke dalam *database*. Untuk data

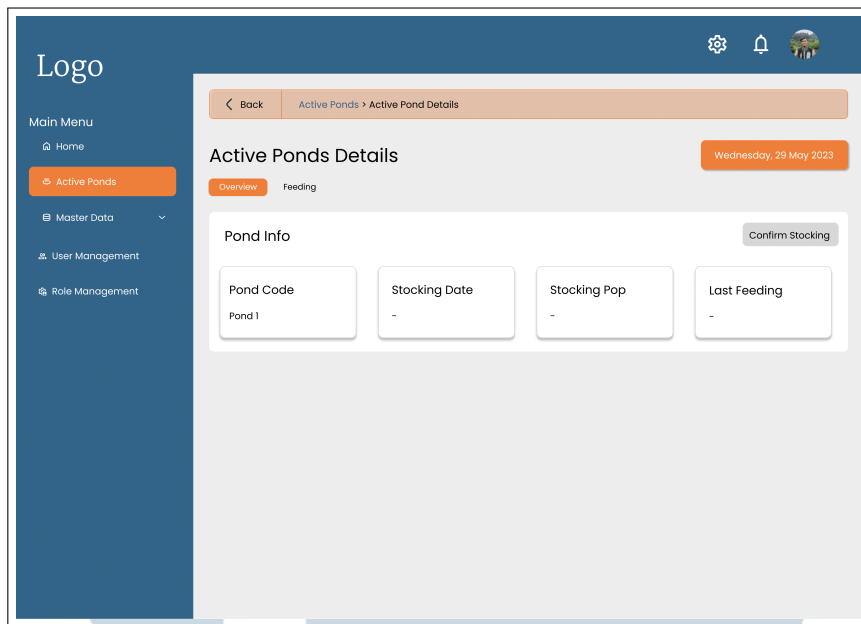
clusters, technician, FR Preset, dan Hatchery diambil dari *database* yang kemudian dibuat menjadi *select options*. Untuk *index* hanya dapat menerima angka lebih besar dari 0 dan lebih kecil sama dengan 1. *Index* tersebut yang akan digunakan untuk melakukan perhitungan rekomendasi pakan.

Pond Code	DoC	Est. Population	Est. MBW (gr)	Last 3d Feed (kg)	Technician
Pond 1	41	500,000	4.1	40 > 100 > 150	Bangata

Gambar 3.63. *Active Ponds Overview List Page*

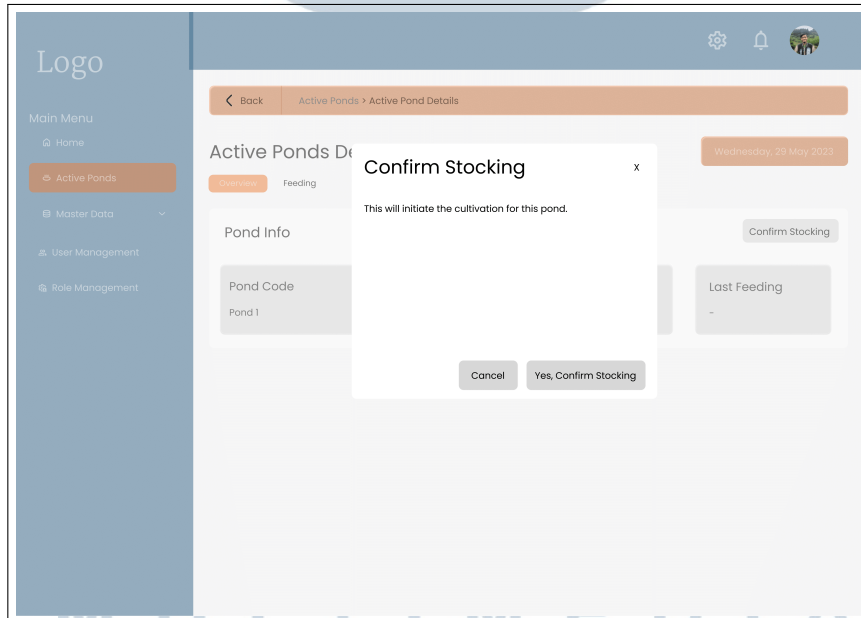
Gambar 3.63 merupakan tampilan kolam yang telah aktif (*active ponds*). Pada halaman ini terdapat *overview* terhadap *active ponds*. Selain itu terdapat *tab* untuk berpindah halaman ke halaman *feeding* untuk melihat data pakan.

UIMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



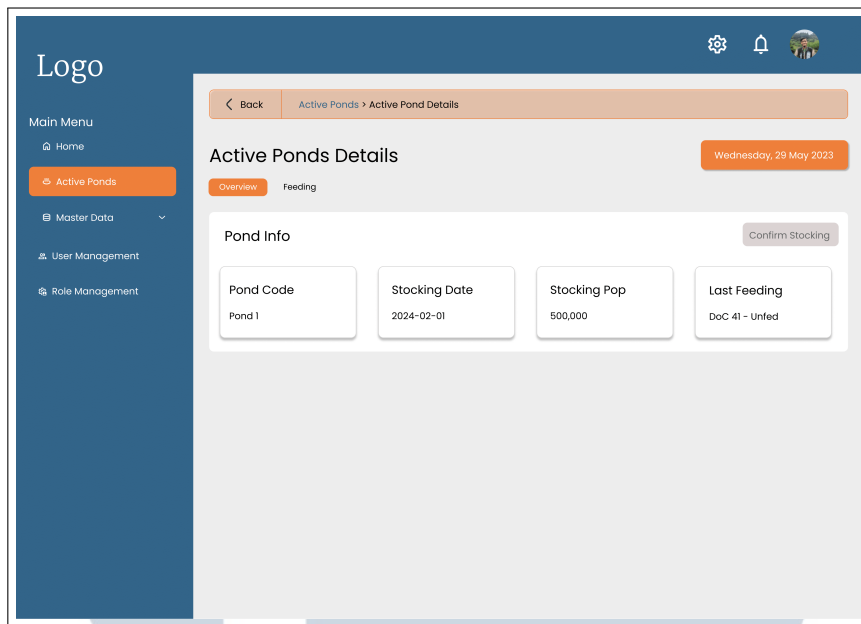
Gambar 3.64. *Active Ponds Overview Detail Page (Not Yet Stocking)*

Gambar 3.64 merupakan tampilan ketika kolam sudah memulai *cycle* namun belum dikonfirmasi.



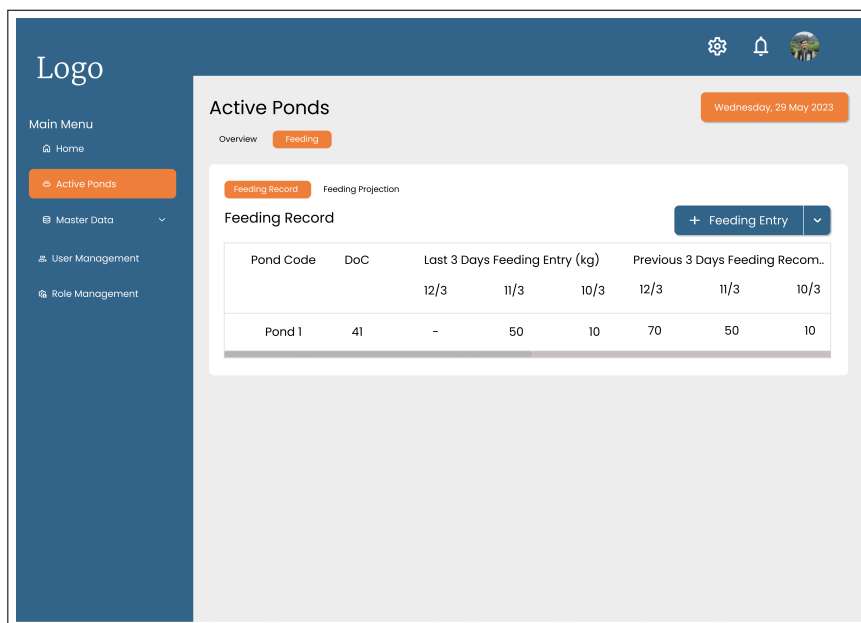
Gambar 3.65. *Modal Confirm Stocking*

Kolam yang belum dikonfirmasi perlu dikonfirmasi dengan tombol yang dapat dipencet untuk memunculkan modal pada Gambar 3.65 untuk mengkonfirmasi dan memulai budidaya.



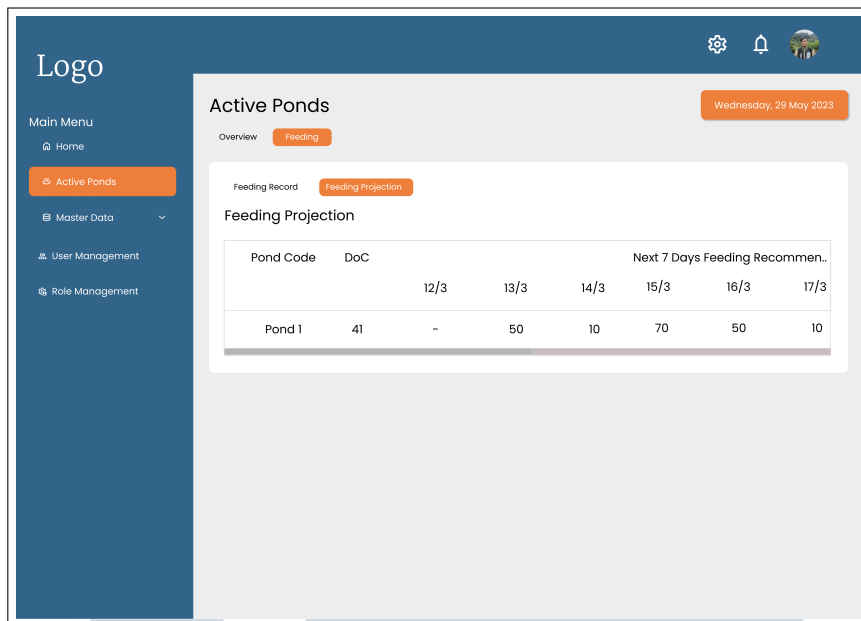
Gambar 3.66. *Active Ponds Overview Detail Page (After Stocking)*

Gambar 3.66 merupakan tampilan ketika kolam sudah berhasil dikonfirmasi populasi dan tanggal tebarannya. Tombol *confirm stocking* sudah tidak bisa dipencet karena status kolam sudah aktif.



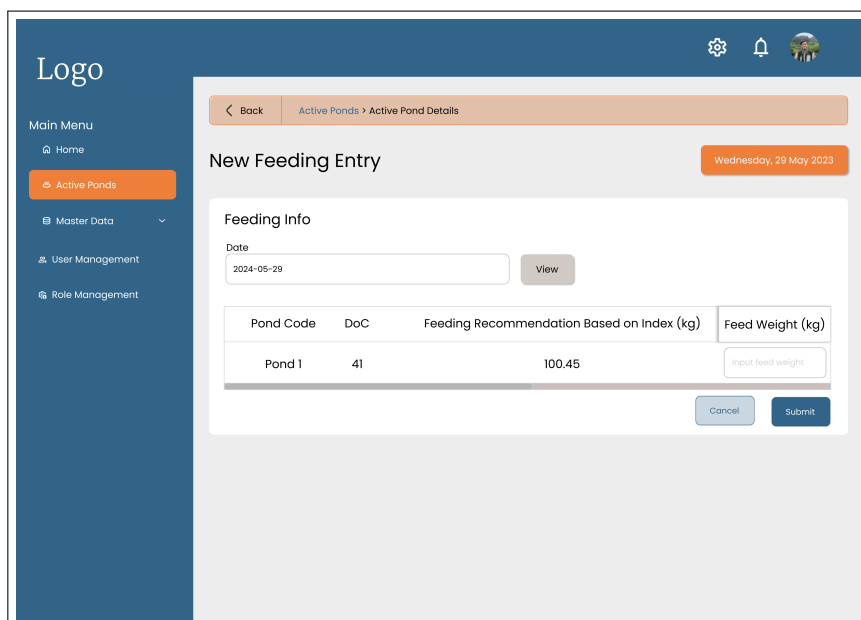
Gambar 3.67. *Feeding Record List Page*

Gambar 3.67 merupakan tampilan awal ketika berada dalam tab *feeding*. Terdapat informasi terkait data pakan tiga hari ke belakang terhitung dari hari ini. Terdapat tombol untuk memasukkan data pakan dan merevisi data pakan.



Gambar 3.68. *Feeding Recommendation List Page*

Terdapat tab lain untuk berpindah ke halaman *feeding projection*. Gambar 3.68 merupakan tampilan apabila sedang berada di *tab feeding recommendation*. Halaman tersebut menampilkan rekomendasi pakan dari hitungan index, FR, dan monte carlo.



Gambar 3.69. *New Feeding Entry Page (Bulk)*

Gambar 3.69 merupakan tampilan untuk memasukkan data pakan secara banyak ke dalam beberapa *ponds* yang aktif. Pengguna diminta untuk memilih

tanggal terlebih dahulu baru untuk mendapatkan data kolam yang belum diberi pakan pada tanggal tersebut. Data yang ditampilkan hanya data yang belum memiliki data pakan sebelumnya.

The screenshot shows a web application interface for 'Feeding Entry Revision'. The page title is 'Feeding Entry Revision' and the date is 'Wednesday, 29 May 2023'. The 'Feeding Info' section includes a date field with the value '2024-05-29' and a 'View' button. Below this is a table with the following data:

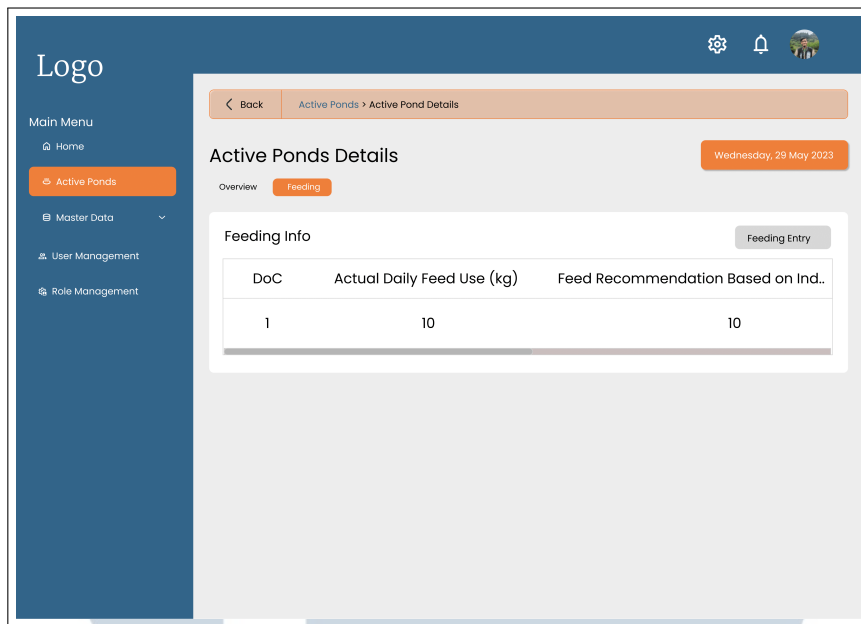
Pond Code	DoC	Feeding Recommendation Based on Index (kg)	Feed Weight (kg)
Pond 1	41	100.45	100

At the bottom of the table are 'Cancel' and 'Submit' buttons.

Gambar 3.70. *Revision Feeding Entry Page (Bulk)*

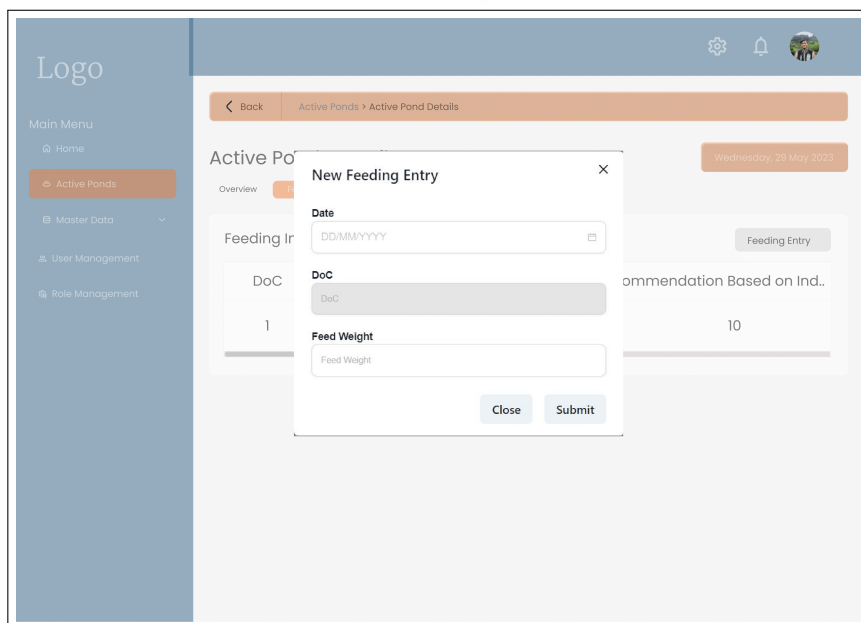
Gambar 3.70 merupakan halaman untuk merevisi data pakan. Hal yang membedakan kedua halaman adalah data yang ditampilkan. Halaman revisi hanya menampilkan data yang sudah memiliki data pakan sebelumnya, sebaliknya halaman *new entry* hanya menampilkan data yang belum memiliki data pakan sebelumnya.

U M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



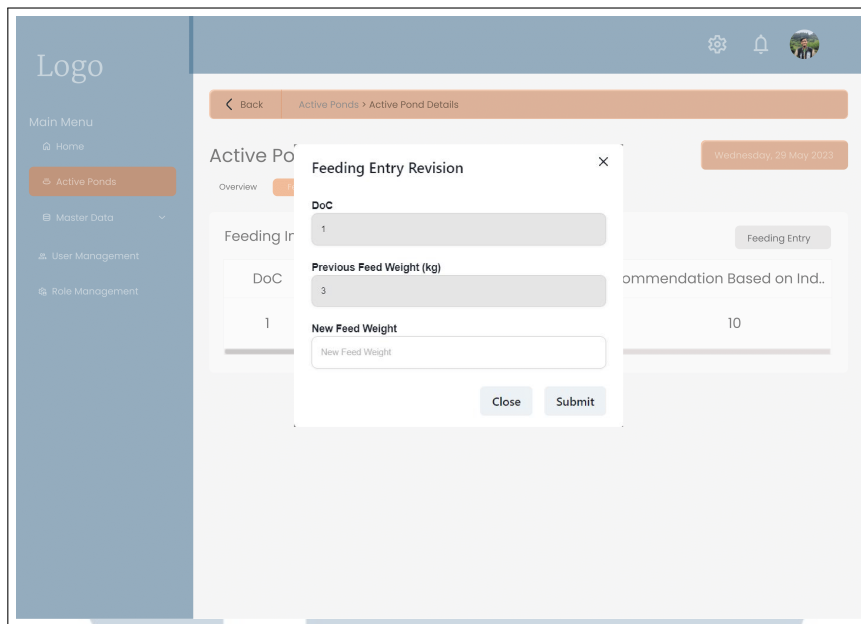
Gambar 3.71. *Active Ponds Feeding Details Page*

Gambar 3.71 merupakan tampilan detail terkait pakan dari sebuah *active pond*. Pada halaman ini terdapat juga tombol untuk memasukkan data pakan pada kolom tersebut.



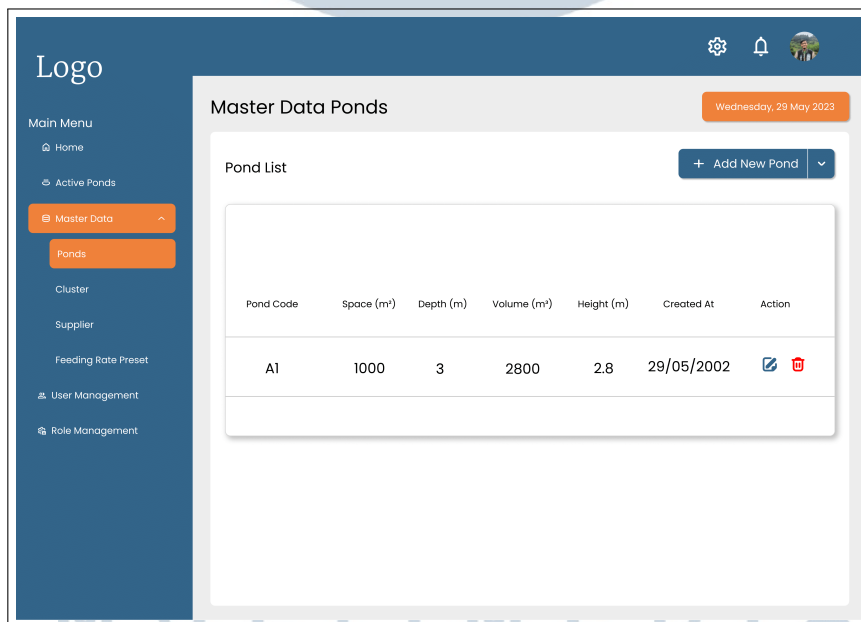
Gambar 3.72. *Modal New Feeding Entry (Single)*

Gambar 3.72 tampilan modal untuk memasukkan data pakan baru ke dalam kolom tersebut. Sama seperti sebelumnya, *new entry* hanya dapat dilakukan pada tanggal yang belum memiliki data pakan sebelumnya.



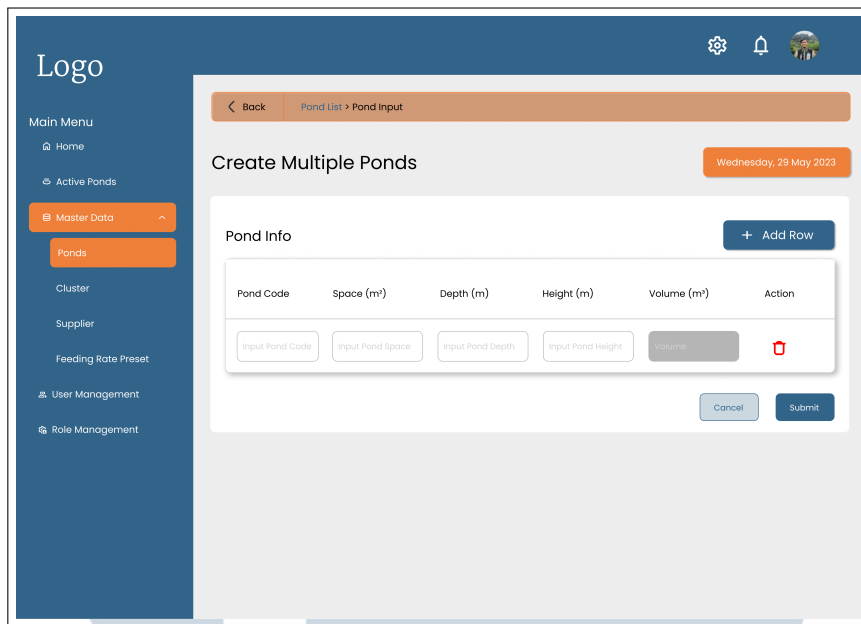
Gambar 3.73. Modal Revision Feeding Entry (Single)

Gambar 3.73 merupakan modal untuk merevisi data pakan yang dapat dimunculkan dengan menekan tombol mata pada ujung kanan tabel pakan.



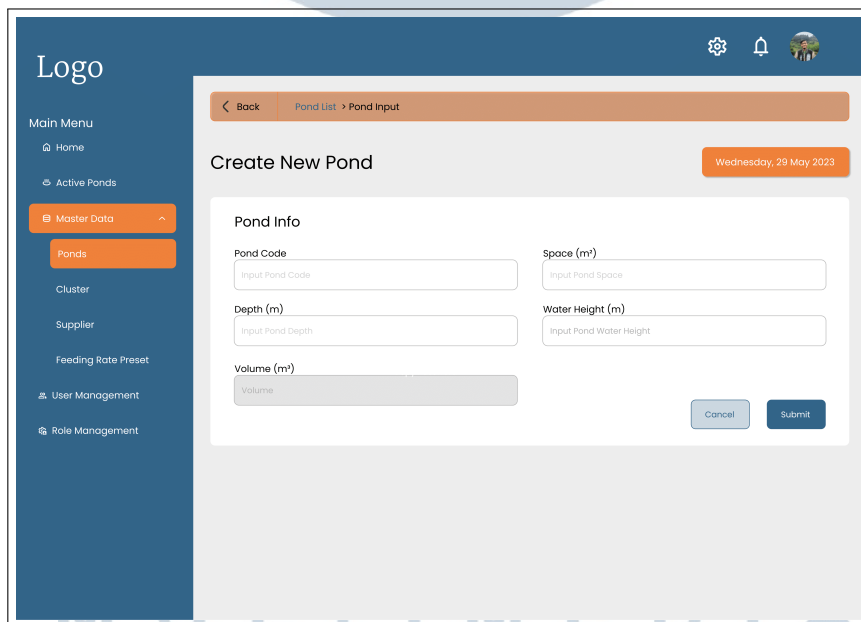
Gambar 3.74. Master Data Ponds List Page

Gambar 3.74 merupakan tampilan *list* dari kolam yang ada. Pada halaman tersebut memungkinkan pengguna untuk melakukan beberapa hal seperti *create single* dan *multiple*, *edit*, dan *delete* kolam yang ada.



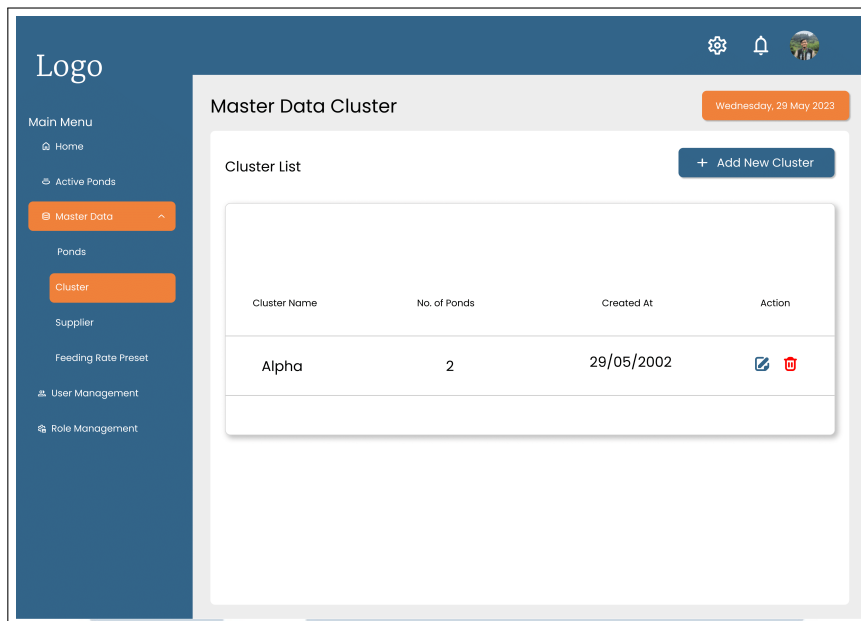
Gambar 3.75. *Master Data Ponds Create Page (Multiple)*

Gambar 3.75 merupakan tampilan *create multiple ponds*. Pada halaman tersebut memungkinkan pengguna untuk membuat banyak kolom sekaligus.



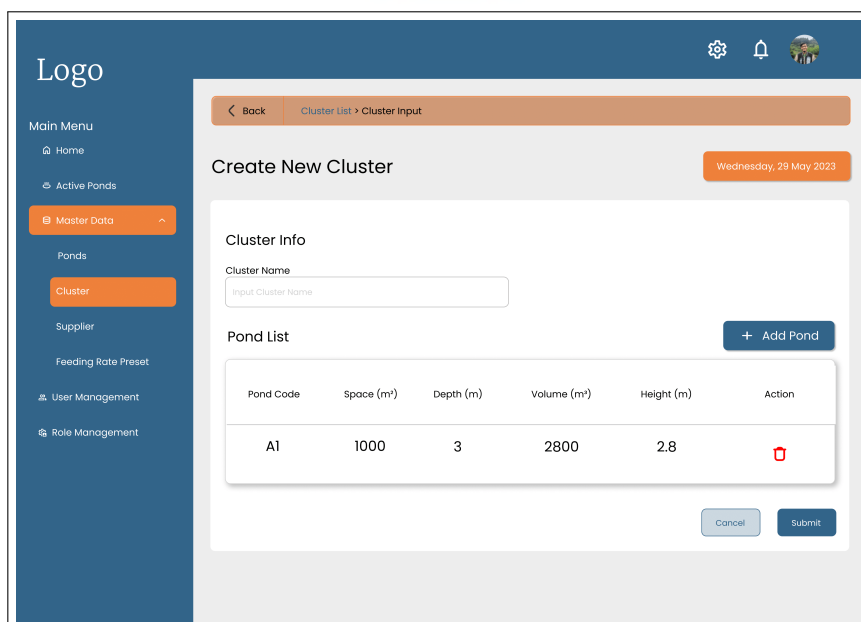
Gambar 3.76. *Master Data Ponds Create Page (Single)*

Gambar 3.76 merupakan tampilan *create single pond*. Sedangkan untuk halaman *edit* sama persis dengan halaman *create single* namun yang membedakan adalah halaman *edit* mengambil data yang sudah ada terlebih dahulu untuk dimasukkan ke dalam *formfield* yang ada.



Gambar 3.77. Master Data Clusters List Page

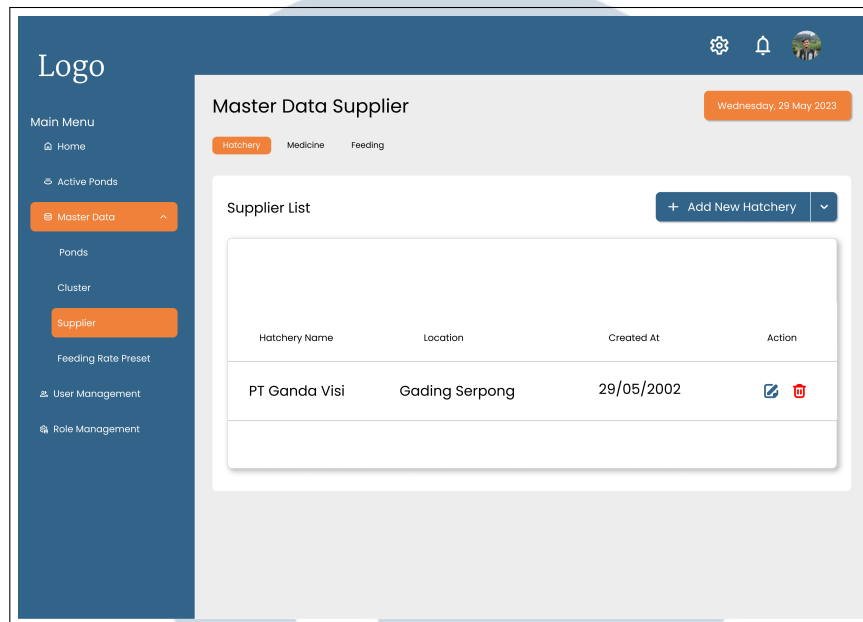
Gambar 3.77 merupakan tampilan *list* dari *clusters* kolom yang ada. Pada halaman tersebut memungkinkan pengguna untuk melakukan beberapa hal seperti *create*, *edit*, dan *delete cluster* yang ada.



Gambar 3.78. Master Data Clusters Create Page

Gambar 3.78 merupakan tampilan saat ingin membuat *cluster* baru. Pada halaman terdapat tombol untuk menambahkan kolam yang ingin dimasukkan ke dalam *cluster* tersebut. Sedangkan untuk halaman *edit* sama persis dengan halaman

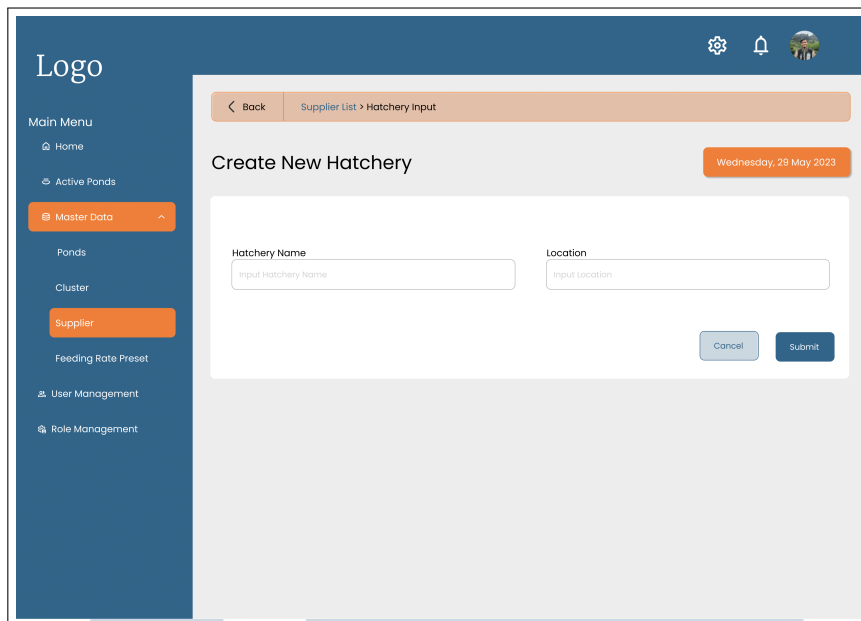
create namun yang membedakan adalah halaman *edit* mengambil data yang sudah ada terlebih dahulu untuk dimasukkan ke dalam *formfield* dan tabel yang ada.



Gambar 3.79. *Master Data Suppliers List Page*

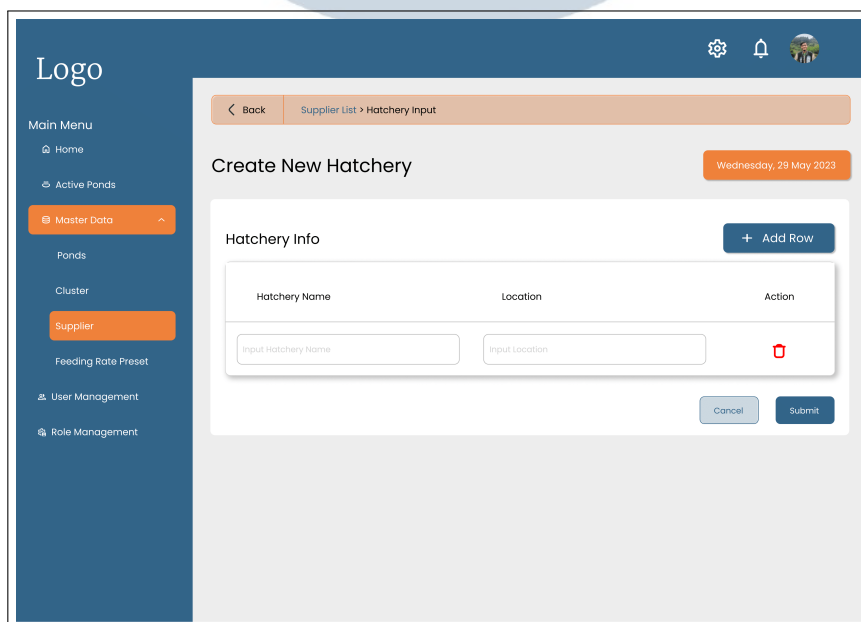
Gambar 3.79 merupakan tampilan dari *master data suppliers* yang menampilkan data *suppliers*. Data *suppliers* dibagi menjadi tiga tipe yaitu *hatchery*, *feeding*, dan *medicine*. Walaupun memiliki tiga tipe yang berbeda namun data yang ditampilkan memiliki kolom yang sama yaitu *name* dan *location*. Pada halaman tersebut memungkinkan pengguna untuk menambahkan data *suppliers* secara banyak maupun hanya satu, *edit* data *supplier* serta menghapus data *supplier* yang ada.

UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



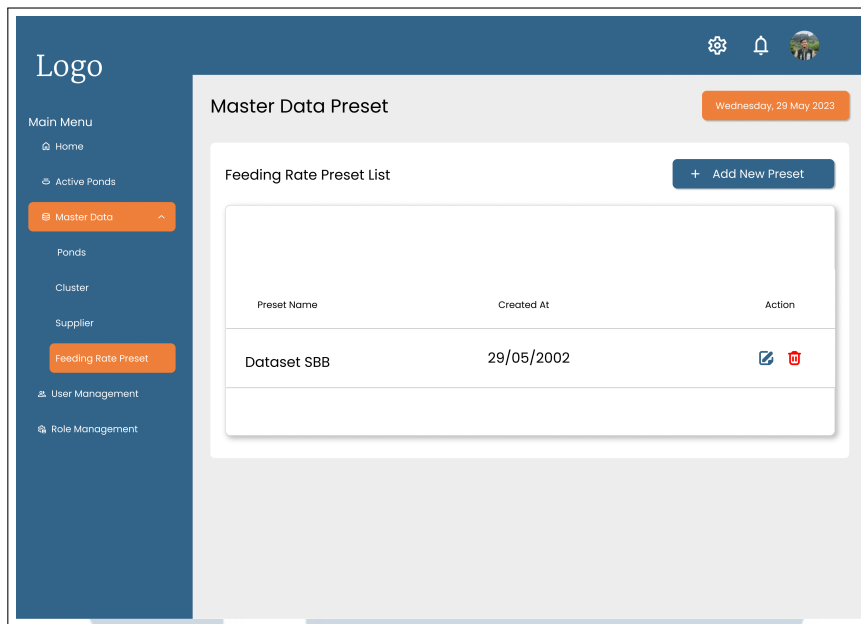
Gambar 3.80. *Master Data Suppliers Create Page (Single)*

Gambar 3.80 merupakan halaman membuat *supplier* baru secara *single*. Pada halaman ini juga digunakan untuk halaman *edit supplier*.



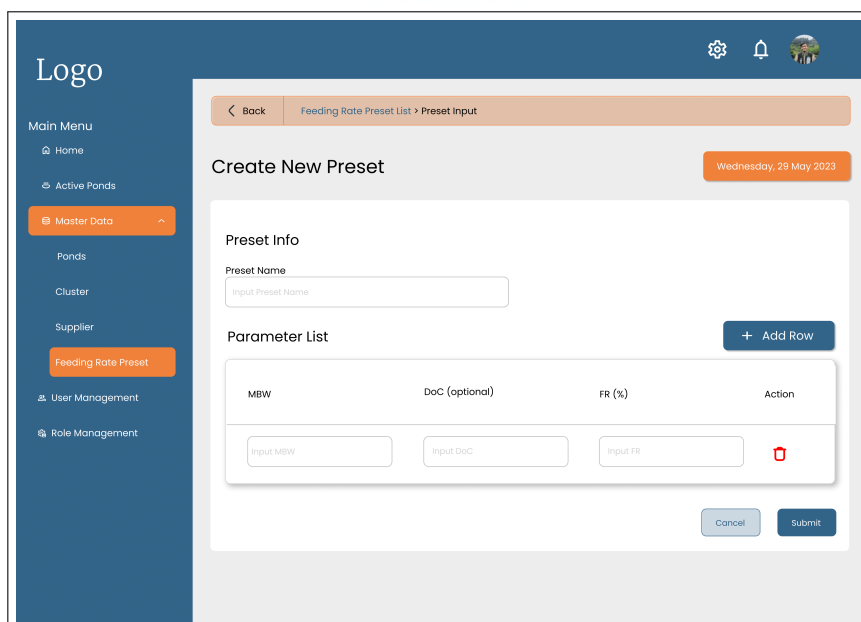
Gambar 3.81. *Master Data Suppliers Create Page (Multiple)*

Gambar 3.81 merupakan halaman untuk membuat *suppliers* secara banyak dan bersamaan. Sama seperti *master data ponds*, fitur *multiple* hanya dapat digunakan untuk membuat *suppliers* baru saja.



Gambar 3.82. Master Data FR Presets List Page

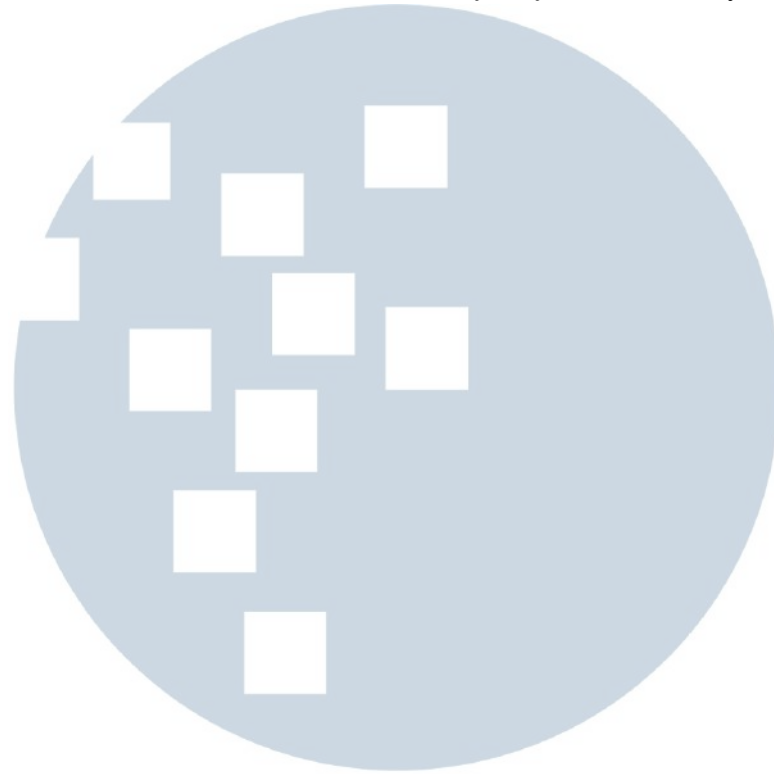
Gambar 3.82 merupakan tampilan *list* dari *FR Presets* yang ada. Pada halaman tersebut memungkinkan pengguna untuk melakukan beberapa hal seperti *create*, *edit*, dan *delete FR Presets* yang ada.



Gambar 3.83. Master Data FR Presets Create Page

Gambar 3.83 merupakan tampilan saat ingin membuat *FR Preset* baru. Pada halaman terdapat tombol untuk menambahkan parameter yang ingin digunakan pada *preset* tersebut. Sedangkan untuk halaman *edit* sama persis dengan halaman

create namun yang membedakan adalah halaman *edit* mengambil data yang sudah ada terlebih dahulu untuk dimasukkan ke dalam *formfield* dan tabel yang ada.



UMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA