BAB IV  
IMPLEMENTASI

1. Desain Use case Diagram

Dibawah ini adalah *Diagram Use Case* yang menggambarkan interaksi antara pengguna dansistem.Berikut ini adalah langkah-langkah dalam membuat *Diagram Use Case,* yaitu

* + - * 1. Identifikasi Aktor

Dibawah ini adalah tabel identifikasi *actor*. Sebagai berikut:

Tabel 4.40 Indentifikasi aktor

| **No** | **NamaAktor** | **Deskripsi** |
| --- | --- | --- |
| 1**.** | *Owner* | Aktor yang telah terdaftar dalam sistem yang dapat mengakses seluruh menu |
| 2**.** | *Owner* | Aktor yang telah terdaftar dalam sistem memiliki hak akses untuk memasukkan daftar menu, pesanan. |
| 3**.** | *Customer* | Aktor yang telah terdaftar dalam sistem memiliki hak akses untuk mencari menu dan memesan. |

* + - * 1. Identifikasi Use case

Dibawah Ini Adalah Tabel Identifikasi *Use Case*. Sebagai Berikut:

Tabel 4.41 Indentifikasi *Use case*

| No | *Use case* | Aktor | Deskripsi |
| --- | --- | --- | --- |
| 1. | Login | *Owner*, *Admin*, *Customer* | *Use Case* menggambarkan kegiatan *Login*, dengan memasukkan *username* dan *password.* |
| 2. | Memasukkan Menu | *Owner* | *Use Case* menggambarkan kegiatan memasukkan menu |
| 3. | Menampilkan Menu | *Owner* | *Use Case* menggambarkan kegiatan menampilkan menu |
| 4. | Mengubah Menu | *Owner* | *Use Case* menggambarkan kegiatan mengubah menu |
| 5. | Delete Menu | *Owner* | *Use Case* menggambarkan kegiatan menghapus menu |
| 6. | Masukkan Data Kesehatan | *Owner*, *Admin* | *Use Case* menggambarkan kegiatan memasukkan data kesehatan |
| 7. | menampilkan Bobot Atribut | *Owner*, *Admin* | *Use Case* menggambarkan kegiatan menampikan bobot atribut |
| 8. | Menghapus Bobot Atribut | *Owner*, *Admin* | *Use Case* menggambarkan kegiatan menghapus bobot atribut |
| 9. | Menampilkan Data Kesehatan | *Owner*, *Admin* | *Use Case* menggambarkan kegiatan menampilkan *database* kesehatan |
| 10. | Menghapus Data Kesehatan | *Owner*, *Admin* | *Use Case* menggambarkan kegiatan menghapus *database* kesehatan |
| 11. | *Clustering*  Pegawai | *Owner*, *Admin*, *Customer* | *Use Case* menggambarkan kegiatan perhitungan *clustering*  pegawai berdasarkan *database* kesehatan yang dipilih |
| 12. | Menampilkan Hasil *Clustering*  Pegawai | *Owner*, *Admin*, *Customer* | *Use Case* menggambarkan kegiatan menampilkan hasil dari perhitungan *clustering*  pegawai |
| 13. | Menghapus Hasil *Clustering*  Pegawai | *Owner*, *Admin*, *Customer* | *Use Case* menggambarkan kegiatan menghapus hasil perhitungan *clustering*  pegawai |
| 14. | Logout | *Owner*, *Admin*, *Customer* | *Use Case* menggambarkan kegiatan logout dari aplikasi |

* + - * 1. *Diagram Use case*

Dibawah ini adalah gambar dari *Diagram Use case*. Sebagai berikut:



Gambar 4. 6 *Use case*

* + - * 1. Narasi Use case
        2. Narasi *Use case* *Login*

Tabel 4.42 *Use case* login

|  |  |  |
| --- | --- | --- |
| *Use case* Name | Login | |
| *Use case* Id | 1 | |
| Actor | *Owner*, *Admin*, *Customer* | |
| Description | *Use Case* menggambarkan kegiatan *Login*, dengan memasukkan *username* dan *password.* | |
| Precondition | Aktor memilih aplikasi | |
| Typical Course of Events | Aksi Aktor  1.Pilih aplikasi  3.Mengisi *user*name dan password  4.Klik submit | Respon Sistem  2.Menampilkan halaman *login*  5.Menampilkan halaman index |
| Alterate Course | jika *username* dan *password* salah maka akan kembali ke halaman *login* dan aktor harus *inputusername* dan *password*kembali. | |
| Conclusion | Aktor berhasil *login* kedalam sistem | |
| Post Condition | Menampilkan halaman index dalam sistem sesuai dengan hak akses | |

* + - * 1. Narasi *Use case* Memasukkan Menu

Tabel 4.43 *Use case* masukkan menu

|  |  |  |
| --- | --- | --- |
| *Use case* Name | Memasukkan Menu | |
| *Use case* Id | 2 | |
| Actor | *Owner* | |
| Description | *Use Case* menggambarkan kegiatan memasukkan menu | |
| Precondition | Aktor telah *login* | |
| Typical Course of Events | Aksi Aktor  1.Pilih menu menu  2.Pilih submenu masukkan menu  4.Mengisi form masukkan menu  5.Klik submit | Respon Sistem  3.Menampilkan form masukkan menu  6.Memasukkan data menu kedalam *database* |
| Alterate Course | 6.jika data yang dimasukkan terdapat kesalahan maka akan kembali ke halaman form masukkan menu | |
| Conclusion | Aktor berhasil memasukkan menu | |
| Post Condition | Menu tersimpan dalam *database* | |

* + - * 1. Narasi *Use case* Menampilkan Menu

Tabel 4.44 *Use case* menampilkan menu

|  |  |  |
| --- | --- | --- |
| *Use case* Name | Menampilkan Menu | |
| *Use case* Id | 3 | |
| Actor | *Owner* | |
| Description | *Use Case* menggambarkan kegiatan menampilkan *user* | |
| Precondition | Aktor telah *login* | |
| Typical Course of Events | Aksi Aktor  1.Pilih menu menu  2.Pilih submenu tampilkan menu | Respon Sistem  3.Menampilkan menu yang ada dalam *database* |
| Alterate Course | 3.jika dalam *database* menu tidak ada maka akan menampilkan tabel kosong | |
| Conclusion | Aktor berhasil menampilkan menu | |
| Post Condition |  | |

* + - * 1. Narasi *Use case* Menghapus Menu

Tabel 4.45 *Use case* menghapus menu

|  |  |  |
| --- | --- | --- |
| *Use case* Name | Menghapus Menu | |
| *Use case* Id | 4 | |
| Actor | *Owner* | |
| Description | *Use Case* menggambarkan kegiatan menghapus*user* | |
| Precondition | Aktor telah *login* | |
| Typical Course of Events | Aksi Aktor  1.Pilih menu menu  2.Pilih submenu hapus menu  4.Pilih menu yang akan dihapus  6.Pilih ya | Respon Sistem  3.Menampilkan menu yang ada dalam *database*  5.Tampilkan pesan peringatan karena menu akan dihapus  7.Menu dengan id yang dipilih terhapus dari tabel menu  8.Menampilkan menu dari *database* yang belum dihapus |
| Alterate Course | 7.jika id menu yang dimaksud tidak ada maka akan kembali ke halaman hapus menu | |
| Conclusion | Aktor berhasil menghapus menu | |
| Post Condition | Tabel menu berkurang satu | |

* + - * 1. Narasi *Use case* Memasukkan Pesanan

Tabel 4.43 *Use case* masukkan pesanan

|  |  |  |
| --- | --- | --- |
| *Use case* Name | Memasukkan Pesanan | |
| *Use case* Id | 5 | |
| Actor | *Owner* | |
| Description | *Use Case* menggambarkan kegiatan memasukkan menu | |
| Precondition | Aktor telah *login* | |
| Typical Course of Events | Aksi Aktor  1.Pilih menu menu  2.Pilih submenu masukkan pesanan  4.Mengisi form masukkan pesanan  5.Klik submit | Respon Sistem  3.Menampilkan form masukkan pesanan  6.Memasukkan data pesanan kedalam *database* |
| Alterate Course | 6.jika data yang dimasukkan terdapat kesalahan maka akan kembali ke halaman form masukkan pesanan | |
| Conclusion | Aktor berhasil memasukkan pesanan | |
| Post Condition | Pesanan tersimpan dalam *database* | |

* + - * 1. Narasi *Use case* Menampilkan Pesanan

Tabel 4.44 *Use case* menampilkan pesanan

|  |  |  |
| --- | --- | --- |
| *Use case* Name | Menampilkan Pesanan | |
| *Use case* Id | 6 | |
| Actor | *Owner* | |
| Description | *Use Case* menggambarkan kegiatan menampilkan pesanan | |
| Precondition | Aktor telah *login* | |
| Typical Course of Events | Aksi Aktor  1.Pilih menu pesanan  2.Pilih submenu tampilkan pesanan | Respon Sistem  3.Menampilkan pesanan yang ada dalam *database* |
| Alterate Course | 3.jika dalam *database* pesanan tidak ada maka akan menampilkan tabel kosong | |
| Conclusion | Aktor berhasil menampilkan pesanan | |
| Post Condition |  | |

* + - * 1. Narasi *Use case* Menghapus Pesanan

Tabel 4.45 *Use case* menghapus pesanan

|  |  |  |
| --- | --- | --- |
| *Use case* Name | Menghapus pesanan | |
| *Use case* Id | 7 | |
| Actor | *Owner* | |
| Description | *Use Case* menggambarkan kegiatan menghapus pesanan | |
| Precondition | Aktor telah *login* | |
| Typical Course of Events | Aksi Aktor  1.Pilih menu pesanan  2.Pilih submenu hapus pesanan  4.Pilih pesanan yang akan dihapus  6.Pilih ya | Respon Sistem  3.Menampilkan pesanan yang ada dalam *database*  5.Tampilkan pesan peringatan karena pesanan akan dihapus  7. Pesanan dengan id yang dipilih terhapus dari tabel pesanan  8.Menampilkan pesanan dari *database* yang belum dihapus |
| Alterate Course | 7.jika id pesanan yang dimaksud tidak ada maka akan kembali ke halaman hapus pesanan | |
| Conclusion | Aktor berhasil menghapus pesanan | |
| Post Condition | Tabel pesanan berkurang satu | |

* + - * 1. Narasi *Use case* Pencarian Menu

Tabel 4.45 *Use case* pencarian menu

|  |  |  |
| --- | --- | --- |
| *Use case* Name | Pencarian Menu | |
| *Use case* Id | 8 | |
| Actor | *Owner* | |
| Description | *Use Case* menggambarkan kegiatan pencarian menu | |
| Precondition | Aktor telah *login* | |
| Typical Course of Events | Aksi Aktor  1.Pilih menu pencarian  2.Masukkan menu yang akan dicari | Respon Sistem  3.Mencari menu menggunakan binary search dan hasil berupa laporan penjualan perhari. |
| Alterate Course | 3.jika menu yang dimaksud tidak ada maka akan kembali ke halaman pencarian | |
| Conclusion | Aktor berhasil mencari menu | |
| Post Condition |  | |

* + - * 1. Narasi *Use case* Ubah Hasil Pencarian Menu Menjadi PDF

Tabel 4.45 *Use case* ubah hasil pencarian menu menjadi pdf

|  |  |  |
| --- | --- | --- |
| *Use case* Name | Pencarian Menu | |
| *Use case* Id | 9 | |
| Actor | *Owner* | |
| Description | *Use Case* menggambarkan kegiatan ubah hasil pencarian menu menjadi pdf | |
| Precondition | Aktor telah *login* | |
| Typical Course of Events | Aksi Aktor  1.Pilih menu pencarian  2.Masukkan menu yang akan dicari  4.Klik tombol Lihat pdf | Respon Sistem  3.Mencari menu menggunakan binary search dan hasil berupa laporan penjualan perhari.  5.Sistem akan mengubah hasil pencarian menu menjadi pdf |
| Alterate Course | 3.jika menu yang dimaksud tidak ada maka akan kembali ke halaman pencarian | |
| Conclusion | Aktor berhasil mengubah hasil pencarian menjadi pdf | |
| Post Condition |  | |

* + - * 1. Narasi *Use case* Mencari Laporan Keuangan

Tabel 4.45 *Use case* mencari laporan keuangan

|  |  |  |
| --- | --- | --- |
| *Use case* Name | Mencari Laporan Keuangan | |
| *Use case* Id | 10 | |
| Actor | *Owner* | |
| Description | *Use Case* menggambarkan mencari laporan keuangan | |
| Precondition | Aktor telah *login* | |
| Typical Course of Events | Aksi Aktor  1.Pilih menu laporan  2.Pilih submenu cari  3.Masukkan tanggal laporan yang akan dicari | Respon Sistem  4.Mencari laporan berdasarkan tanggal |
| Alterate Course | 4.jika laporan yang dimaksud tidak ada maka akan kembali ke halaman cari | |
| Conclusion | Aktor berhasil mencari laporan berdasarkan tanggal yang dimasukkan | |
| Post Condition |  | |

* + - * 1. Narasi *Use case* Ubah Laporan Keuangan Menjadi PDF

Tabel 4.45 *Use case* ubah laporan keuangan menjadi pdf

|  |  |  |
| --- | --- | --- |
| *Use case* Name | Ubah Laporan Keuangan Menjadi PDF | |
| *Use case* Id | 11 | |
| Actor | *Owner* | |
| Description | *Use Case* menggambarkan kegiatan ubah laporan keuangan menjadi pdf | |
| Precondition | Aktor telah *login* | |
| Typical Course of Events | Aksi Aktor  1.Pilih menu laporan  2.Pilih submenu cari  3.Masukkan tanggal laporan yang akan dicari  5.Klik tombol lihat pdf | Respon Sistem  4.Mencari laporan berdasarkan tanggal  6.Sistem akan merubah laporan menjadi pdf |
| Alterate Course | 4.jika laporan yang dimaksud tidak ada maka akan kembali ke halaman cari | |
| Conclusion | Aktor berhasil mencari laporan berdasarkan tanggal yang dimasukkan | |
| Post Condition |  | |

* + - * 1. Desain Activity diagram

*Activity diagram* menggambarkan aktifitas-aktifitas yang terjadi dalam sistem *clustering*  pegawai

* + - * 1. *Activity diagram* login

Dibawah ini adalah gambar dari aktivitas *Diagram login.* Sebagai berikut:

Gambar 4. 7 *Activity diagram* login

Aktifitas yang terjadi pada *Activity diagram login,* adalah aktifitas actor yang telah terdaftar dalam sistem, diawali dengan aktor mengakses *url login* dan akan menampilkan halaman *login.* Selanjutnya actor menginpu t*username* dan *password* yang telah terdaftar dalam sistem, aktor dapat menekan tombol *submit* untuk masuk kedalam sistem. Sistem akan melakukan proses validasi, apabila *username* atau *password* tidakvalid, maka sistem akan menolak masuk dan akan menampilkan kembali halaman *login* untuk diisi kembali *username* dan *password*, dan jika *username* dan *password* valid, makasistem akan menampilkan halaman *home* dari actor yang melakukan *login*.

* + - * 1. *Activity diagram* Memasukkan Menu

Gambar 4. 8 *Activity diagram* masukkan menu

Aktifitas yang terjadi pada *Activity diagram* Memasukkan Menudiawali dengan aktor memilih menu menu, lalu pilih *submenu* masukkan menu. Sistem akan menampilkan form untuk memasukkan menu, setelah diisi klik submit maka data yang dikirim akan di cek apakah tidak melanggar validasi atau tidak, jika ya maka akan kembali ke form memasukkan menu, jika tidak data menu akan tersimpan dalam *database*.

* + - * 1. *Activity diagram* Menampilkan Menu

Gambar 4. 9 *Activity diagram* menampilkan menu

Aktifitas yang terjadi pada Activity diagram Menampilkan Menu diawali dengan aktor memilih menu menu, lalu pilih submenu tampilkan menu. Sistem akan menampilkan seluruh menu dari *database* pada halaman tampil menu.

* + - * 1. *Activity diagram* Mengubah Menu

Gambar 4. 10 *Activity diagram* mengubah menu

Aktifitas yang terjadi pada Activity diagram Mengubah Menu diawali dengan aktor memilih menu menu, lalu pilih submenu ubah menu. Sistem akan menampilkan seluruh menu dari *database* pada halaman ubah menu. *Owner* akan memilih menu yang ingin dirubah setelah dipilih, sistem akan mencari menu pada *database* berdasarkan id\_menu yang telah dipilih. Setelah mendapatkan menu berdasarkan id\_menu yang telah dipilih sistem akan menampilkan kesuluruhan data pada form ubah data menu. Selanjutnya *admin* mengubah salah satu atau semua data menu, setelah selesai mengubah *admin* klik ubah. Setelah itu sistem akan melakukan validasi terhadap data menu yang telah dirubah jika ada yang salah maka akan kembali ke form ubah menu, jika tidak ada yang salah maka akan tersimpan dalam *database*.

* + - * 1. *Activity diagram* Menghapus Menu

Gambar 4. 11 *Activity diagram* menghapus menu

Aktifitas yang terjadi pada *Activity diagram* Menghapus Menudiawali dengan aktor memilih menu menu, lalu pilih *submenu* hapus menu. Sistem akan menampilkan seluruh menu dari *database* pada halaman hapus menu. *Owner* akan memilih menu yang ingin dihapus setelah dipilih, sistem akan memberikan peringatan serta pertanyaan apakah menu dengan id\_menu yang dipilih ingin dihapus jika menjawab tidak maka akan kembali ke halaman hapus menu dan jika ya maka akan dihapus dari *database*.

* + - * 1. *Activity diagram* Memasukkan Pesanan

Gambar 4. *Activity diagram* masukkan menu

Aktifitas yang terjadi pada *Activity diagram* Memasukkan Pesanandiawali dengan aktor memilih menu pesanan, lalu pilih *submenu* masukkan pesanan. Sistem akan menampilkan form untuk memasukkan pesanan, setelah diisi klik submit maka data yang dikirim akan di cek apakah tidak melanggar validasi atau tidak, jika ya maka akan kembali ke form memasukkan pesanan, jika tidak data pesanan akan tersimpan dalam *database*.

* + - * 1. *Activity diagram* Menampilkan Pesanan

Gambar 4. *Activity diagram* menampilkan pesanan

Aktifitas yang terjadi pada Activity diagram Menampilkan Pesanan diawali dengan aktor memilih menu pesanan, lalu pilih submenu tampilkan pesanan. Sistem akan menampilkan seluruh pesanan dari *database* pada halaman tampil pesanan.

* + - * 1. *Activity diagram* Mengubah Pesanan

Gambar 4. *Activity diagram* mengubah pesanan

Aktifitas yang terjadi pada Activity diagram Mengubah Pesanan diawali dengan aktor memilih menu pesanan, lalu pilih submenu ubah pesanan. Sistem akan menampilkan seluruh pesanan dari *database* pada halaman ubah pesanan. *Owner* akan memilih pesanan yang ingin dirubah setelah dipilih, sistem akan mencari pesanan pada *database* berdasarkan id\_pesanan yang telah dipilih. Setelah mendapatkan pesanan berdasarkan id\_pesanan yang telah dipilih sistem akan menampilkan kesuluruhan data pada form ubah data pesanan. Selanjutnya *admin* mengubah salah satu atau semua data pesanan, setelah selesai mengubah *admin* klik ubah. Setelah itu sistem akan melakukan validasi terhadap data pesanan yang telah dirubah jika ada yang salah maka akan kembali ke form ubah pesanan, jika tidak ada yang salah maka akan tersimpan dalam *database*.

* + - * 1. *Activity diagram* Menghapus Pesanan

Gambar 4. *Activity diagram* menghapus pesanan

Aktifitas yang terjadi pada *Activity diagram* Menghapus Pesanandiawali dengan aktor memilih menu pesanan, lalu pilih *submenu* hapus pesanan. Sistem akan menampilkan seluruh pesanan dari *database* pada halaman hapus pesanan. *Owner* akan memilih pesanan yang ingin dihapus setelah dipilih, sistem akan memberikan peringatan serta pertanyaan apakah pesanan dengan id\_pesanan yang dipilih ingin dihapus jika menjawab tidak maka akan kembali ke halaman hapus pesanan dan jika ya maka akan dihapus dari *database*.

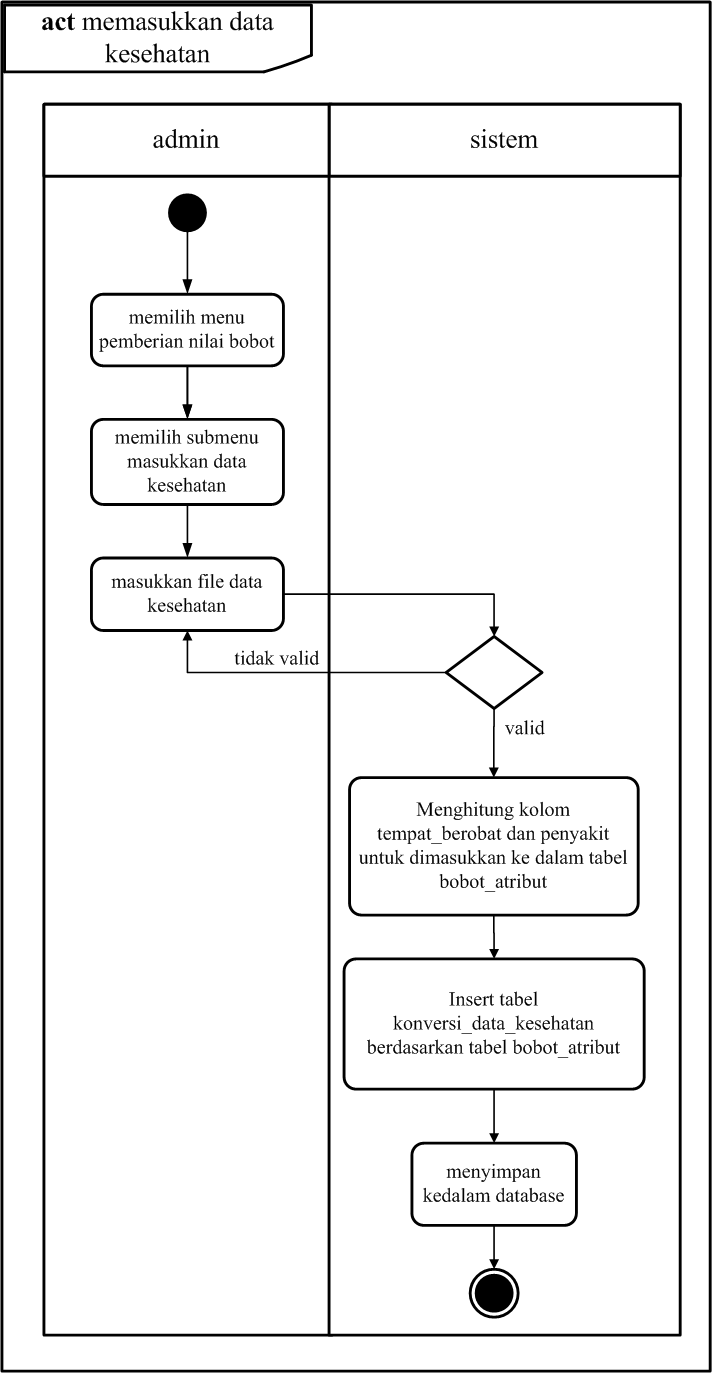
* + - * 1. *Activity diagram* Pencarian Menu

Gambar 4. *Activity diagram* menghapus pesanan

Aktifitas yang terjadi pada *Activity diagram* Pencarian Menudiawali dengan aktor memilih menu pencarian. Sistem akan menampilkan kolom pencarian pada halaman pencarian. *Owner, Admin* dan *Customer* akan memasukkan menu yang akan dicari dan sistem akan mencari menu menggunakan binary search dan menampilkan hasil pencarian berupa tabel

akan memilih pesanan yang ingin dihapus setelah dipilih, sistem akan memberikan peringatan serta pertanyaan apakah pesanan dengan id\_pesanan yang dipilih ingin dihapus jika menjawab tidak maka akan kembali ke halaman hapus pesanan dan jika ya maka akan dihapus dari *database*.

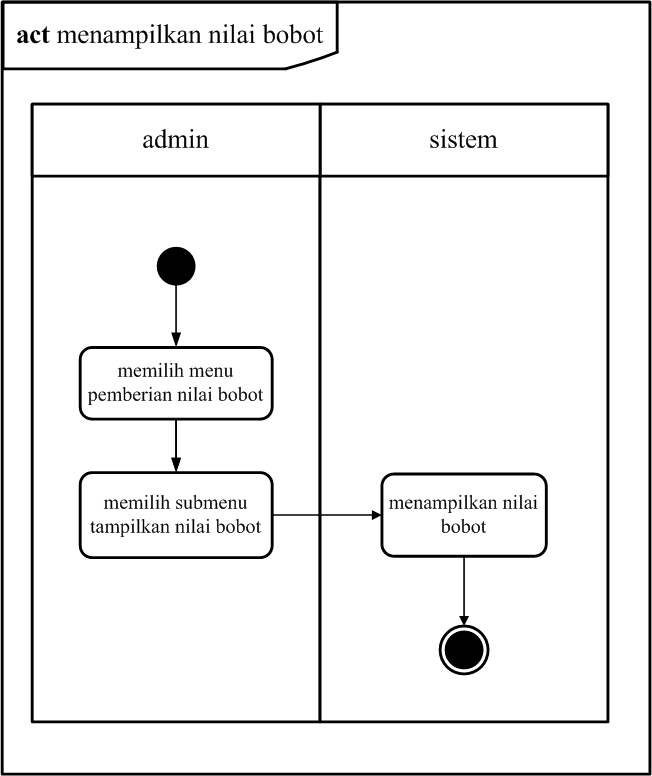
* + - * 1. *Activity diagram* Memasukkan Data Kesehatan



Gambar 4. 12 *Activity diagram* masukkan data kesehatan

Aktifitas yang terjadi pada *Activity diagram* Memasukkan Data Kesehatan diawali dengan aktor memilih menu bobot atribut, lalu pilih *submenu* masukkan *database* kesehatan. Sistem akan menampilkan *textboxt* untuk memasukkan *file* *database* kesehatan yang telah dipersiapkan, setelah memilih *file* *database* kesehatan klik masukkan. File akan dikirim ke sistem untuk validasi, jika terdapat data yang kosong maka akan kembali ke halaman masukkan *database* kesehatan dan jika tidak sistem akan melakukan tiga proses insert kedalam tiga tabel yaitu tabel bobot\_atribut, konversi\_data\_kesehatan dan data\_kesehatan. Pertama dari *file* *database* kesehatan yang masuk akan dihitung jumlah tempat berobat dan penyakit setelah itu diberikan nilai dan dimasukkan kedalam *database*. Selanjutnya membuat data untuk tabel konversi\_data\_kesehatan, berdasarkan nilai dari tabel bobot\_atribut dan *file* *database* kesehatan maka terbentuklah sebuah data konversi *database* kesehatan dengan tipe data kolom rawat\_jalan, rawat\_inap, tempat\_berobat dan penyakit adalah numeric bukan nominal. Selanjutnya *file* *database* kesehatan akan dimasukkan kedalam tabel data\_kesehatan.

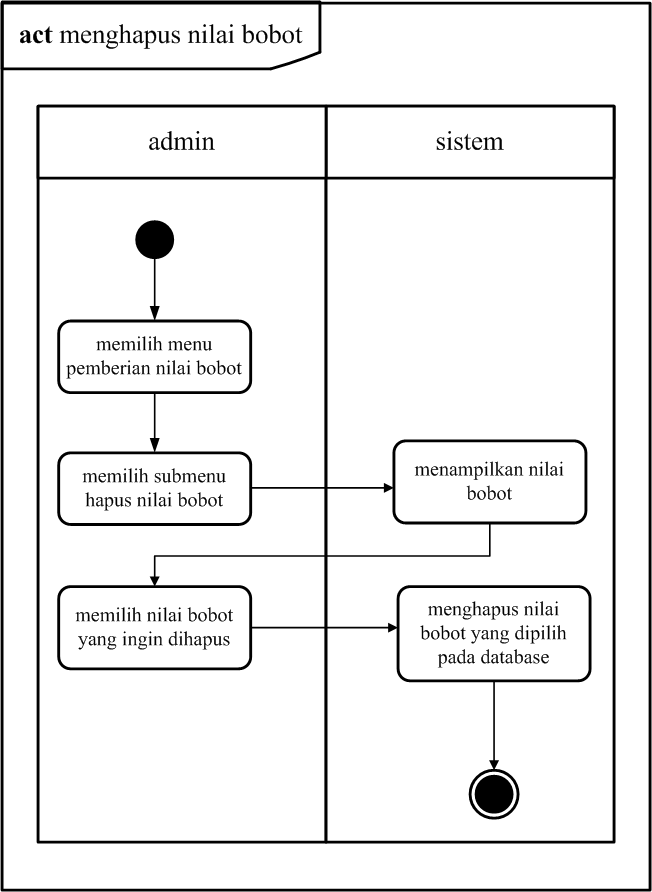
* + - * 1. *Activity diagram*Menampilkan Bobot Atribut



Gambar 4. 13 *Activity diagram* menampilkan bobot atribut

Aktifitas yang terjadi pada *Activity diagram* Menampilkan Bobot Atribut diawali dengan aktor memilih menu bobot atribut, lalu pilih *submenu* tampilkan nilai bobot. Sistem akan menampilkan seluruh bobot atribut dari *database* pada halaman tampil nilai bobot.

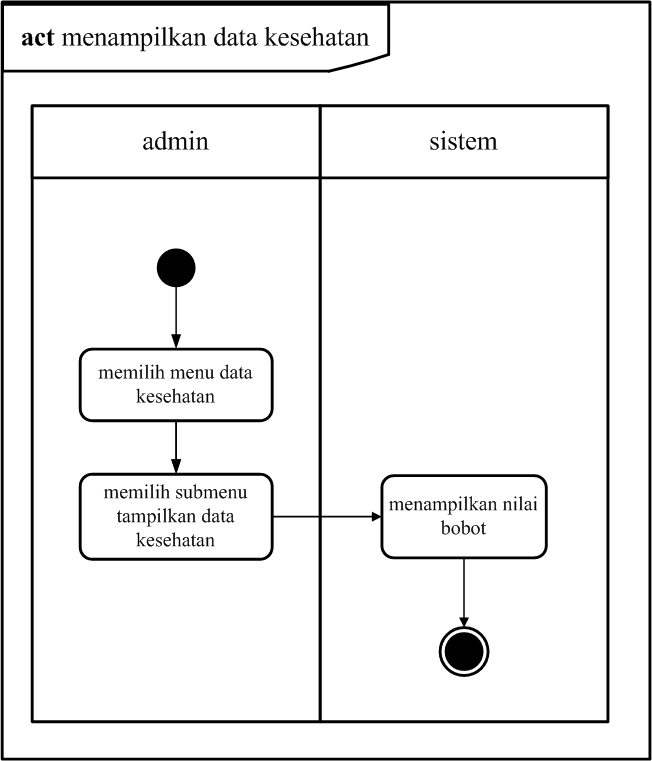
* + - * 1. *Activity diagram*Menghapus Bobot Atribut



Gambar 4. 14 *Activity diagram* menghapus bobot atribut

Aktifitas yang terjadi pada *Activity diagram* Menghapus Bobot Atribut diawali dengan aktor memilih menu bobot atribut, lalu pilih *submenu* hapus bobot atribut. Sistem akan menampilkan seluruh bobot atribut dari *database* pada halaman hapus bobot atribut. *Owner* akan memilih bobot atribut dengan input\_ke berapa yang ingin dihapus setelah dipilih, sistem akan memberikan peringatan serta pertanyaan apakah bobot atribut dengan input\_ke yang dipilih ingin dihapus jika menjawab tidak maka akan kembali ke halaman hapus bobot atribut dan jika ya maka akan dihapus dari *database*.

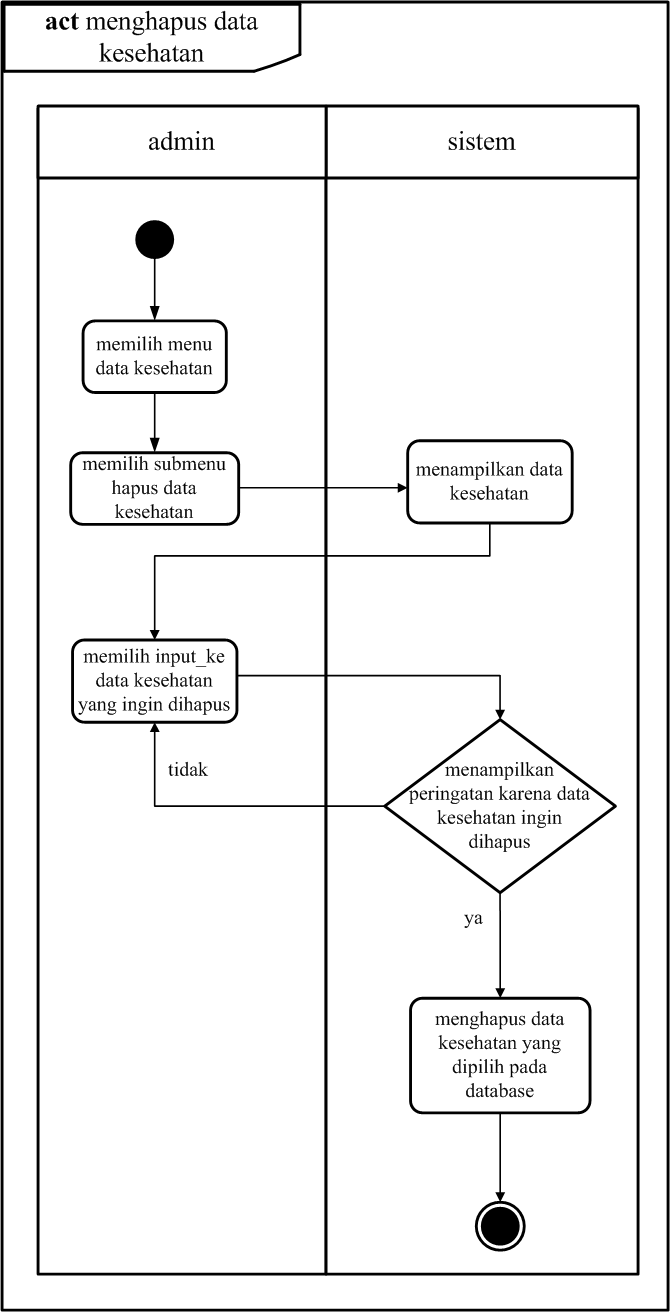
* + - * 1. *Activity diagram*Menampilkan Data Kesehatan



Gambar 4. 15 *Activity diagram* menampilkan data kesehatan

Aktifitas yang terjadi pada *Activity diagram* Menampilkan Data Kesehatan diawali dengan aktor memilih menu *database* kesehatan, lalu pilih submenu tampilkan *database* kesehatan. Sistem akan menampilkan seluruh *database* kesehatandari *database* pada halaman tampil *database* kesehatan.

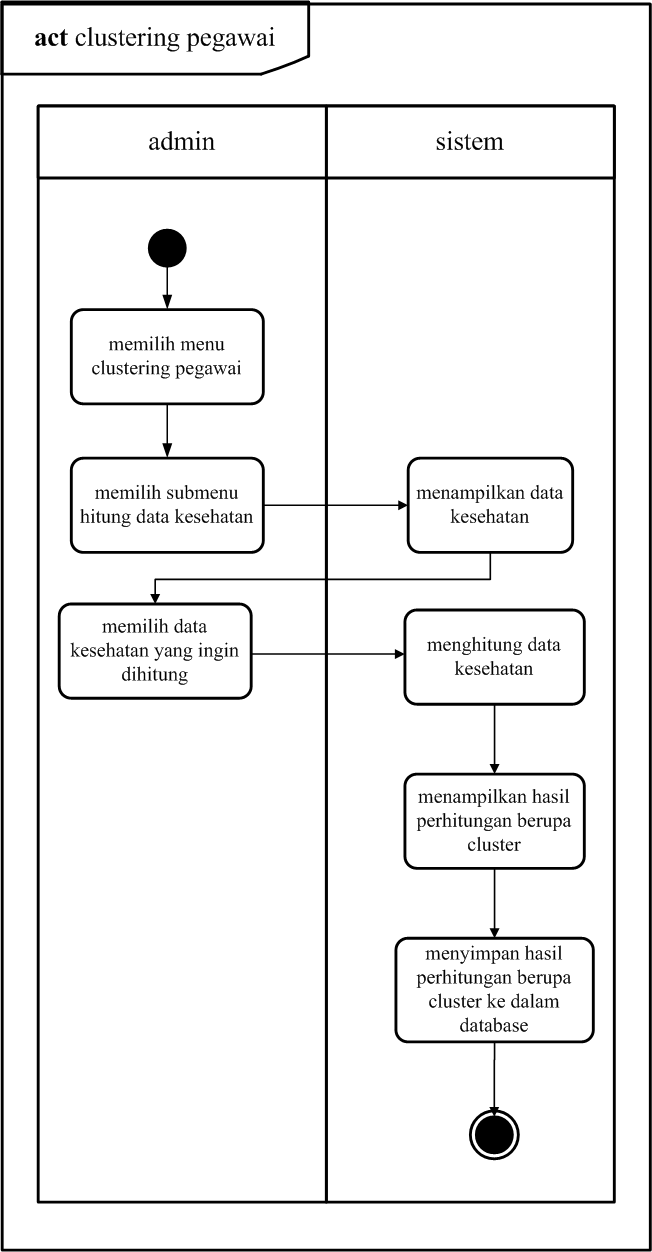
* + - * 1. *Activity diagram* Menghapus Data Kesehatan



Gambar 4. 16 *Activity diagram* menghapus data kesehatan

Aktifitas yang terjadi pada *Activity diagram* Menghapus Data Kesehatan diawali dengan aktor memilih menu *database* kesehatan, lalu pilih submenu hapus *database* kesehatan. Sistem akan menampilkan seluruh *database* kesehatandari *database* pada halaman hapus *database* kesehatan. Aktor akan memilih *database* kesehatan yang ingin dihapus berdasarkan input\_ke, setelah dipilih sistem akan memberikan peringatan serta pertanyaan apakah *database* kesehatan dengan input\_ke yang dipilih ingin dihapus jika menjawab tidak maka akan kembali ke halaman hapus *database* kesehatan dan jika ya maka akan dihapus dari *database*.

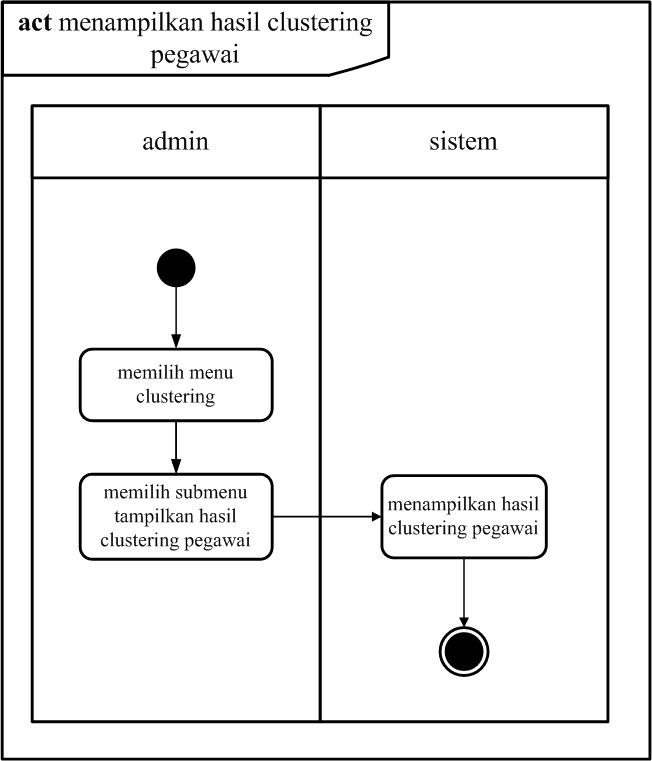
* + - * 1. *Activity diagramClustering*  Pegawai



Gambar 4. 17 *Activity diagram* *clustering*  pegawai

Aktifitas yang terjadi pada *Activity diagram* *Clustering*  Pegwai diawali dengan aktor memilih menu *clustering*  pegawai, lalu pilih submenu *clustering*  pegawai. Sistem akan menampilkan *database* kesehatan berdasarkan input\_ke, selanjutnya aktor memilih *database* kesehatan yang ingin dijadikan data training. Sistem akan melaksanakan perhitungan *clustering*  pegawai menggunakan metode *K-Means++, K-Means, COP-KMeans*. Setelah selesai menghitung hasil akan disimpan pada *database* lalu hasil akan ditampilkan pada halaman *clustering* .

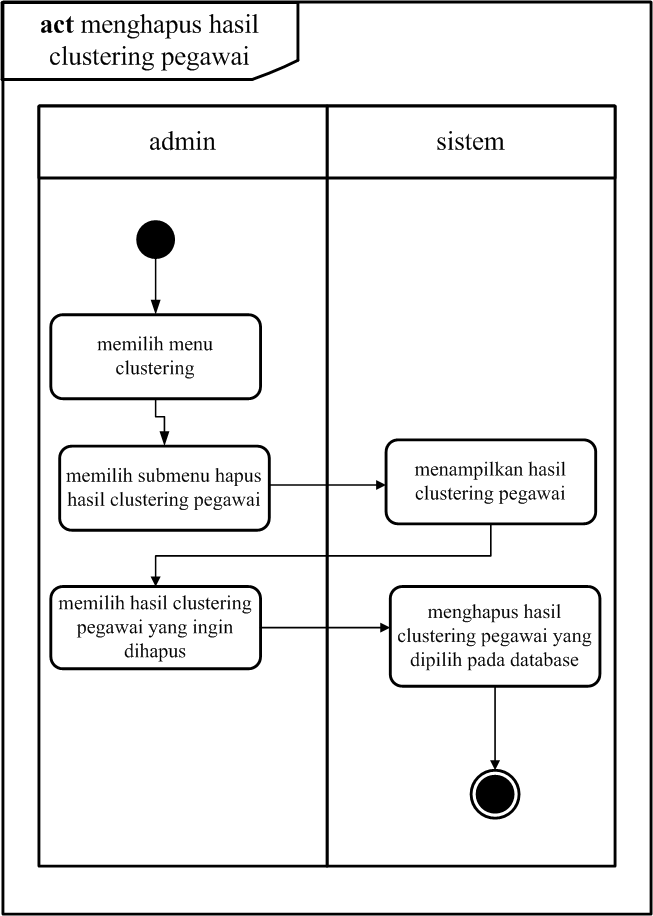
* + - * 1. *Activity diagram* Menampilkan Hasil *Clustering*  Pegawai



Gambar 4. 18 *Activity diagram* menampilkan hasil *clustering*  pegawai

Aktifitas yang terjadi pada *Activity diagram* Menampilkan Hasil *Clustering*  Pegawai diawali dengan aktor memilih menu *clustering* , lalu pilih submenu tampilkan hasil *clustering*  pegawai. Sistem akan menampilkan seluruh hasil *clustering*  pegawaidari *database* pada halaman tampil *clustering*  pegawai.

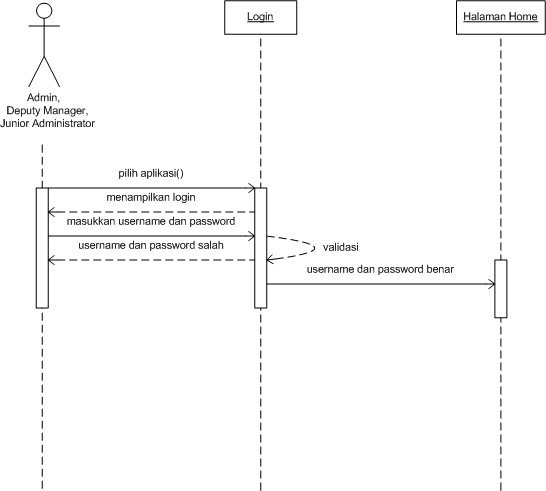
* + - * 1. *Activity diagram* Menghapus Hasil *Clustering*  Pegawai



Gambar 4. 19 *Activity diagram* menghapus hasil *clustering*  pegawai

Aktifitas yang terjadi pada *Activity diagram* Menghapus Hasil *Clustering*  Pegawai diawali dengan aktor memilih menu *clustering* , lalu pilih submenu hapus hasil *clustering*  pegawai. Sistem akan menampilkan seluruh hasil *clustering*  pegawai dari *database* pada halaman hapus hasil *clustering*  pegawai. Aktor akan memilih hasil *clustering*  pegawai yang ingin dihapus setelah dipilih, sistem akan memberikan peringatan serta pertanyaan apakah hasil *clustering*  pegawai dengan input\_ke yang dipilih ingin dihapus jika menjawab tidak maka akan kembali ke halaman hapus hasil *clustering*  pegawai dan jika ya maka akan dihapus dari *database*.

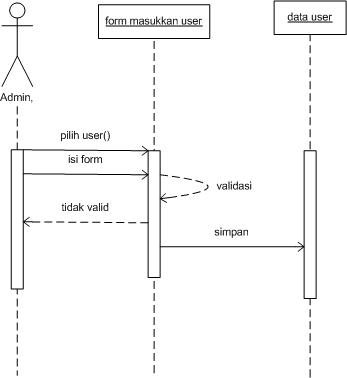
* + - * 1. Desain Sequence Diagram
        2. Sequence Diagram Login



Gambar 4. 20 *Sequence diagaram*  login

*Sequence diagram* diatas menjelaskan actor saat melakukan *login.* Terdapat tiga actor yang melakukan *login,* yaitu *Owner, Admin, Customer.* Pertama kali aktor mengetikkan *username* dan *password* pada halaman*login*, maka sistem akan memvalidasi, jika *username* dan *password* benar, maka akan menampilkan halaman *home,* jika *username* dan *password* salah, maka sistem akan menampilkan halaman *login* untuk kembali melakukan *login*.

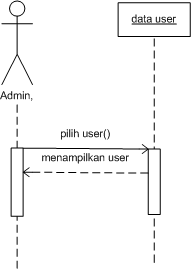
* + - * 1. Sequence Diagram Memasukkan Menu



Gambar 4. 21 *Sequence diagaram*  masukkan menu

*Sequence diagram* diatas menjelaskan actor saat memasukkan menu*.* Aktor yang dapat mengelola menuadalah *Owner.* Pertama kaliaktor memilih menu lalu tampil form masukkan menu, selanjutnya aktor memasukkan data menu, pilih tombol masukkan untuk post ke sistem. Setelah data menu dikirim ke sistem akan dilakukan validasi untuk mencegah adanya data yang tidak diinginkan, jika hasil dari validasi adalah tidak valid maka akan kembali ke halaman form masukkan menu, jika hasil dari validasi adalah valid data akan disimpan ke dalam *database*.

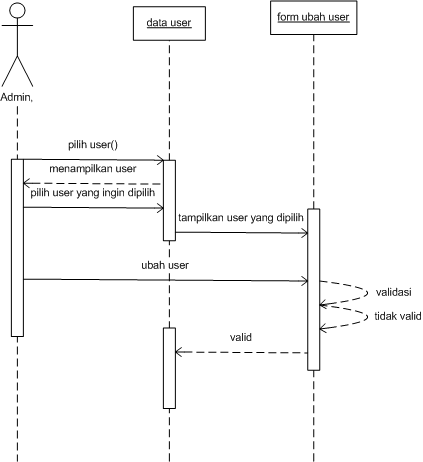
* + - * 1. Sequence Diagram Menampilkan Menu



Gambar 4. 22 *Sequence diagaram*  menampilkan menu

*Sequence diagram* diatas menjelaskan actor saat menampilkan menu*.* Aktor yang dapat menampilkan menuadalah *Owner.* Pertama kaliaktor memilih menu lalu akan tampil seluruh menu dari *database*.

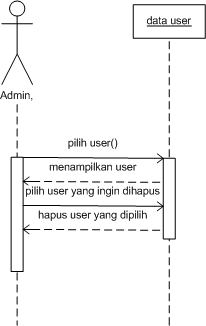
* + - * 1. Sequence Diagram Mengubah Menu



Gambar 4. 23 *Sequence diagaram*  mengubah menu

*Sequence diagram* diatas menjelaskan actor saat mengubah menu*.* Aktor yang dapat mengubah menuadalah *Owner.* Pertama kali aktor memilih menu lalu akan tampil seluruh menu dari *database*, setelah muncul pilih menu yang akan dirubah. Sistem akan mengambil id\_menu yang telah dipilih dan memunculkan seluruh data menu berdasarkan id\_menu yang telah dipilih ke dalam form ubah menu. Selanjutnya aktor akan mengubah data menu setelah selesai klik ubah. Ketika data dikirim sistem akan melakukan validasi terhadap data yang dikirim jika hasil validasi adalah salah maka akaan kembali ke form ubah menu.

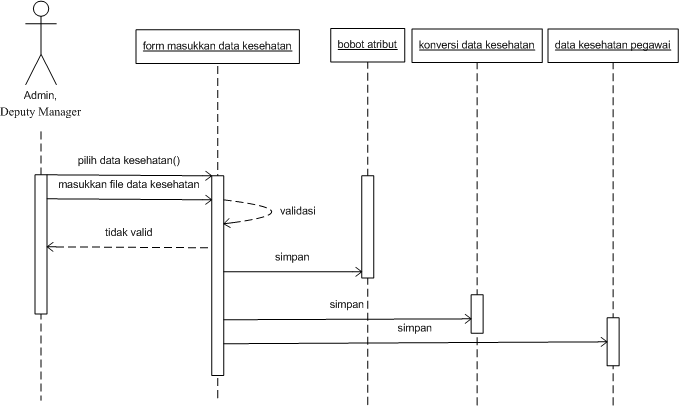
* + - * 1. Sequence Diagram Menghapus Menu



Gambar 4. 24 *Sequence diagaram*  mengapus menu

*Sequence diagram* diatas menjelaskan actor saat menghapus menu*.* Aktor yang dapat menghapu s*user* adalah *Owner.* Pertama kaliaktor memilih menu lalu akan tampil seluruh menu dari *database*, setelah muncul pilih menu yang akan dihapus. Sistem akan mengambil id\_menu yang telah dipilih dan memunculkan peringatan karena menu akan dihapus jika aktor menjawab tidak maka akan kembali ke halaman hapus menu, jika ya maka menu berdasarkan id\_menu yang dipilih akan dihapus dari *database*.

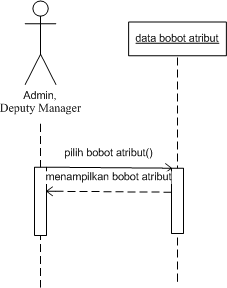
* + - * 1. Sequence Diagram Masukkan Data Kesehatan



Gambar 4. 25 *Sequence diagaram*  masukkan data kesehatan

*Sequence diagram*diatasmenjelaskanactorsaatmemasukkan *database* kesehatan*.*Aktoryang dapatmengelola*user* adalah*Admin dan Admin.* Pertama kaliaktor memilih *database* kesehatan lalu tampil *textboxt* untuk memasukkan *file* *database* kesehatan, selanjutnya aktor memasukkan *file* *database* kesehatan yang telah dipilih, pilih tombol masukkan untuk post ke sistem. Setelah *file* *database* kesehatan dikirim ke sistem akan dilakukan validasi untuk mencegah adanya *cell* yang kosong, jika hasil dari validasi adalah tidak valid maka akan kembali ke halaman masukkan *database* kesehatan, jika hasil dari validasi adalah valid sistem akan melakukan tiga proses insert kedalam tiga tabel yaitu tabel bobot\_atribut, konversi\_data\_kesehatan dan data\_kesehatan. Pertama dari *file* *database* kesehatan yang masuk akan dihitung jumlah tempat berobat dan penyakit setelah itu diberikan nilai dan dimasukkan kedalam *database*. Selanjutnya membuat data untuk tabel konversi\_data\_kesehatan, berdasarkan nilai dari tabel bobot\_atribut dan *file* *database* kesehatan maka terbentuklah sebuah data konversi *database* kesehatan dengan tipe data kolom rawat\_jalan, rawat\_inap, tempat\_berobat dan penyakit adalah numeric bukan nominal. Selanjutnya *file* *database* kesehatan akan dimasukkan kedalam tabel data\_kesehatan.

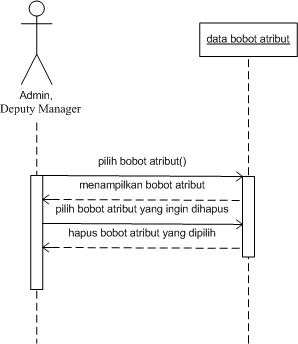
* + - * 1. Sequence Diagram Menampilkan Bobot Atribut



Gambar 4. 26 *Sequence diagaram*  menampilkan bobot atribut

Sequence diagram diatas menjelaskan aktor saat menampilkan bobot atribut. Aktor yang dapat menampilkan bobot atribut adalah *Owner dan Admin*. Pertama kali aktor memilih bobot atribut lalu akan tampil seluruh bobot atribut dari *database*.

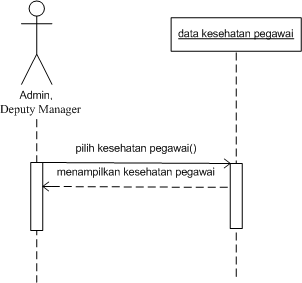
* + - * 1. Sequence Diagram Menghapus Bobot Atribut



Gambar 4. 27 *Sequence diagaram*  menghapus bobot atribut

Sequence diagram diatas menjelaskan aktor saat menghapus bobot atribut. Aktor yang dapat menghapusbobot atribut adalah *Owner dan Admin*. Pertama kali aktor memilih bobot atribut lalu akan tampil seluruh bobot atribut dari *database*, setelah muncul pilih input\_ke bobot atribut yang akan dihapus. Sistem akan mengambil input\_ke yang telah dipilih dan memunculkan peringatan karena bobot atribut akan dihapus jika aktor menjawab tidak maka akan kembali ke halaman hapus bobot atribut, jika ya maka bobot atribut berdasarkan input\_ke yang dipilih akan dihapus dari *database*.

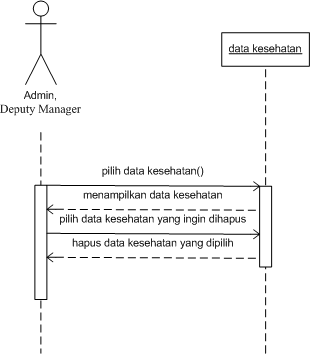
* + - * 1. Sequence Menampilkan Data Kesehatan



Gambar 4. 28 *Sequence diagaram*  menampilkan data kesehatan

Sequence diagram diatas menjelaskan aktor saat menampilkan *database* kesehatan. Aktor yang dapat menampilkan *database* kesehatan adalah *Owner dan Admin*. Pertama kali aktor memilih *database* kesehatan lalu akan tampil seluruh *database* kesehatan dari *database*.

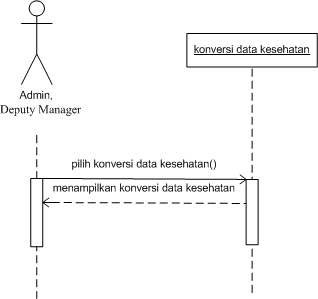
* + - * 1. Sequence Diagram Menghapus Data Kesehatan



Gambar 4. 29 *Sequence diagaram*  menghapus data kesehatan

Sequence diagram diatas menjelaskan aktor saat menghapus *database* kesehatan. Aktor yang dapat menghapus menu adalah *Owner dan Admin*. Pertama kali aktor memilih *database* kesehatan lalu akan tampil seluruh *database* kesehatan dari *database*, setelah muncul pilih *database* kesehatanyang akan dihapus. Sistem akan mengambil input\_ke yang telah dipilih dan memunculkan peringatan karena *database* kesehatan akan dihapus jika aktor menjawab tidak maka akan kembali ke halaman hapus *database* kesehatan, jika ya maka *database* kesehatan berdasarkan input\_ke yang dipilih akan dihapus dari *database*.

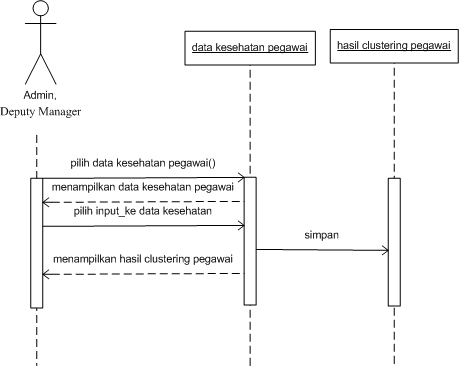
* + - * 1. Sequence Diagram Menampilkan Konversi Data Kesehatan



Gambar 4. 30 *Sequence diagaram*  menampilkan konversi data kesehatan

Sequence diagram diatas menjelaskan aktor saat menampilkan konversi *database* kesehatan. Aktor yang dapat menampilkan konversi *database* kesehatan adalah *Owner dan Admin*. Pertama kali aktor memilih konversi *database* kesehatan lalu akan tampil seluruh konversi *database* kesehatan dari *database*.

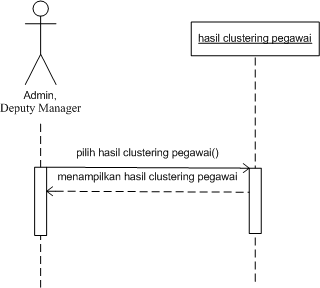
* + - * 1. Sequence Diagram *Clustering*  Pegawai



Gambar 4. 31 *Sequence diagaram*  *clustering*  pegawai

Sequence diagram diatas menjelaskan aktor saat *clustering*  pegawai. Aktor yang berhak mengakses *clustering*  pegawai adalah *Owner, Admin dan Customer*. Sistem akan menampilkan *database* kesehatan berdasarkan input\_ke, selanjutnya aktor memilih *database* kesehatan yang ingin dijadikan data training. Sistem akan melaksanakan perhitungan *clustering*  pegawai menggunakan metode *K-Means++, K-Means, COP-KMeans*. Setelah selesai menghitung hasil akan disimpan pada *database* lalu hasil akan ditampilkan pada halaman *clustering* .

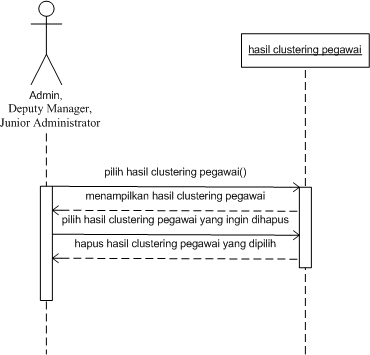
* + - * 1. Sequence Diagram Menampilkan Hasil *Clustering*  Pegawai



Gambar 4. 32 *Sequence diagaram*  *clustering*  pegawai

Sequence diagram diatas menjelaskan aktor saat menampilkan hasil *clustering*  pegawai. Aktor yang dapat menampilkan hasil *clustering*  pegawai adalah *Owner, Admin dan Customer*. Pertama kali aktor memilih hasil *clustering*  pegawai lalu akan tampil seluruh hasil *clustering*  pegawai dari *database*.

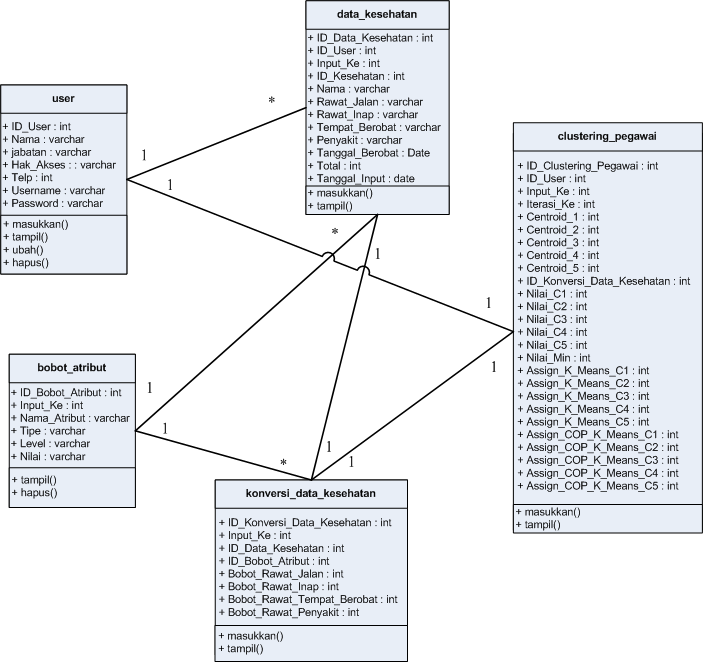
* + - * 1. Sequence Diagram Menghapus Hasil *Clustering*  Pegawai



Gambar 4. 33 *Sequence diagaram*  menghapus hasil *clustering*  pegawai

Sequence diagram diatas menjelaskan aktor saat menghapus hasil *clustering*  pegawai. Aktor yang dapat menghapus menu adalah *Owner, Admin dan Customer*. Pertama kali aktor memilih hasil *clustering*  pegawai lalu akan tampil seluruh hasil *clustering*  pegawai dari *database*, setelah muncul pilih hasil *clustering*  pegawai yang akan dihapus. Sistem akan mengambil input\_ke yang telah dipilih dan memunculkan peringatan karena hasil *clustering*  pegawai akan dihapus jika aktor menjawab tidak maka akan kembali ke halaman hapus hasil *clustering*  pegawai, jika ya maka hasil *clustering*  pegawai berdasarkan id\_menu yang dipilih akan dihapus dari *database*.

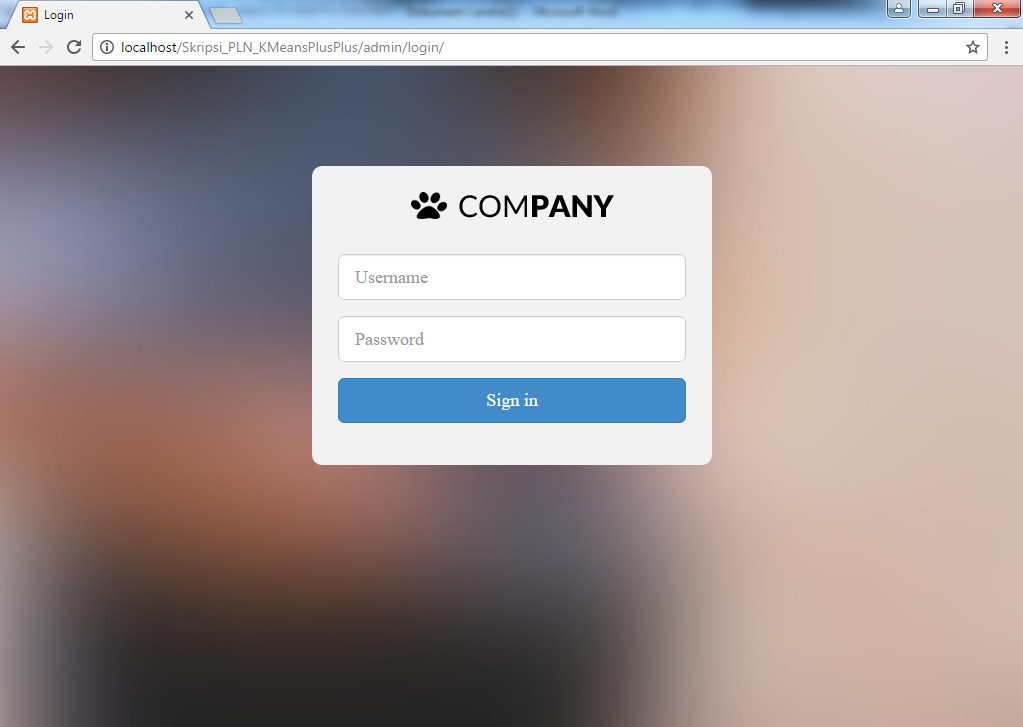
1. Desain *Class* Diagram



Gambar 4. 34 *Class* diagram

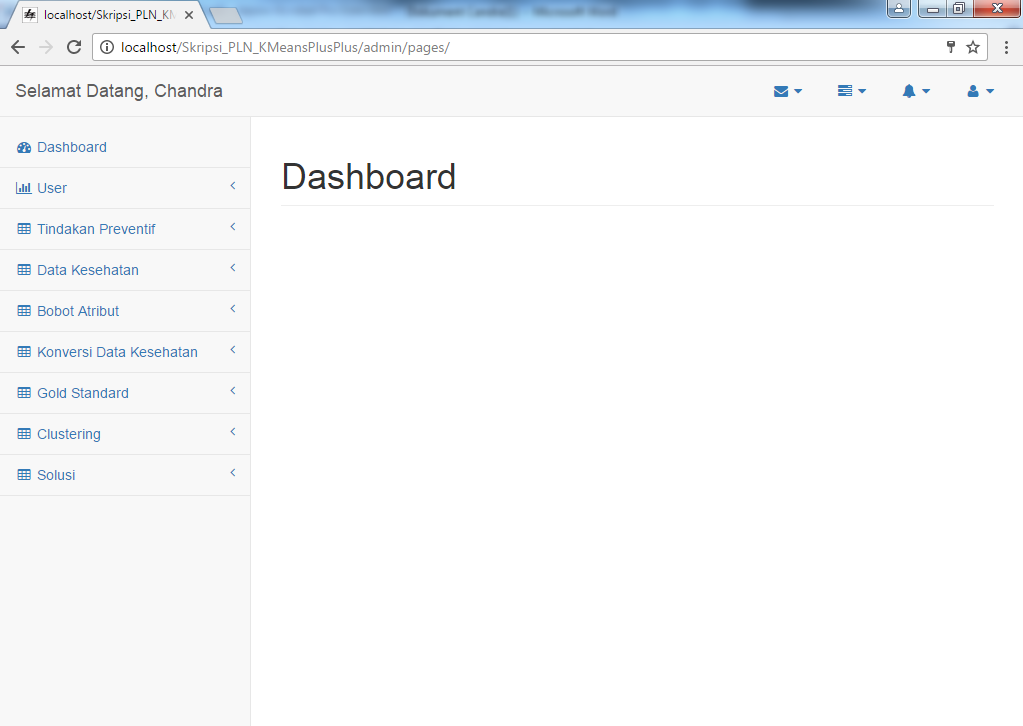
Keterangan :

1. Class menu memiliki hubungan 1 *(one)* to \* *(many)* dengan class data\_kesehatan, artinya menu dapat memiliki banyak data\_kesehatan.
2. Class menu memiliki hubungan 1 *(one)* to 1 *(one)* dengan class *cluster*ing\_pegawai, artinya menu hanya dapat menghitung 1 kali *cluster*ing.
3. Class data\_kesehatan memiliki hubungan \* *(many)* to 1 *(one)* dengan class bobot\_atribut, artinya masing-masing dari rawat jalan, rawat inap, tempat berobat akan memiliki 1 bobot atribut.
4. Class bobot\_atribut memiliki hubungan 1 *(one)* to 1 *(one)* dengan class konversi\_data\_kesehatan, artinya bobot\_atribut dapat memiliki 1 data konversi data kesehatan.
5. Class data\_kesehatan memiliki hubungan 1 *(one)* to 1 *(one)* dengan class konversi\_data\_kesehatan, artinya data\_kesehatan memiliki 1 konversi data kesehatan dimana rawat jalan, rawat inap, tempat berobat telah dirubah menjadi angka.
6. Class konversi\_data\_kesehatan memiliki hubungan 1 *(one)* to 1 *(one)* dengan class *cluster*ing\_pegawai, artinya 1 konversi\_data\_kesehatan akan memiliki 1 kelompok pada 1 *cluster*ing\_pegawai.
7. Desain Interface
8. Login



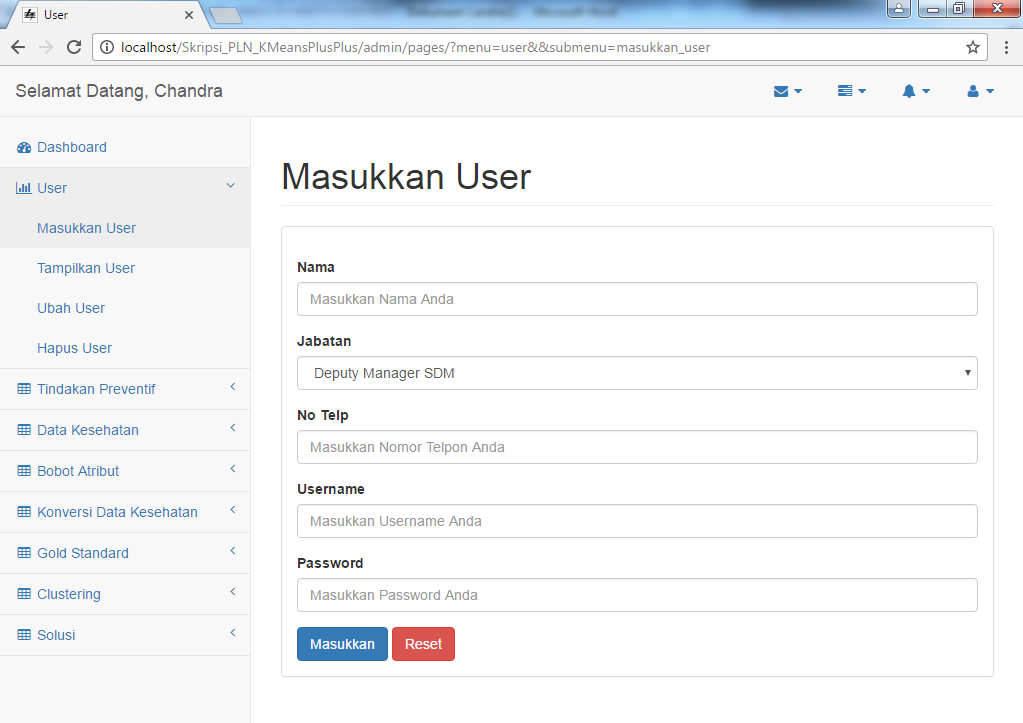
Gambar 4. 35 Login

1. Halaman Menu Utama



Gambar 4. 36 Halaman Menu Utama

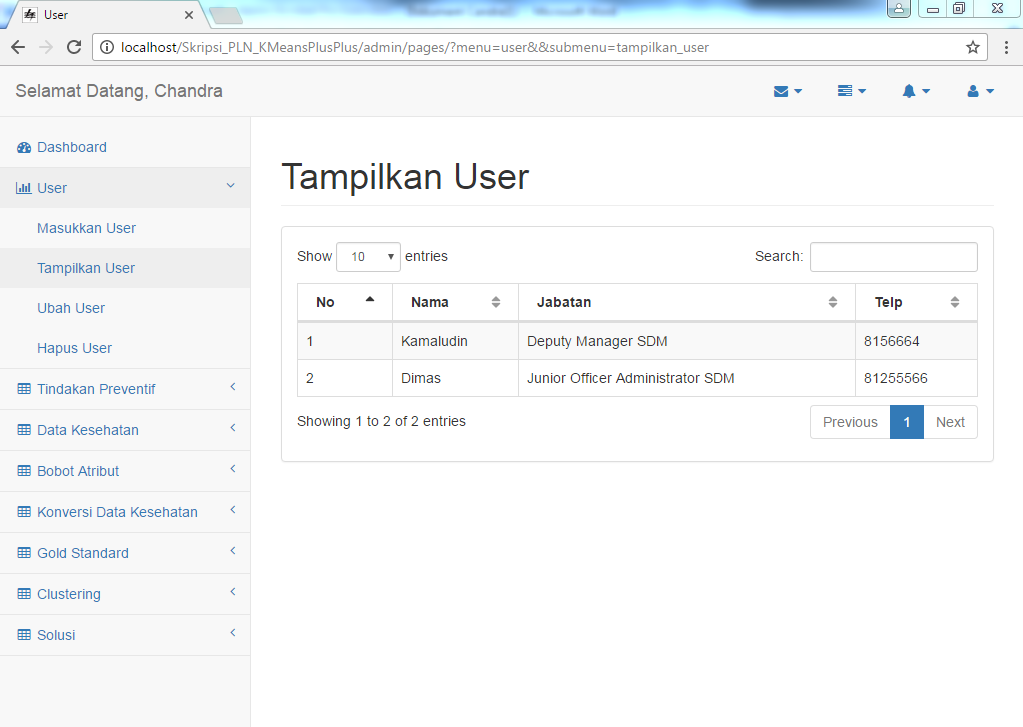
1. Halaman Memasukkan Menu



Gambar 4. 37 Halaman Manage Menu (Masukan Menu)

Menu masukkan menu berguna untuk memasukkan *Customer* dan *Admin*.

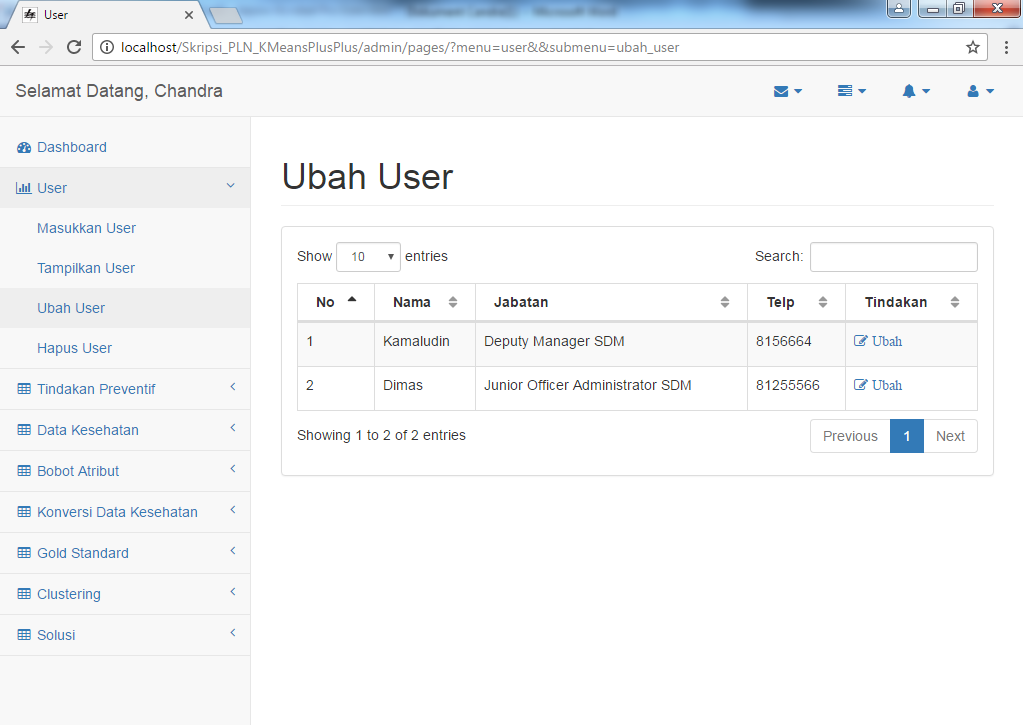
1. Halaman Tampilkan Menu



Gambar 4. 38 Halaman Tampilkan Menu

Tampilan menu yang diberikan hak akses kedalam Aplikasi Perencanaan Program Kesehatan Pegawai.

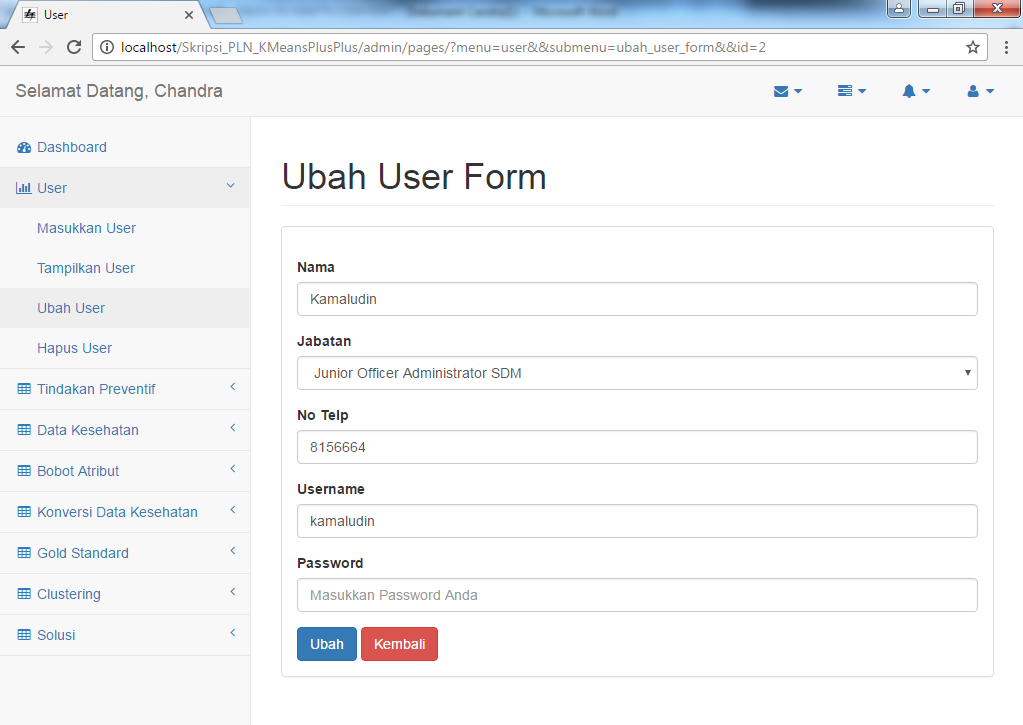
1. Halaman Ubah Menu



Gambar 4. 39 Halaman Ubah Menu

Jika terdapat kesalahan ketika memasukkan data menu maka dapat di ubah pada menu ubah menu. Dengan cara klik ‘Ubah’ pada sisi kanan Aplikasi PPKP.

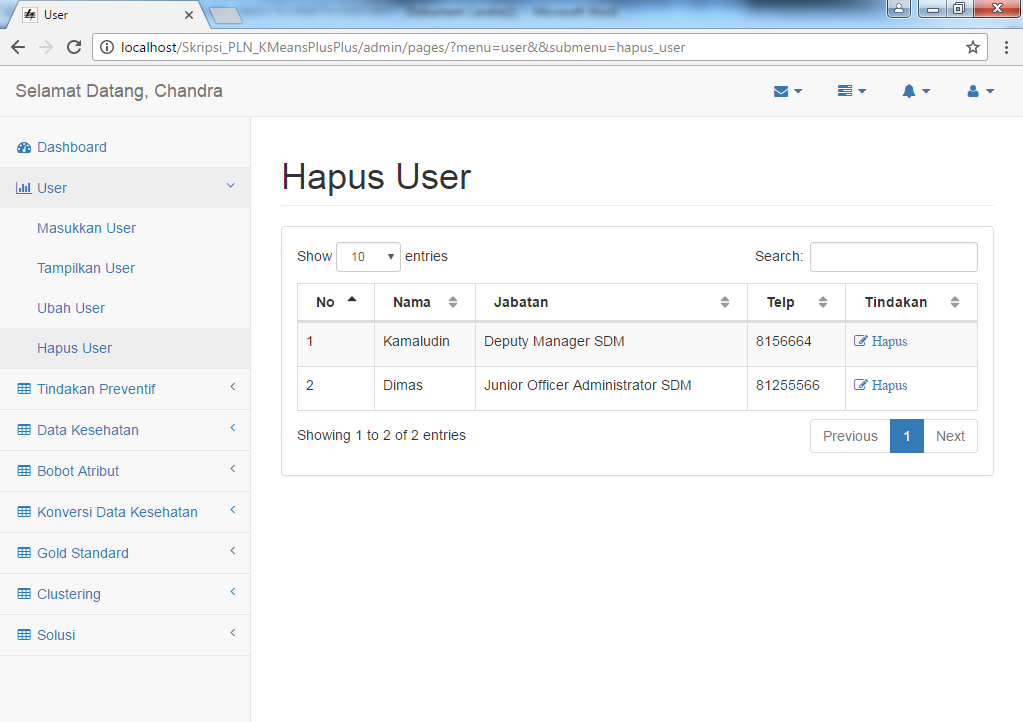
1. Halaman Ubah Menu Form



Gambar 4. 40 Form Ubah Menu

Form ubah menu digunakan untuk mengubah data menu yang salah pada *database*.

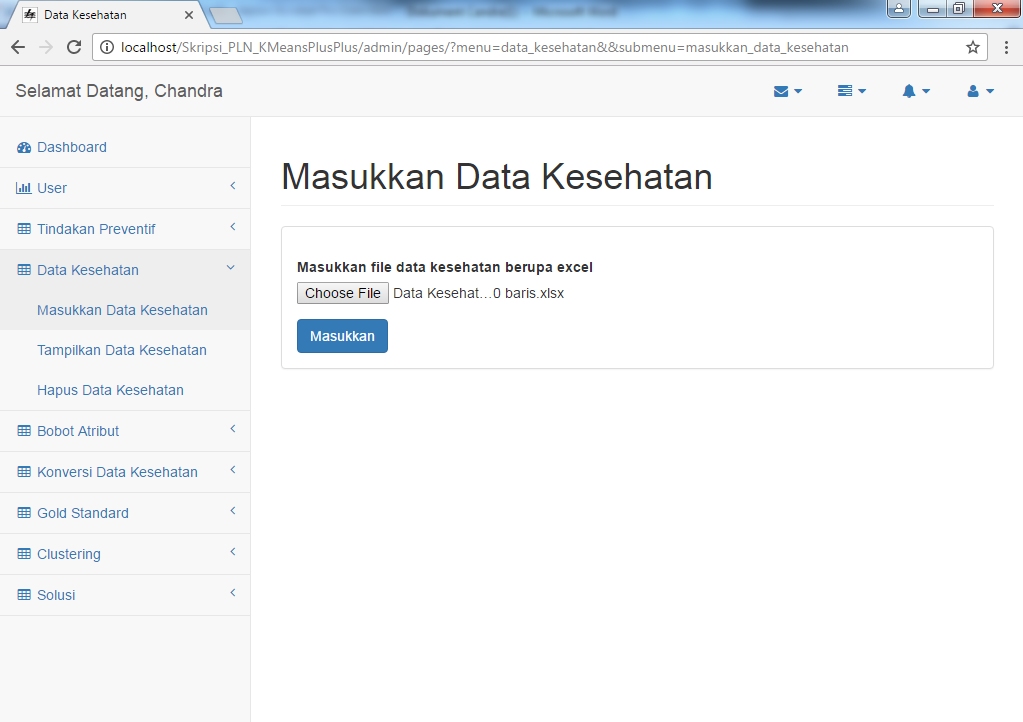
1. Halaman Hapus Menu



Gambar 4. 41 Halaman hapus menu

Jika menu yang bersangkutan tidak diperbolehkan mengakses aplikasi PPKP maka harus dihapus menggunakan menu hapus menu.

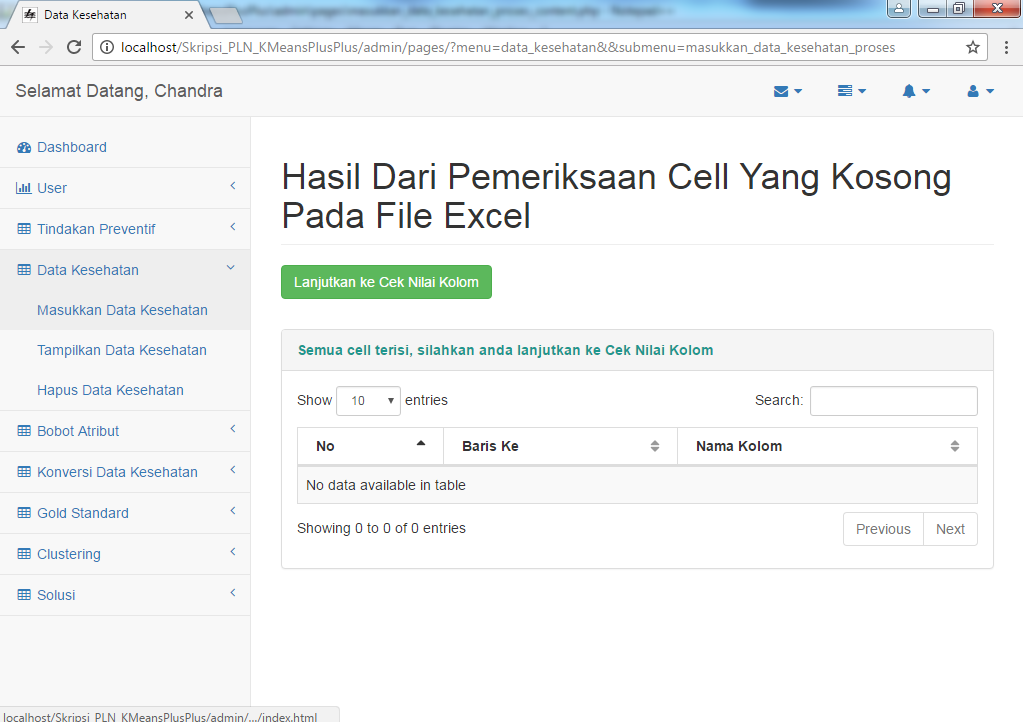
1. Halaman Masukkan Data Kesehatan



Gambar 4. 42 Halaman masukkan data kesehatan

Untuk memulai *clustering* dibutuhkan data masukkan yaitu data kesehatan pegawai melalui menu masukkan data kesehatan.

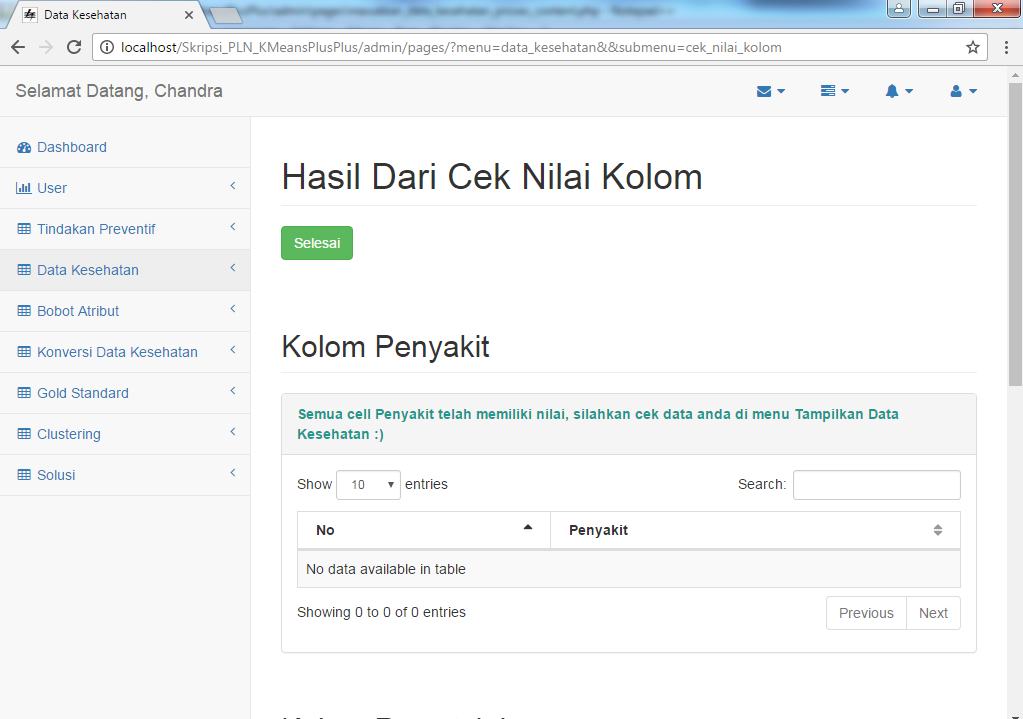
1. Halaman Hasil Pemeriksaan Cell Yang Kosong



Gambar 4. 43 Halaman hasil pemeriksaan *cell* yang kosong

File yang telah di*upload* akan di periksa *cell* yang kosong, jika tidak ada maka akan lanjut ke tahap cek nilai pada *database*, jika iya aplikasi akan menolak *file* yang telah di*upload*.

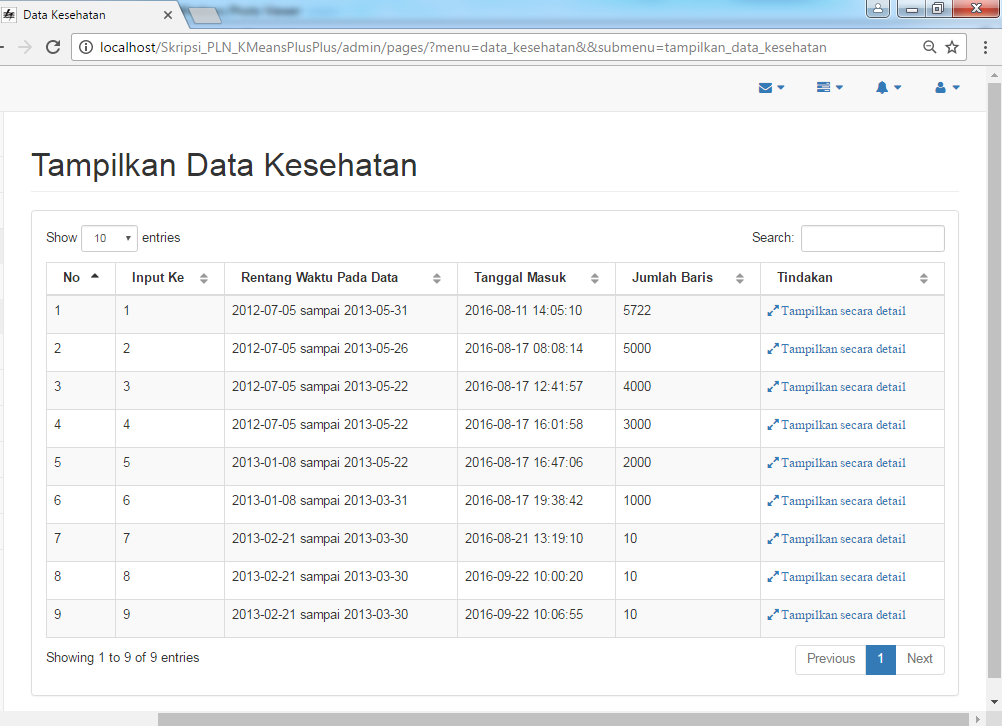
1. Halaman Cek Nilai Kolom Pada Database



Gambar 4. 44 Halaman cek nilai kolom pada *database*

Jika tidak ada *cell* yang kosong maka akan dilanjutkan di tahap cek nilai kolom penyakit. Jika penyakit tidak ada di *database* atau telah ada tetapi belum diberi nilai maka akan muncul nama penyakit yang bermasalah dan harus diberi nilai jika data yang dimasukkan ingin di hitung. Jika penyakit ada dan telah diberi nilai maka hasil cek nilai kolom akan kosong.

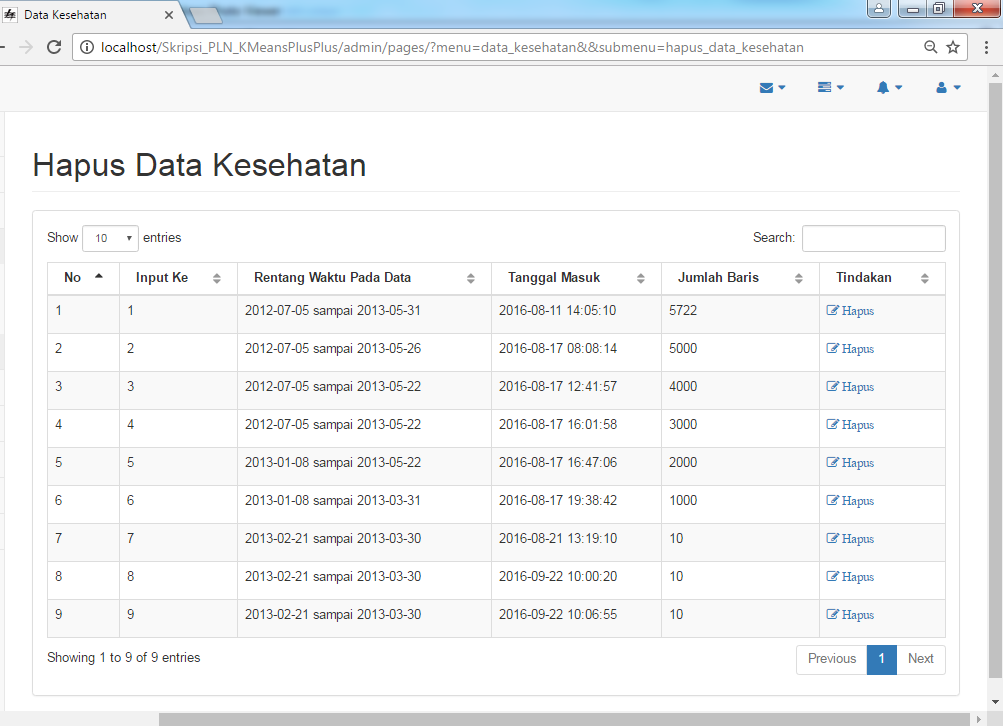
1. Halaman Tampilkan Data Kesehatan



Gambar 4. 45 Halaman tampilkan data kesehatan

Setelah melewati proses cek nilai kolom penyakit maka data yang telah dimasukkan dapat dilihat pada menu tampilkan data kesehatan.

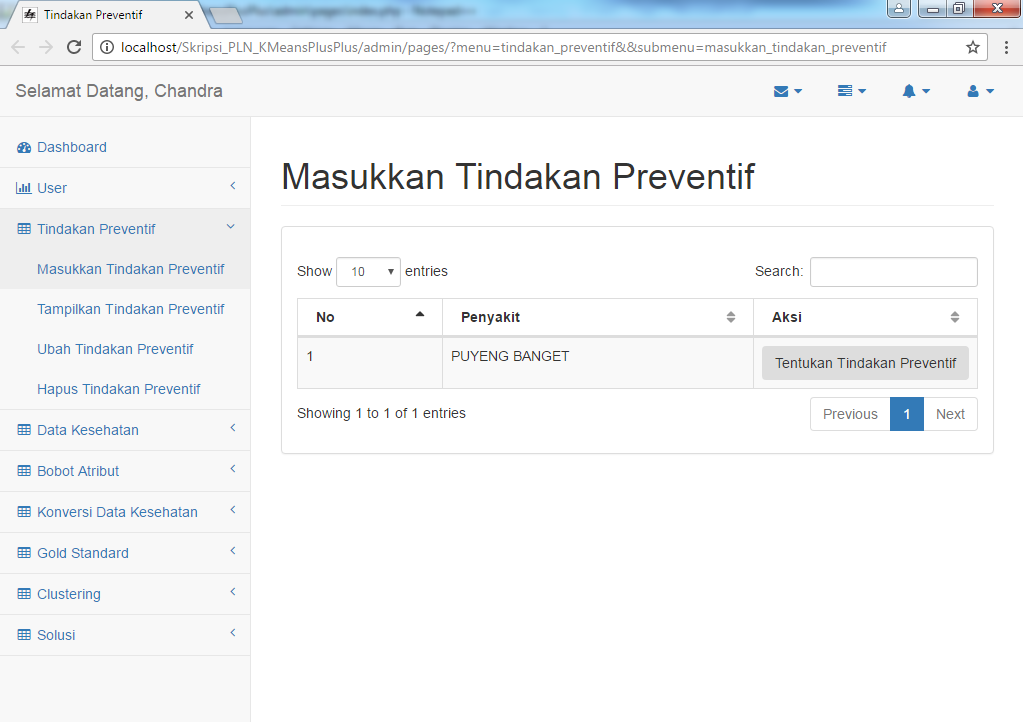
1. Halaman Hapus Data Kesehatan



Gambar 4. 46 Halaman hapus data kesehatan

Jika data yang telah dimasukkan terdapat kesalahan maka harus dihapus secara keseluruhan.

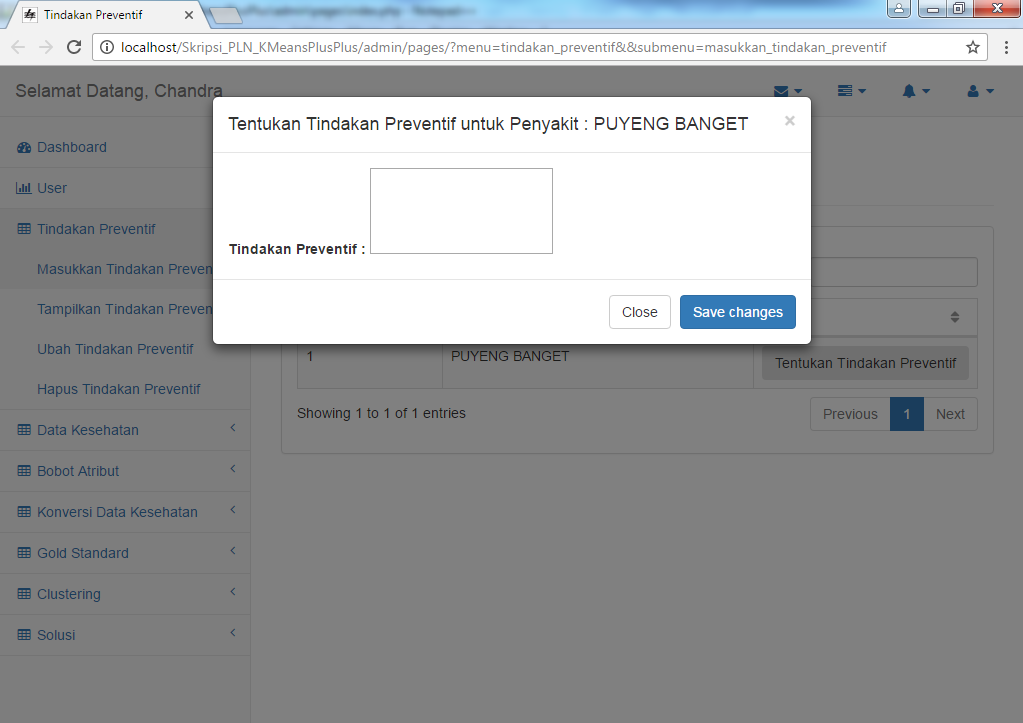
1. Halaman Masukkan Program Kesehatan



Gambar 4. 47 Halaman tampilkan data kesehatan

Jika ketika memasukkan data kesehatan terdapat penyakit yang tidak ada dalam *database* maka akan muncul penyakit pada menu masukkan program kesehatan. Jika tidak diberikan nilai maka data masukkan yang memiliki penyakit bersangkutan tidak akan bisa dihitung.

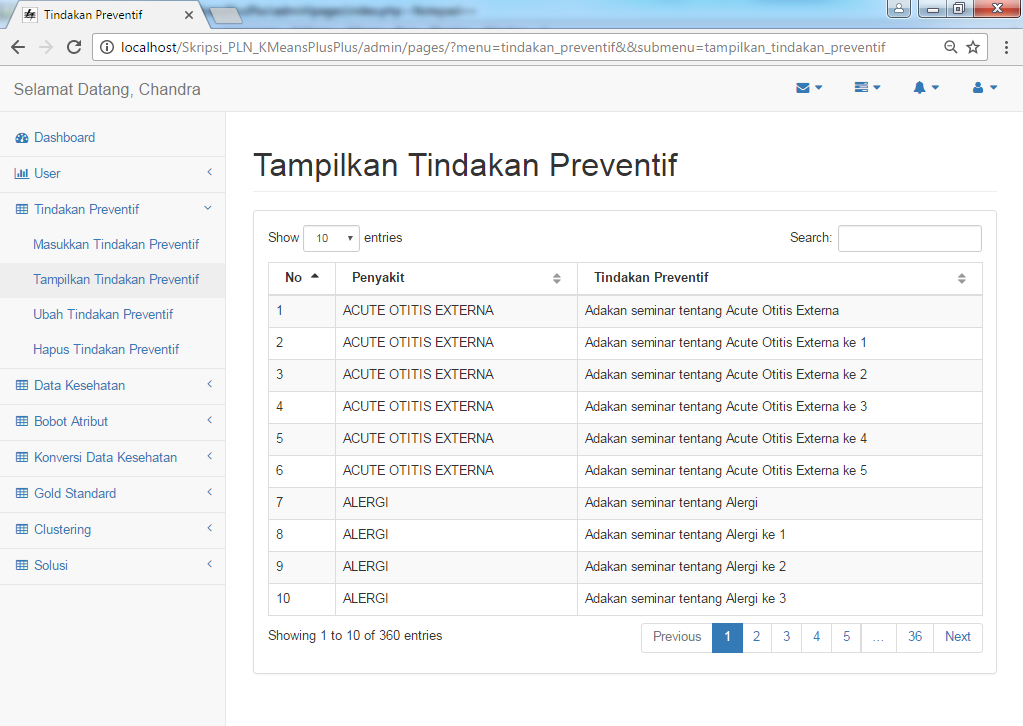
1. Form Masukkan Program Kesehatan



Gambar 4. 48 Form masukan program kesehatan

Form untuk memasukkan program kesehatan terhadap penyakit yang bersangkutan.

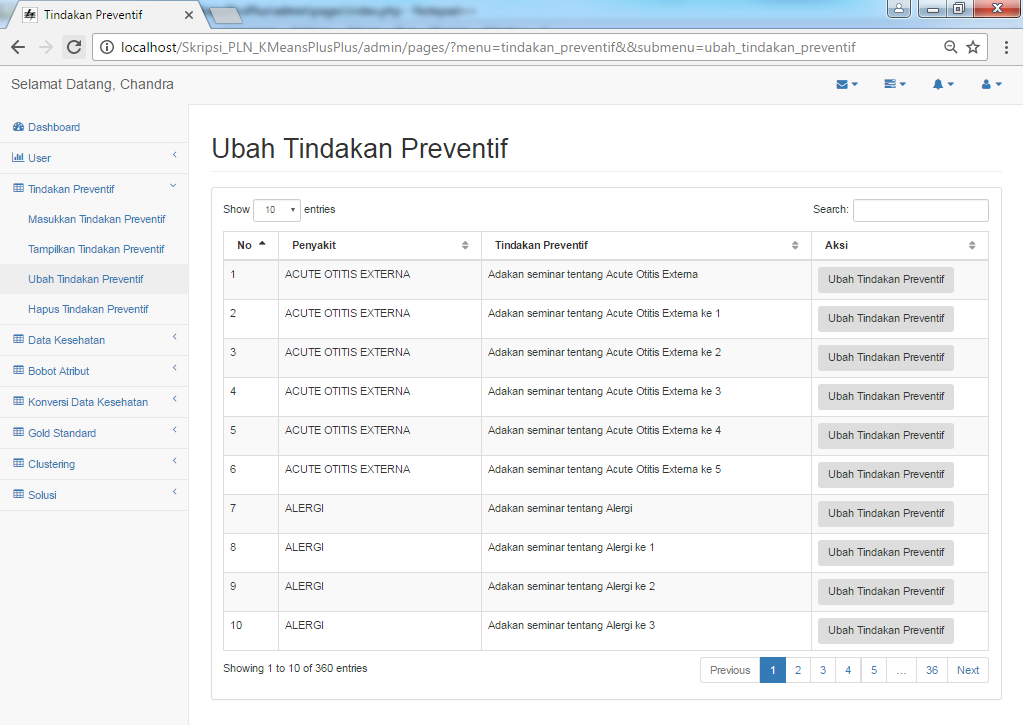
1. Halaman Tampilkan Program Kesehatan



Gambar 4. 49 Halaman tampilkan program kesehatan

Jika ingin melihat semua program kesehatan yang tidak bermasalah, menu dapat mengakses menu tampilkan program kesehatan.

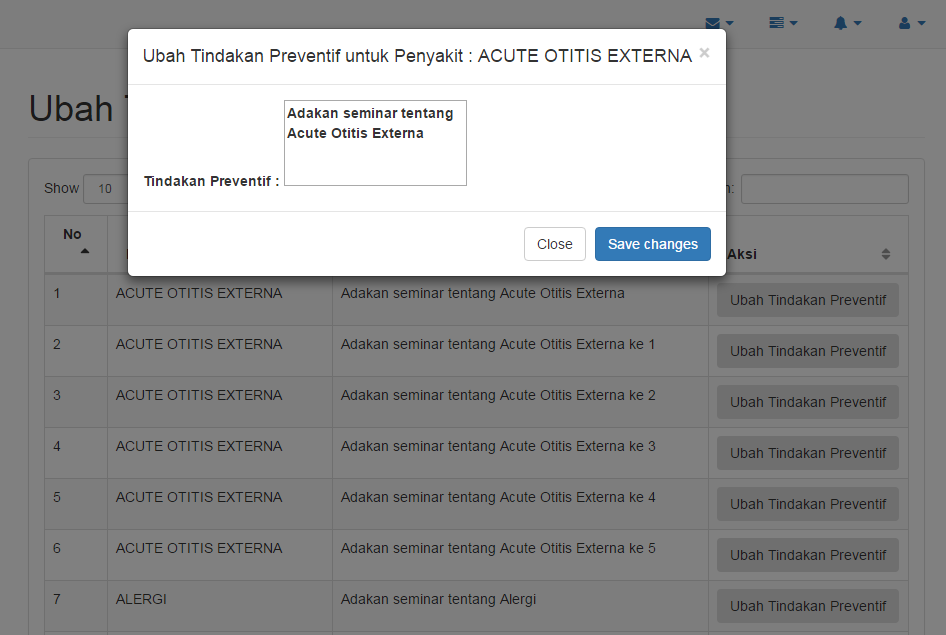
1. Halaman Ubah Program Kesehatan



Gambar 4. 50 Halaman ubah program kesehatan

Jika ingin merubah program kesehatan pada *database*, menu dapat mengakses menu ubah program kesehatan.

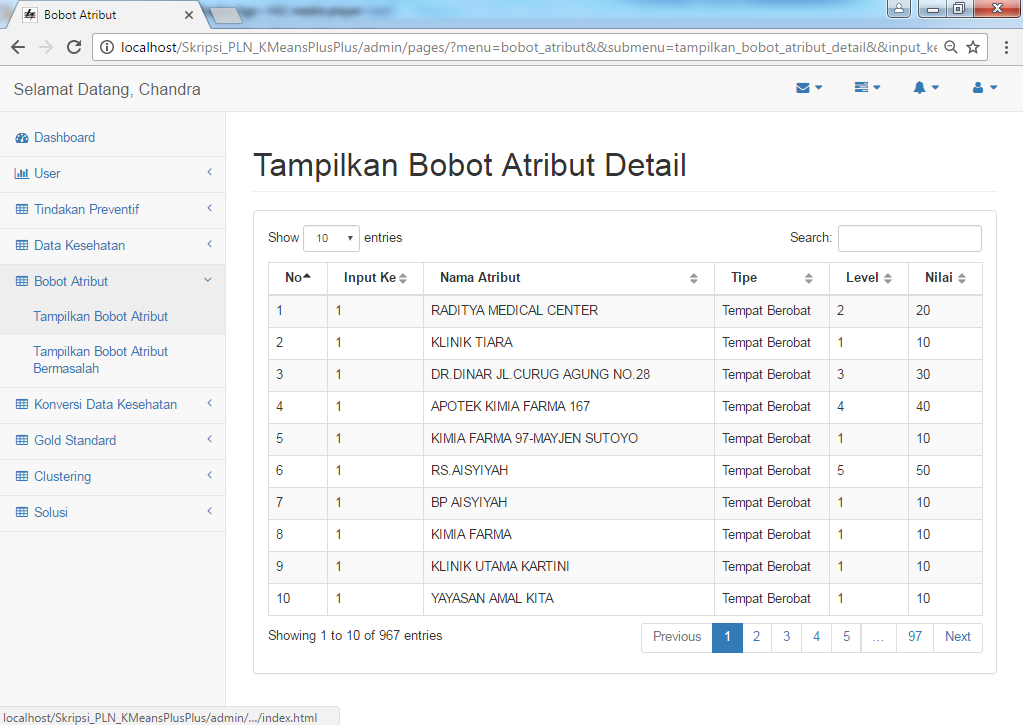
1. Halaman Ubah Program Kesehatan Form



Gambar 4. 51 Form ubah program kesehatan

Untuk mengganti program kesehatan menu dapat klik tombol “Ubah Program Kesehatan” pada bagian kanan Aplikasi PPKP.

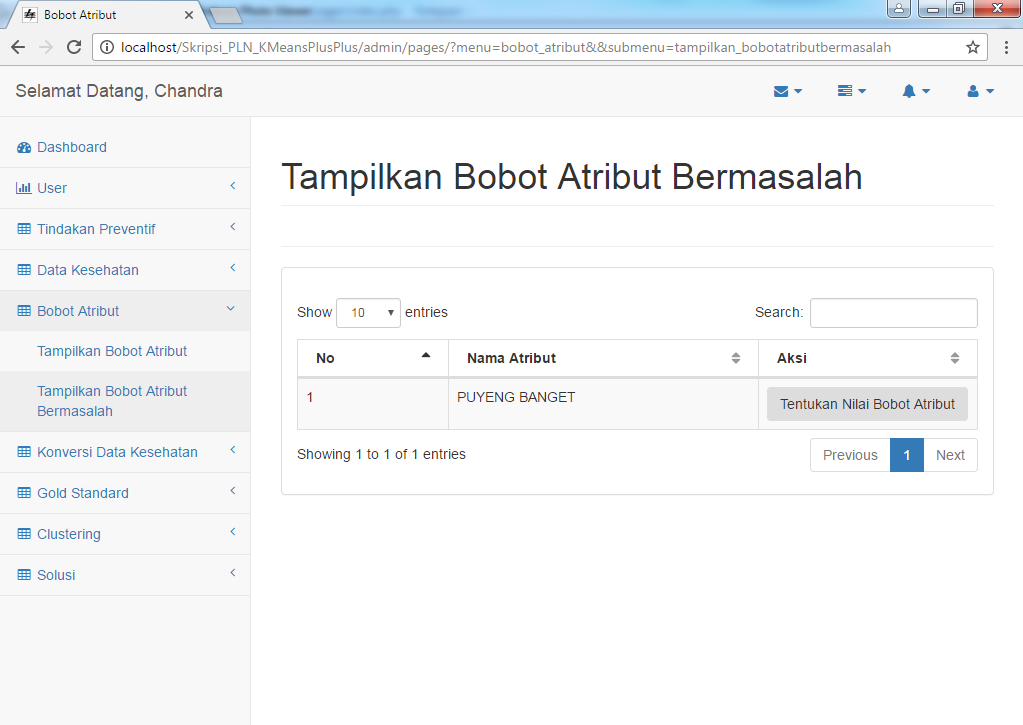
1. Halaman Tampikan Bobot Atribut



Gambar 4. 52 Halaman tampikan bobot atribut detail

Setelah *file* telah di*upload* maka pada kolom rawat jalan, rawat inap, penyakit dan tempat berobat akan dirubah menjadi angka dan bisa dilihat pada Menu Bobot Atribut.

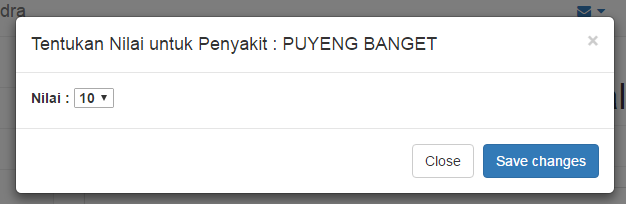
1. Halaman Tampilkan Bobot Atribut Bermasalah



Gambar 4. 53 Halaman tampikan bobot atribut detail

Jika tidak terdapat nilai penyakit pada *database* maka Menu Bobot Atribut Bermasalah menampilkan sebanyak jumlah penyakit atau tempat berobat yang bersalah.

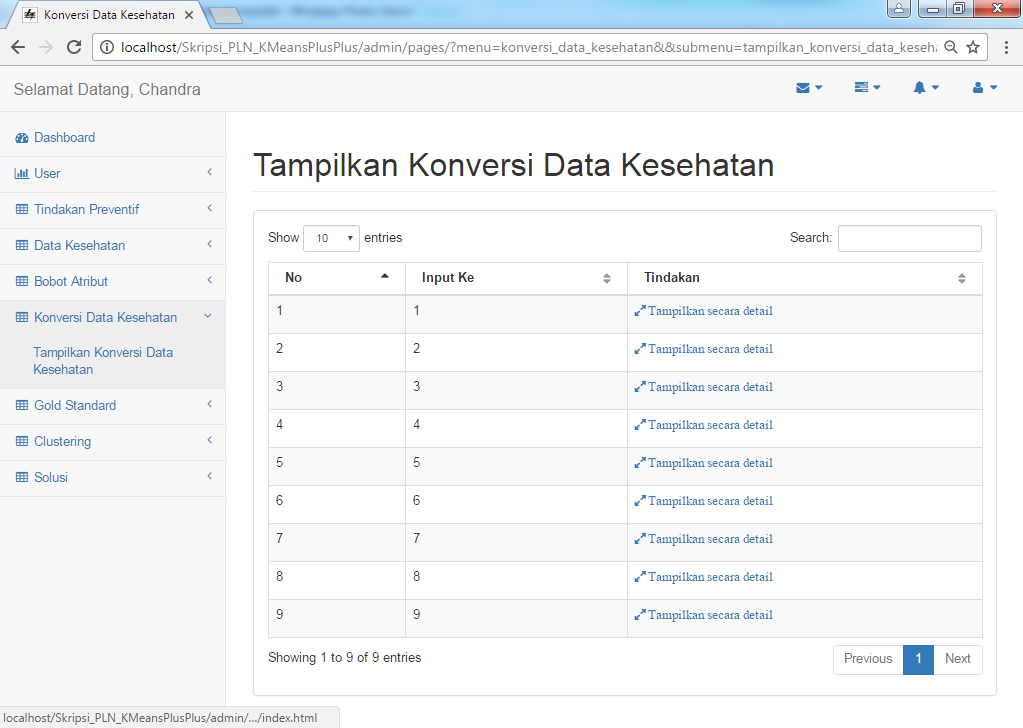
1. Form Tampilkan Bobot Atribut Bermasalah



Gambar 4. 54 Form menentukan nilai penyakit

Untuk menentukan nilai penyakit yang bersangkutan maka klik tombol “Tentukan Nilai Bobot Atribut” agar menampilkan form nilai.

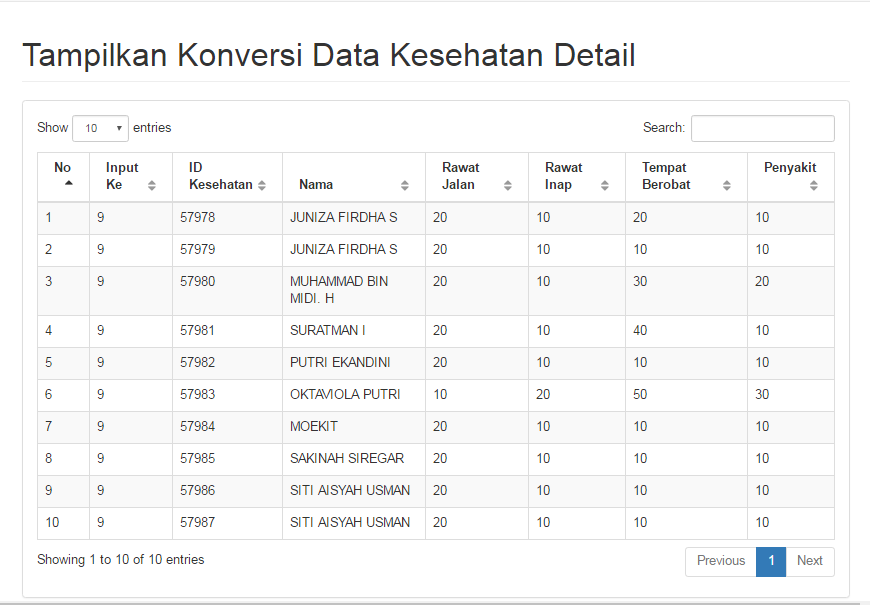
1. Halaman Tampilkan Konversi Data Kesehatan



Gambar 4. 55 Halaman tampilkan konversi data kesehatan

Setelah data dimasukkan dan diubah menjadi angka, jika menu ingin melihat hasil konversi seluruh data dapat dilihat pada menu tampilkan konversi data kesehatan. Untuk melihat lebih detail klik *link* “Tampilkan secara detail”.

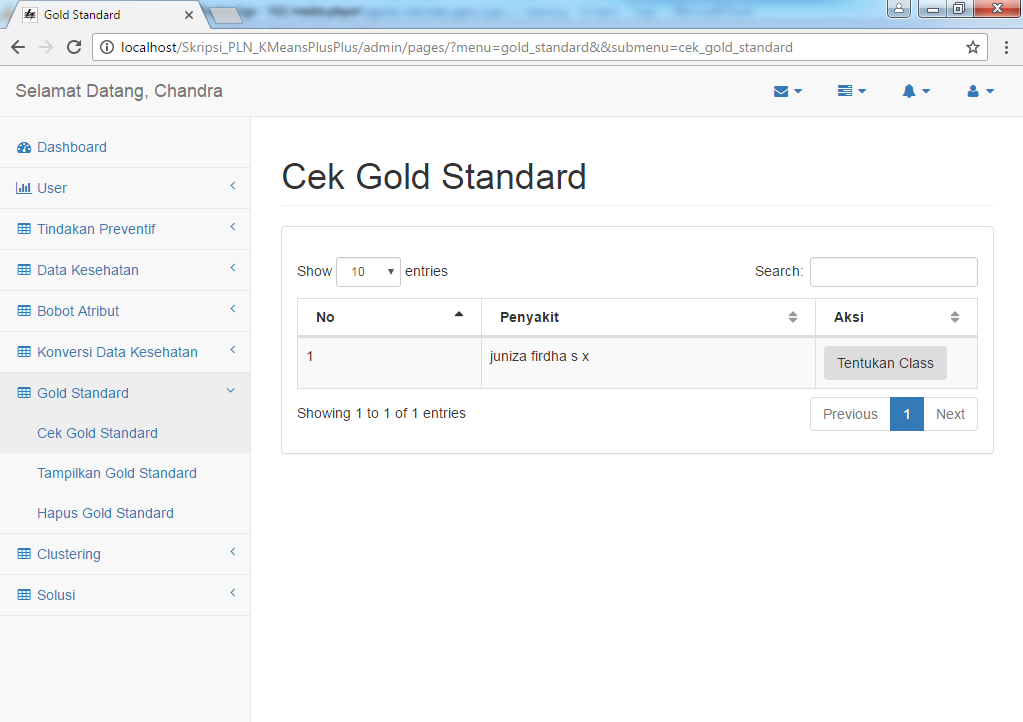
1. Halaman Tampilkan Konversi Data Kesehatan Detail



Gambar 4. 56 Halaman tampilkan detail konversi data

Jika menu klik *link* “Tampilkan secara detail” maka akan muncul secara detail konversi data yang telah dimasukkan menjadi angka.

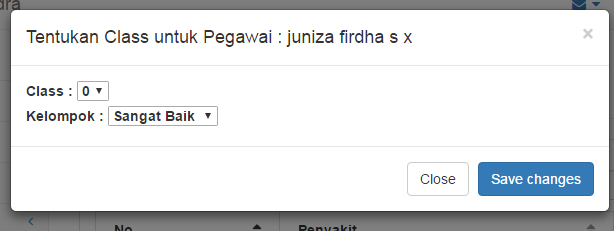
1. Halaman Cek *Gold Standard*



Gambar 4. 57 Cek *Gold Standard*

Untuk menentukan *class* dan kelompok pada *gold standard* klik menu cek *gold standard*. Jika terdapat nama yang belum ditentukan maka akan muncul pada tabel.

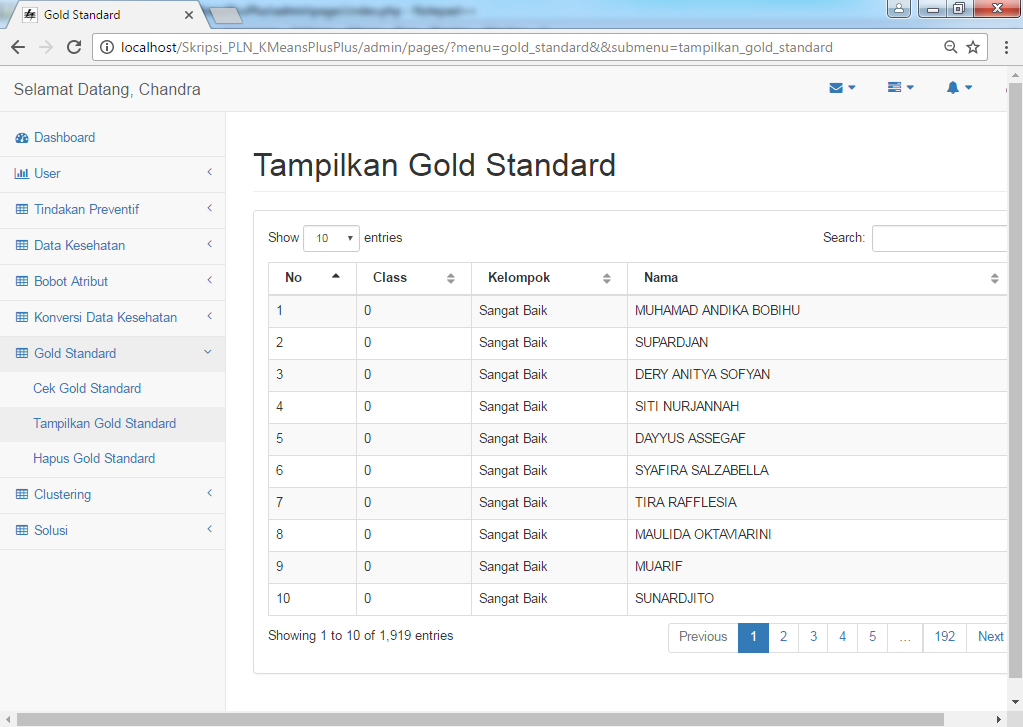
1. Form Menentukan *Class*



Gambar 4. 58 menentukan *class*

Untuk menentukan *class* dan kelompok pegawai yang bersangkutan klik tombol “Tentukan *Class*” lalu akan muncul form untuk memasukkan *class* dan kelompok.

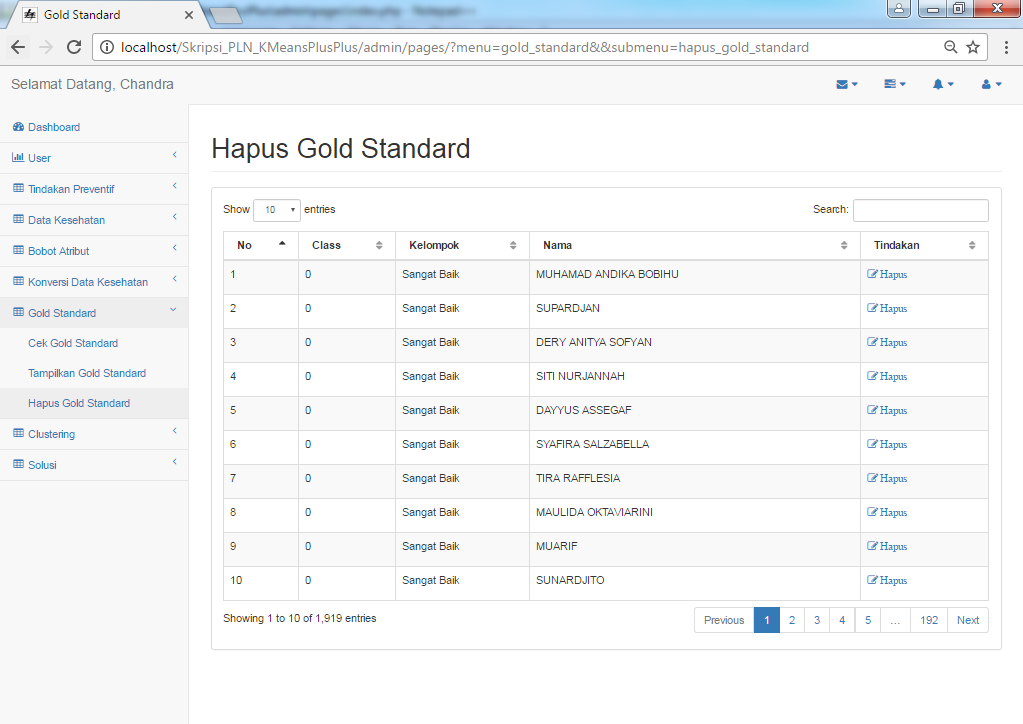
1. Halaman Tampilkan *Gold Standard*



Gambar 4. 59 menampilkan *gold standard*

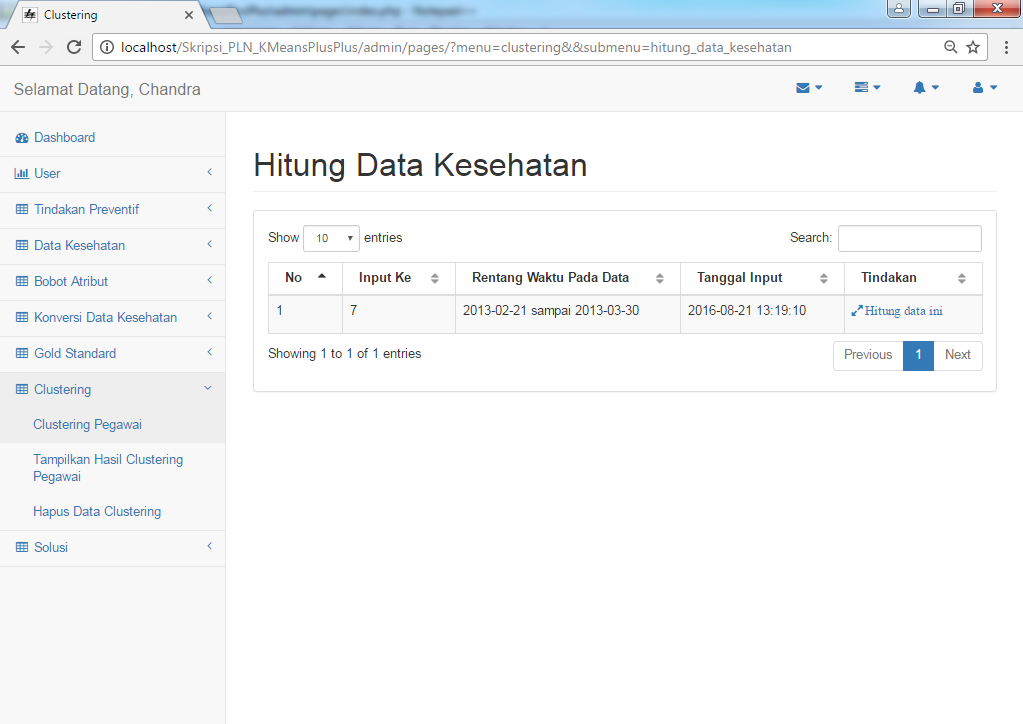
Untuk menampilkan *gold standard* pilih Menu Tampilkan *Gold Standard*. Semua data *gold standard* yang tidak bermasalah akan muncul pada menu ini.

1. Halaman Hapus *Gold Standard*



Gambar 4. 60 menghapus gold standar

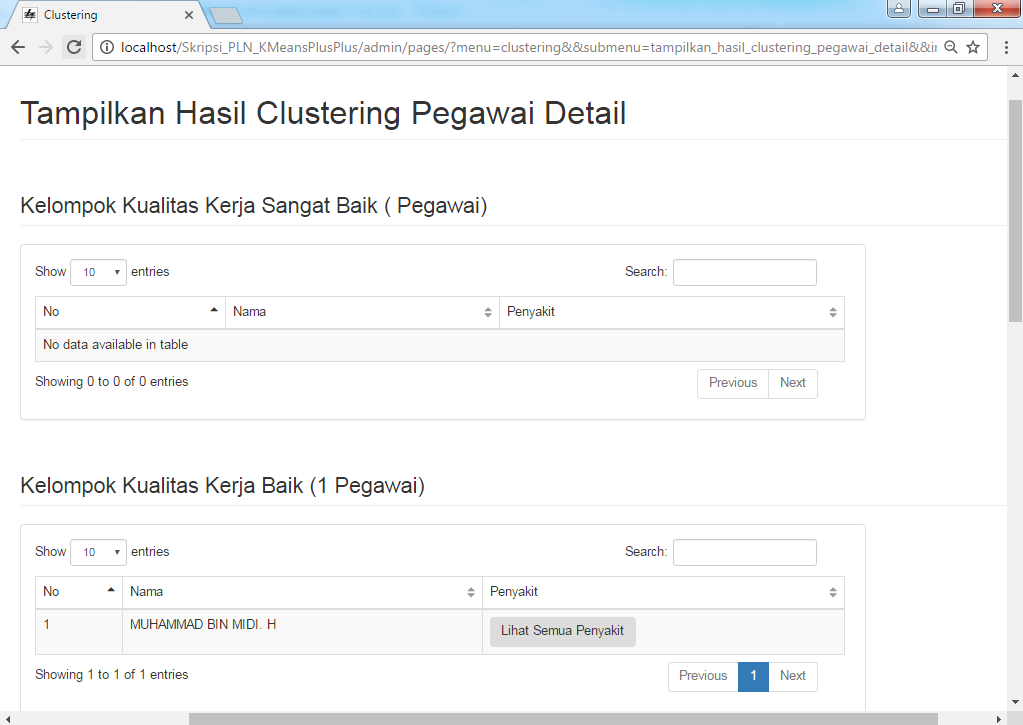
1. Halaman Hitung Data Kesehatan *Clustering*



Gambar 4. 61 menghitung gold standar

Jika data yang telah dimasukkan tidak mengalami masalah maka ketika menu klik Menu *Clustering* Pegawai akan muncul data yang telah dimasukkan menu untuk dihitung hingga menghasilkan program kesehatan. Untuk mulai menghitung klik *link* hitung data ini, maka akan diproses oleh Aplikasi PPKP.

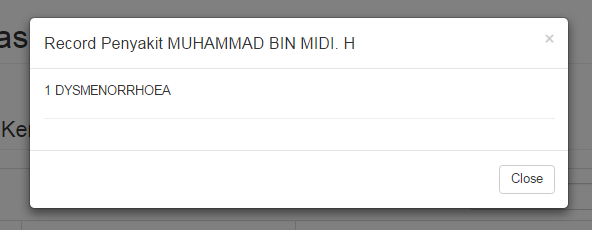
1. Halaman Tampilkan Hasil *Clustering* Pegawai



Gambar 4. 62 Hasil *clustering* pegawai

Untuk melihat hasil dari perhitungan pada Menu *Clustering* Pegawai, menu dapat klik Menu Tampilkan Hasil *Clustering* Pegawai. Pada menu ini akan menampilkan kelompok kualitas kerja yang telah di *clustering* dengan nama pegawai serta penyakit yang diderita.

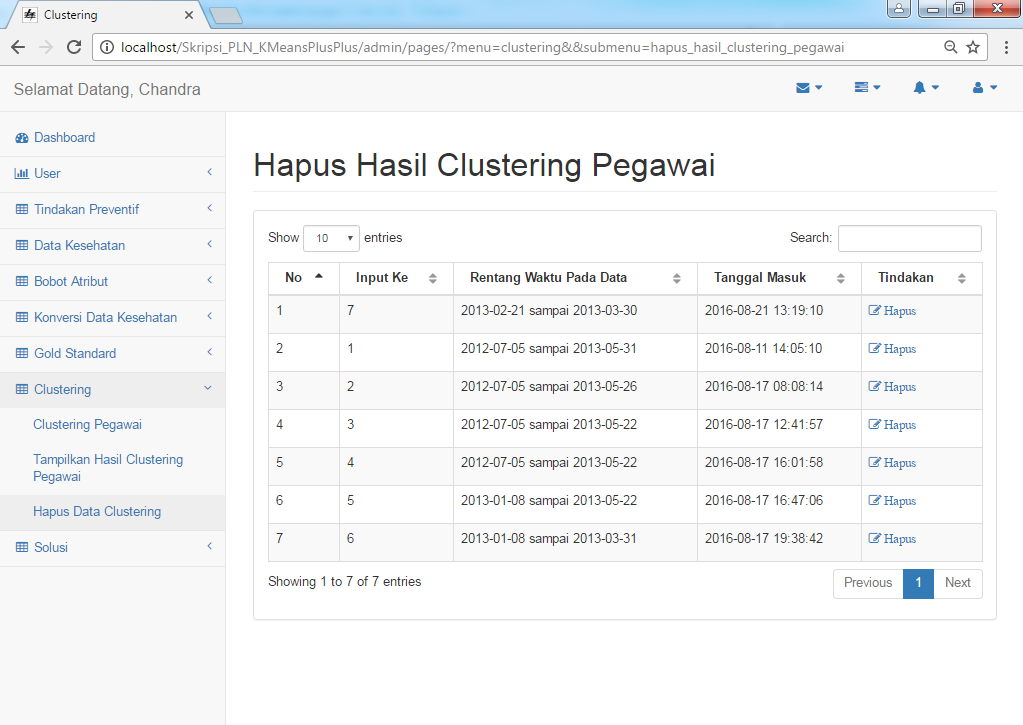
1. *Record* penyakit pegawai



Gambar 4. 63 hasil *clustering*

Jika menu ingin melihat *record* penyakit pegawai yang bersangkutan, menu dapat klik tombol “Lihat Semua Penyakit”.

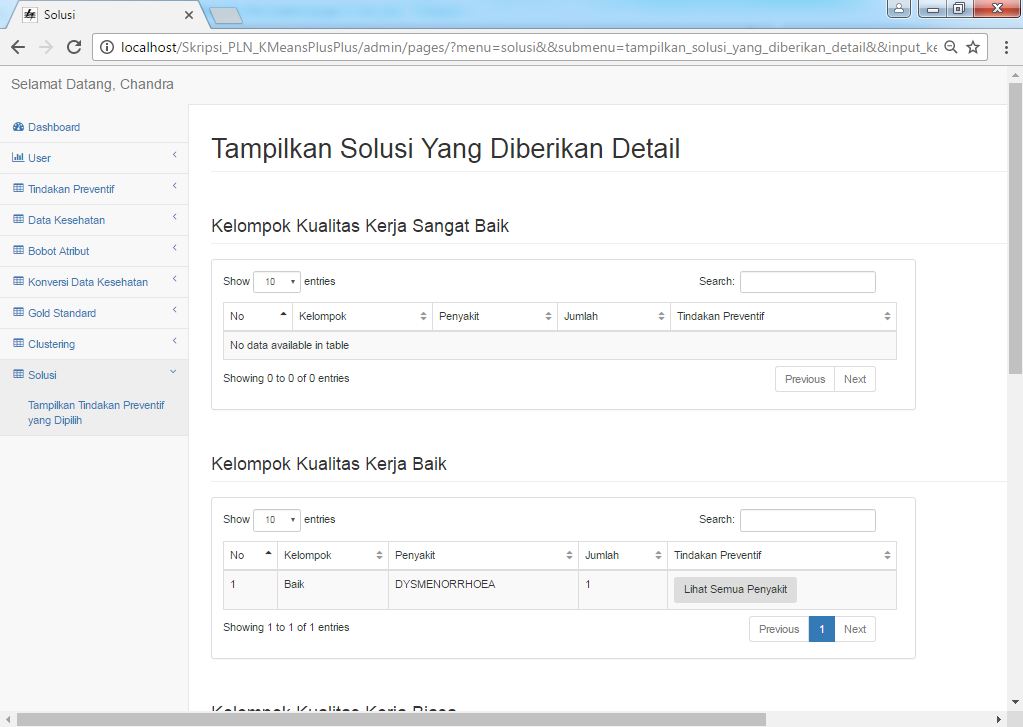
1. Halaman Hapus Hasil *Clustering* Pegawai



Gambar 4. 64 hapus *clustering* pegawai

Jika terdapat kesalahan pada data yang dimasukkan pada saat awal proses, maka menu harus menghapus seluruh data yang berkaitan dengan cara klik *link* “Hapus” pada sisi kanan Aplikasi PPKP.

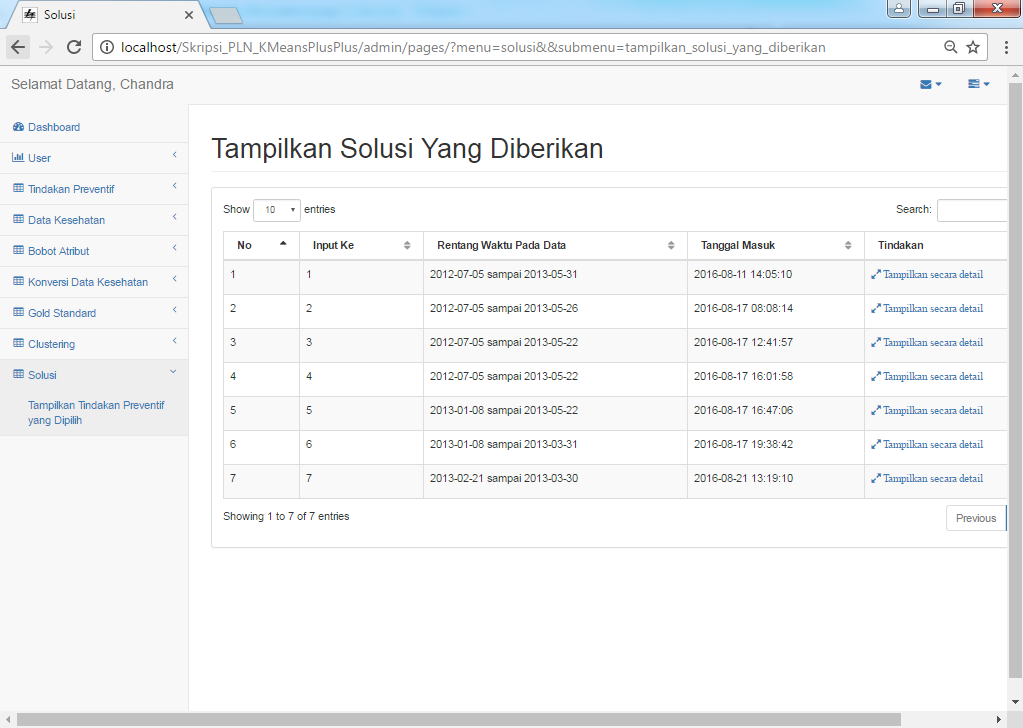
1. Halaman Tampilkan Solusi Yang Diberikan Secara Detail



Gambar 4. 65 menampikan solusi detail

Hasil akhir dari perhitungan K-Means adalah solusi berupa program kesehatan yang tepat untuk data yang telah dimasukkan oleh menu. Untuk melihat solusi yang telah dimasukkan pada *database*, menu dapat klik Menu Tampilkan Program Kesehatan.

1. Halaman Tampilkan Solusi Yang Diberikan



Gambar 4. 66 solusi yang diberikan

Hasil akhir dari perhitungan K-Means adalah solusi berupa program kesehatan yang tepat untuk data yang telah dimasukkan oleh menu. Untuk melihat solusi yang telah dimasukkan pada *database*, menu dapat klik Menu Tampilkan Program Kesehatan.

1. Fase Implementasi

Dalam tahap implementasi dibagi menjadi 2 tahapan yaitu membangun sistem dan menguji sistem. Untuk lebih jelasnya akan dipaparkan dibawah ini.

* + - 1. Membangun Sistem

Pada tahap ini sistem dibangun dengan beberapa software pendukung diantaranya XAMPP v3.2.1 yang mencakup Apache 2.4.9 sebagai web server, PHP 5.5.11 sebagai bahasa pemrograman, dan MySQL versi 5.6.16 sebagai *database*. Untuk software editor peneliti menggunakan Notepad++, untuk mengolah gambar peneliti menggunakan Adobe Photoshop CS5. Kemudian sebagai server basis data menggunakan server: Localhost, server type:MySQL, versi server 5.5.48-cll – MySQL Community Server (GPL). Sedangkan untuk sistem operasi peneliti menggunakan Windows7.

1. Pengujian Sistem

Pada tahap ini sistem yang dibangun akan diuji dengan menggunakan metode pengujian black-box testing. Pengujian dilakukan pada sistem secara keseluruhan menggunakan *black box* untuk memastikan sistem berjalan sesuai dengan yang direncanakan. Pengujian ini dilakukan pada tanggal 21 Desember 2015 di PT PLN P2B JB oleh Kepala Divisi SDM.

Dibawah ini merupakan tabel hasil pengujian yang dilakukan terhadap aplikasi peningkatan kesehatan pegawai PLN P2B JB :

Tabel 4.55 Skenario Pengujian Black Box

| No | *Use case* | Hasil Yang Diharapkan |
| --- | --- | --- |
| 1 | *Login* | Menampilkan formulir *username* dan *password* |
| 2 | Memasukkan Menu | Menyimpan data menu yang baru |
| 3 | Menampilkan Menu | Menampilkan data menu secara kesuluruhan dari *database* |
| 4 | Mengubah Menu | Mengubah data menu berdasarkan id\_menu yang telah dipilih |
| 5 | Menghapus Menu | Menghapus data menu berdasarkan id\_menu yang telah dipilih |
| 6 | Masukkan Data Kesehatan | Menyimpan data kesehatan pegawai yang telah dipersiapkan kedalam *database* |
| 7 | Menampilkan Bobot Atribut | Menampilkan seluruh bobot atribut dalam *database* |
| 8 | Menampilkan Data Kesehatan | Menampilkan data kesehatan pegawai dari *database* berdasarkan input\_ke |
| 9 | Menghapus Data Kesehatan | Menghapus data kesehatan pada *database* berdasarkan input\_ke |
| 10 | Menampilkan Konversi Data Kesehatan | Menampilkan konversi data kesehatan pegawai dari *database* berdasarkan input\_ke |
| 11 | *Clustering*  Pegawai | Menghitung data kesehatan pegawai berdasarkan input\_ke pada *database* menggunakan metode K-Means++ dan COP-KMeans dan menghasilkan *cluster* kualitas kerja |
| 12 | Menampilkan Hasil *Clustering*  Pegawai | Menampilkan hasil *clustering*  pegawai dari *database* berdasarkan input\_ke |
| 13 | Menghapus Hasil *Clustering*  Pegawai | Menghapus hasil *clustering*  pegawai berdasarkan input\_ke yang telah dipilih |

Tabel 4.56 Rincian Pengujian Black Box

| No | *Use case* | Data Masukkan | Yang Diharapkan | Pengamatan | Kesimpulan |
| --- | --- | --- | --- | --- | --- |
| 1 | *Login* | Menu memasukkan *username* dan *password* | Menampilkan formulir *username* dan *password* | Sistem menampilkan form *username* dan *password* | Sukses |
| 2 | Memasukkan Menu | Menu memilih menu masukkan menu | Menyimpan data menu yang baru | Sistem menampilkan halaman dengan form masukkan menu | Sukses |
| 3 | Menampilkan Menu | Menu memilih menu menampilkan menu | Menampilkan data menu secara kesuluruhan dari *database* | Sistem menampilkan halaman menu yang telah ada pada *database* | Sukses |
| 4 | Mengubah Menu | Menu memilih menu mengubah menu | Mengubah data menu berdasarkan id\_menu yang telah dipilih | Sistem menampilkan halaman dengan form mengubah menu | Sukses |
| 5 | Menghapus Menu | Menu memilih menu menghapus menu | Menghapus data menu berdasarkan id\_menu yang telah dipilih | Sistem menampilkan halaman menghapus menu | Sukses |
| 6 | Masukkan Data Kesehatan | Menu memilih menu masukkan data kesehatan | Menyimpan data kesehatan pegawai yang telah dipersiapkan kedalam *database* | Sistem menampilkan halaman dengan form masukkan data kesehatan | Sukses |
| 7 | Menampilkan Bobot Atribut | Menu memilih menu menampilkan bobot atribut | Menampilkan seluruh bobot atribut dalam *database* | Sistem menampilkan halaman bobot atribut yang telah ada pada *database* | Sukses |
| 8 | Menampilkan Data Kesehatan | Menu memilih menu menampilkan data kesehatan | Menampilkan data kesehatan pegawai dari *database* berdasarkan input\_ke | Sistem menampilkan halaman data kesehatan yang telah ada pada *database* | Sukses |
| 9 | Menghapus Data Kesehatan | Menu memilih menu menghapus data kesehatan | Menghapus data kesehatan pada *database* berdasarkan input\_ke | Sistem menampilkan halaman menghapus data kesehatan | Sukses |
| 10 | Menampilkan Konversi Data Kesehatan | Menu memilih menu menampilkan konversi data kesehatan | Menampilkan konversi data kesehatan pegawai dari *database* berdasarkan input\_ke | Sistem menampilkan halaman konversi data kesehatan yang telah ada pada *database* | Sukses |
| 11 | *Clustering*  Pegawai | Menu memilih menu *clustering*  pegawai | Menghitung data kesehatan pegawai berdasarkan input\_ke pada *database* menggunakan metode K-Means++ dan COP-KMeans dan menghasilkan *cluster* kualitas kerja | Sistem menampilkan halaman *clustering*  pegawai | Sukses |
| 12 | Menampilkan Hasil *Clustering*  Pegawai | Menu memilih menu menampilkan hasil *clustering*  pegawai | Menampilkan hasil *clustering*  pegawai dari *database* berdasarkan input\_ke | Sistem menampilkan halaman konversi data kesehatan yang telah ada pada *database* | Sukses |
| 13 | Menghapus Hasil *Clustering*  Pegawai | Menu memilih menu menghapus hasil *clustering*  pegawai | Menghapus hasil clusatering pegawai berdasarkan input\_ke yang telah dipilih | Sistem menampilkan halaman menghapus hasil *clustering*  pegawai | Sukses |

Selain itu, juga dilakukan pengujian sistem untuk *clustering*  kesehatan pegawai dilakukan menggunakan satu buah perangkat komputer jinjing atau laptop dengan spesifikasi Memori RAM 4 GB, Processor Intel Core i3, sistem operasi Windows 7 64-bit, dan Browser Google Chrome. Dengan total sebanyak 5.706 *record* pada *database* kesehatan pegawai. Hasil pengujian yang dilakukan dapat dilihat pada tabel berikut.

Tabel 4.57 Hasil pengujian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pengujian ke** | **Jumlah *Record*** | **Jumlah Cluster** | **Waktu Eksekusi** | **Akurasi** | |
| **Nilai** | **Kesimpulan** |
| 1 | 5.706 | 5 | 134 menit 14 detik | 0,87493 | Tepat |
| 2 | 5.706 | 5 | 113 menit 55 detik | 0,54906 | Tepat |
| 3 | 5.706 | 5 | 60 menit 49 detik | 0,03549 | Tidak Tepat |
| Rata-Rata Akurasi | | | | 0,48649 | Tidak Tepat |

Berdasarkan hasil dari tiga kali pengujian tampak bahwa algoritma K-Means++ dan COP-Kmeans mampu mengelompokkan kualitas kerja dengan benar sebanyak dua kali yaitu pada pengujian pertama dan kedua sedangkan pengujian ketiga mengelompokkan kualitas kerja dengan tidak tepat. Perbedaan kesimpuilan antara pengujian satu sampai tiga dikarenakan centroid awal ditentukan secara acak sehingga jika dilakukan pengujian yang keempat kali memiliki kesimpulan yang berbeda. Karena Kepala Divisi SDM hanya mengambil kesimpulan pertama menimbang waktu yang dibutuhkan sangat lama, maka pada penelitian ini hanya mengambil akurasi dan kesimpulan pada pengujian yang pertama.