

BAB IV

HASIL DAN PEMBAHASAN

4.1 Kebutuhan Perangkat Keras (*Hardware*)

Spesifikasi minimal perangkat keras yang digunakan untuk mengimplementasikan aplikasi penyampaian informasi ini sebagai berikut :

- a. *Processor* Intel P4 1.6 GH
- b. VGA 64 MB
- c. RAM 512 MB
- d. *Harddisk* 80 GB
- e. LCD Monitor 15"
- f. *CD room* standar
- g. *Keyboard* dan *mouse* standar
- h. *Printer* standar

4.2 Kebutuhan Perangkat Lunak (*Software*)

Spesifikasi minimal perangkat lunak yang digunakan untuk membuat dan mengimplementasikan aplikasi penyampaian informasi ini sebagai berikut :

1. Sistem operasi komputer yang bisa digunakan yaitu *Microsof Windows XP*.
2. Bahasa pemrograman yang digunakan untuk pengoperasian sistem pakar mendiagnosa tingkat stress adalah *Borland Delphi 7*.
3. *Ms. Access 2007* sebagai media penyimpanan *database*.
4. *Crystal Report 8.5* digunakan untuk mencetak laporan dari hasil penelusuran sistem pendiagnosa tingkat stress.

4.3 Implementasi Sistem

Tahap ini menjelaskan bagaimana program ini bekerja untuk membantu *user* dalam proses pengelolaan data-data menggunakan aplikasi yang dibuat. Untuk lebih jelasnya, berikut ini adalah penjelasan program aplikasi pendeteksi tingkat stress menggunakan metode perhitungan CBR :

4.3.1 Form Login

Form login berfungsi untuk melakukan pengecekan status pengguna, apakah memiliki otoritas terhadap aplikasi atau tidak.

Pertama, pengguna memilih *User Name*, selanjutnya pengguna meng-*input*-kan *password*. Fungsi tombol **Login** adalah untuk masuk ke menu utama. Jika *user name* dan *password* terdaftar pada database, maka akan menampilkan *form* menu utama. Tombol **Keluar** berfungsi untuk menutup program. Untuk lebih jelasnya dapat dilihat pada gambar 4.1 berikut ini :



LOGIN	
USER NAME	Made
PASSWORD	***
HAK AKSES	ADMIN
<input type="button" value="✓ LOGIN"/>	<input type="button" value="✗ KELUAR"/>

Gambar 4.1 Tampilan Form Login

4.3.2 Form Menu Utama

Form menu utama merupakan menu utama dari aplikasi sistem. *Form* ini berfungsi sebagai menu utama dalam mengakses form-form lain pada aplikasi. *Form* ini terdiri dari beberapa *menu bar*, yaitu **File**, **Data**, **Laporan** dan **Info**. Menu **File**, terdiri dari Sub menu :

Manajemen User : Berfungsi untuk menampilkan *Form* Manajemen *User*.

Form ini digunakan untuk mengatur siapa saja yang mempunyai hak akses terhadap aplikasi.

Ganti Password : Berfungsi untuk menampilkan *Form* Ganti *Password*.

Input Indikator : Berfungsi untuk menampilkan *Form* *Input* Indikator.

Input Kasus : Berfungsi untuk menampilkan *Form* *Input* Kasus.

Exit : Berfungsi untuk keluar dari program.

Menu **Data**, terdiri dari sub menu:

Input Penyakit : Berfungsi untuk menampilkan *Form* *Input* Penyakit.

Input Kasus Baru : Berfungsi untuk menampilkan *Form* *Input* Kasus Baru.

Ketika *user* melakukan klik pada sub menu tertentu misalkan *sub menu input* Indikator, maka akan ditampilkan *form input* Indikator. Pada kondisi ini, nilai *enable* pada *form* menu utama adalah *false* sampai dengan *user* menutup *form child*. Sehingga memungkinkan *user* untuk fokus pada form yang dipilih. Untuk lebih jelasnya dapat dilihat pada gambar 4.2 di bawah ini :



Gambar 4.2 Tampilan *Form* Menu Utama

4.3.3 Form Manajemen User

Form manajemen user digunakan untuk *me-manage* data *user*, yaitu menambah, meng-*edit*, dan menghapus data *user*.

Tombol **Baru** : Berfungsi untuk menambah (mengisi) data *user* yang belum di-*input*-kan sebelumnya, dengan cara mengisi keterangan data *user* di kolom yang telah disediakan.

Tombol **Simpan** : Berfungsi untuk menyimpan data *user* yang telah diisi pada dalam *database*.

Tombol **Edit** : Berfungsi untuk merubah/mengganti data *user* yang telah tersimpan di dalam *database* sebelumnya, dengan cara mencari data *user* pada bagian **Pencarian Data**, kemudian mengganti data *user*. Selanjutnya menekan kembali tombol simpan untuk proses penyimpanan data.

Tombol **Hapus** : Berfungsi untuk menghapus data *user* dengan cara memilih data *user* kemudian menekan tombol **Hapus**.

Tombol **Keluar** : Berfungsi untuk menutup *form*.

Form Manajemen User dapat dilihat pada gambar 4.3 di bawah ini :



Gambar 4.3 Tampilan *Form* Manajemen User

4.3.4 Form Ganti Password

Form ganti password digunakan untuk mengganti *password user*. Ketika form ini dijalankan, maka akan secara otomatis data user akan ditampilkan. *Password* dapat diganti dengan cara memasukkan *password* baru pada kolom yang telah disediakan, kemudian tekan tombol **simpan** untuk menyimpan data *password*. Untuk lebih jelasnya dapat dilihat pada gambar 4.4 di bawah ini :

Gambar 4.4 Tampilan *Form Ganti Password*

4.3.5 Form Input Indikator

Form Input Indikator digunakan untuk *me-manage* data Indikator, yaitu menambah, *meng-edit*, dan menghapus data Indikator.

Tombol **Baru** : Berfungsi untuk menambah (mengisi) data *Indikator* yang belum di-*input*-kan sebelumnya, dengan cara mengisi keterangan data *Indikator* di kolom yang telah disediakan.

Tombol **Simpan** : Berfungsi untuk menyimpan data Indikator yang telah diisi pada dalam *database*.

Tombol **Edit** : Berfungsi untuk merubah/mengganti data *Indikator* yang telah tersimpan di dalam *database* sebelumnya, dengan cara mencari data *Indikator* pada bagian **Pencarian Data**, kemudian mengganti data *Indikator*. Selanjutnya menekan kembali tombol simpan untuk proses penyimpanan data.

Tombol **Hapus** : Berfungsi untuk menghapus data *Indikator* dengan cara memilih data *Indikator* kemudian menekan tombol **Hapus**.

Tombol **Keluar** : Berfungsi untuk menutup *form*.

Form Input Indikator dapat dilihat pada gambar 4.5 di bawah ini :

No_Indikator	Nama_Indikator	Bobot_Indikator
IND-001	Sakit Lambung	0.4
IND-002	Sariawan	0.8
IND-003	Berdebar-debar	0.6
IND-004	Perubahan Suasana Hati	1
IND-005	Mudah Marah	1
IND-006	Merasa Tidak Nyaman	0.9
IND-007	Sulit Konsentrasi	0.8

Gambar 4.5 Tampilan *Form* Indikator

4.3.6 Form Input Kasus

Form Input Kasus digunakan untuk *me-manage* data Kasus, yaitu menambah, *meng-edit*, dan menghapus data Kasus.

Tombol **Baru** : Berfungsi untuk menambah (mengisi) data Kasus yang belum di-*input*-kan sebelumnya, dengan cara mengisi keterangan data Kasus di kolom yang telah disediakan.

Tombol **Simpan** : Berfungsi untuk menyimpan data Kasus yang telah diisi pada dalam *database*.

Tombol **Edit** : Berfungsi untuk merubah/mengganti data Kasus yang telah tersimpan di dalam *database* sebelumnya, dengan cara mencari data Kasus pada bagian **Pencarian Data**, kemudian mengganti data Kasus. Selanjutnya menekan kembali tombol simpan untuk proses penyimpanan data.

Tombol **Hapus** : Berfungsi untuk menghapus data Kasus dengan cara memilih data Kasus kemudian menekan tombol **Hapus**.

Tombol **Keluar** : Berfungsi untuk menutup *form*.

Form Input Kasus dapat dilihat pada gambar 4.6 di bawah ini :

Gambar 4.6 Tampilan *Form* Kasus

4.3.7 Form Input Penyakit

Form Input Penyakit digunakan untuk *me-manage* data Penyakit, yaitu menambah, meng-*edit*, dan menghapus data konsumen.

Tombol **Baru** : Berfungsi untuk menambah (mengisi) data Penyakit yang belum di-*input*-kan sebelumnya, dengan cara mengisi keterangan data Penyakit di kolom yang telah disediakan.

Tombol **Simpan** : Berfungsi untuk menyimpan data Penyakit yang telah diisi pada dalam *database*.

Tombol **Edit** : Berfungsi untuk merubah/mengganti data Penyakit yang telah tersimpan di dalam *database* sebelumnya, dengan cara mencari data Penyakit pada bagian **Pencarian Data**, kemudian mengganti data Penyakit. Selanjutnya menekan kembali tombol simpan untuk proses penyimpanan data.

Tombol **Hapus** : Berfungsi untuk menghapus data Penyakit dengan cara memilih data Penyakit kemudian menekan tombol **Hapus**.

Tombol **Keluar** : Berfungsi untuk menutup *form*.

Form Input Penyakit dapat dilihat pada gambar 4.7 di bawah ini :

Id_Penyakit	Nama_Penyakit	Peni
P-001	Stress Ringan	(ME
P-002	Stress Sedang	(Me
P-003	Stress Berat	(Me
P-004	Tidak Mengalami Stress	(ME

Gambar 4.7 Tampilan *Form* Penyakit

4.3.8 *Form* Kasus Baru

Form Input Data Kasus Baru digunakan untuk me-*manage* data Kasus Baru, yaitu menambah, meng-*edit*, dan menghapus data Kasus Baru.

Tombol **Baru** : Berfungsi untuk menambah (mengisi) data Kasus Baru yang belum di-*input*-kan sebelumnya, dengan cara mengisi keterangan data Kasus Baru di kolom yang telah disediakan.

Tombol **Simpan** : Berfungsi untuk menyimpan data Indikator yang telah diisi pada dalam *database*.

Tombol **Edit** : Berfungsi untuk merubah/mengganti data Kasus Baru yang telah tersimpan di dalam *database* sebelumnya, dengan cara mencari data Kasus Baru pada bagian **Pencarian Data**, kemudian mengganti data Kasus Baru.

Selanjutnya menekan kembali tombol simpan untuk proses penyimpanan data.

Tombol **Hapus** : Berfungsi untuk menghapus data Kasus Baru dengan cara memilih data Kasus Baru kemudian menekan tombol **Hapus**.

Tombol **Keluar** : Berfungsi untuk menutup *form*.

Tahap selanjutnya setelah menyimpan data kasus baru adalah menyimpan data indikator. Klik tombol *input* data indikator, kemudian menghitung *similarity* menggunakan tombol perhitungan CBR. Untuk lebih jelasnya dapat dilihat pada gambar 4.8 berikut ini :

Gambar 4.8 Tampilan *Form Input* Kasus Baru

4.3.9 Form Laporan

Form ini berfungsi untuk mencetak laporan hasil tes Penyakit. Pilih jenis cetak laporan, kemudian pilih jenis periode bulan atau tahun. Setelah itu tekan tombol **cetak**. Untuk lebih jelasnya dapat dilihat pada gambar 4.9 di bawah ini :



Gambar 4.9 Tampilan Form Cetak Laporan

Laporan ini digunakan untuk menampilkan laporan Kasus Baru. Laporan dapat disajikan berdasarkan periode tertentu. Untuk lebih jelasnya, bentuk laporan hasil tes Penyakit dapat dilihat pada gambar 4.10 di bawah ini :

RSJ Provinsi Lampung
Jl. Kurungan Nyawa, Gedung Tataan, Kabupaten Pesawaran

LAPORAN HASIL TEST PASIEN
PERBULAN

No	Id Kasus Baru	Nama Pasien	Tanggal	Kasus	Similitary / Kedekatan	Hasil
1	Ksb-001	Rudi	Feb-16-2017	K-002	0.145	Stress Sedang
2	Ksb-002	Safrizal	Feb-16-2017	K-002	0.145	Stress Sedang

Bandar Lampung, 16-February-2017
Dokter

(.....)

Gambar 4.10 Tampilan Laporan Kasus Baru

4.4 Pengujian

4.4.1 Uji Aplikasi *Black Box*

Penulis menggunakan metode *black box testing* untuk melakukan pengetesan pada sistem, dimana sistem dilihat langsung pada aplikasinya tanpa perlu mengetahui struktur programnya. Pengetesan ini dilakukan untuk menentukan bahwa program tersebut sudah layak atau belum untuk digunakan serta telah memenuhi kebutuhan yang yang diharapkan atau belum.

4.4.1.1 Tes Fungsionalitas

Pengetesan ini dilaksanakan dengan melihat hasil (*output*) yang dihasilkan apakah telah sesuai dengan yang diharapkan dengan menggunakan input yang berlainan. Pengetesan fungsionalitas dilakukan pada form Input Data Kasus.

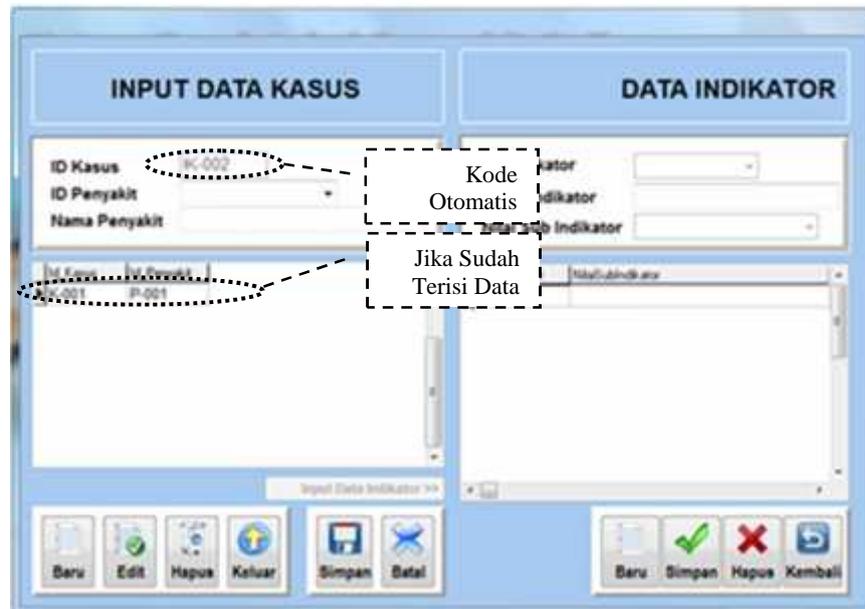
a. Penggunaan Tombol Baru

Ketika dilakukan klik pada tombol baru maka sistem melakukan pengosongan pada tiap-tiap orang dan secara otomatis menentukan nomor penerimaan sesuai dengan urutannya. Jika pada tabel belum terdapat data sama sekali, maka secara *default* nomor penerimaan akan diisikan “K-001” seperti pada gambar 4.11 di bawah ini :



Gambar 4.11 Tampilan Setelah Klik Tombol Baru (Data Kosong)

Pengujian tahap selanjutnya, dilakukan penggunaan tombol baru ketika data sudah ada pada tabel. Terlihat pengurutan nomor akan dilanjutkan pada “K-002” dan seterusnya. Seperti pada gambar 4.12 berikut ini :



Gambar 4.12 Tampilan Setelah Klik Tombol Baru (Data Terisi)

Hasil pengujian :

Tombol baru melakukan fungsinya seperti yang diharapkan.

b. Penggunaan Tombol Simpan

Sebelum menekan tombol simpan, terlebih dahulu dilakukan pengisian data. Tombol simpan akan melakukan proses menyimpan data ke dalam tabel pada *database*. Pengujian dilakukan dengan beberapa tahap. Pertama, dilakukan pengisian data pada *form* seperti pada gambar 4.13 berikut ini :

Gambar 4.13 Tampilan Sebelum Klik Tombol Simpan

Tahap selanjutnya, dilakukan penggunaan tombol simpan. Data pada *form* tersimpan ke table seperti terlihat pada gambar 4.14 di bawah ini :

Gambar 4.14 Tampilan Setelah Klik Tombol Simpan

Hasil pengujian :

Tombol simpan melakukan fungsinya seperti yang diharapkan.

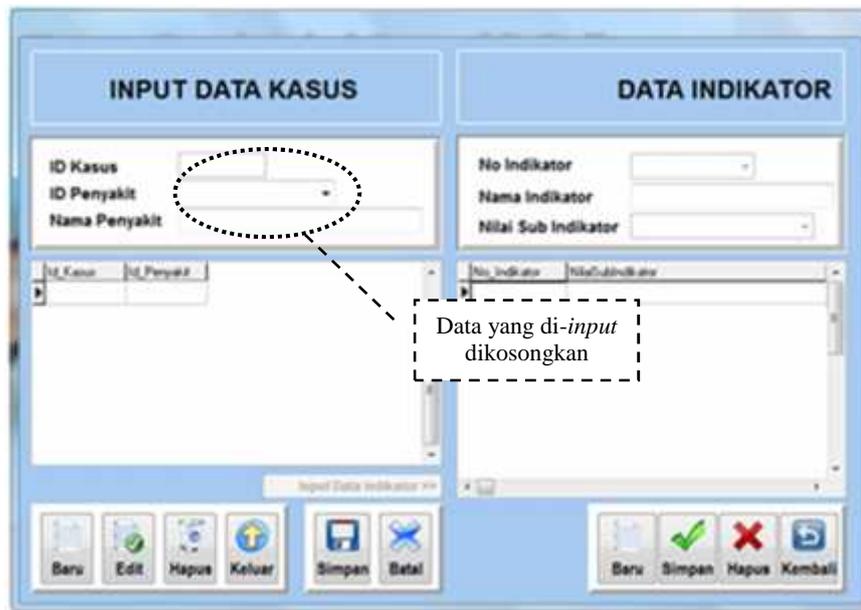
c. Penggunaan Tombol Batal

Tombol batal merupakan tombol pilihan lain selain tombol simpan, tombol batal digunakan jika *user* memutuskan untuk tidak menyimpan data baru ataupun data hasil *edit*. Efek yang dihasilkan berupa pengosongan form. Sama seperti tombol baru, hanya saja tidak ada pemberian kode otomatis. Tahap pertama, data di-*input*-kan pada *form*. Untuk lebih jelasnya, dapat dilihat pada gambar 4.15 di bawah ini :

The image shows a software interface with two main sections: 'INPUT DATA KASUS' and 'DATA INDIKATOR'. The 'INPUT DATA KASUS' section contains three input fields: 'ID Kasus' (empty), 'ID Penyakit' (containing 'P-001'), and 'Nama Penyakit' (containing 'Stress Ringan'). The 'DATA INDIKATOR' section contains three input fields: 'No Indikator' (empty), 'Nama Indikator' (empty), and 'Nilai Sub Indikator' (empty). Below these fields are two empty tables. At the bottom of the interface, there are two sets of buttons. The left set includes 'Baru', 'Edit', 'Hapus', and 'Keluar'. The right set includes 'Baru', 'Simpan', 'Hapus', and 'Kembali'. A dashed box highlights the input fields with the text 'Data di-input pada form'.

Gambar 4.15 Tampilan Sebelum Klik Tombol Batal

Tahap selanjutnya, pengujian dilakukan dengan menekan tombol batal. Maka form yang sudah terisi dengan data akan dikosongkan. Untuk lebih jelasnya, pengujian tombol batal dapat dilihat pada gambar 4.16 sebagai berikut :



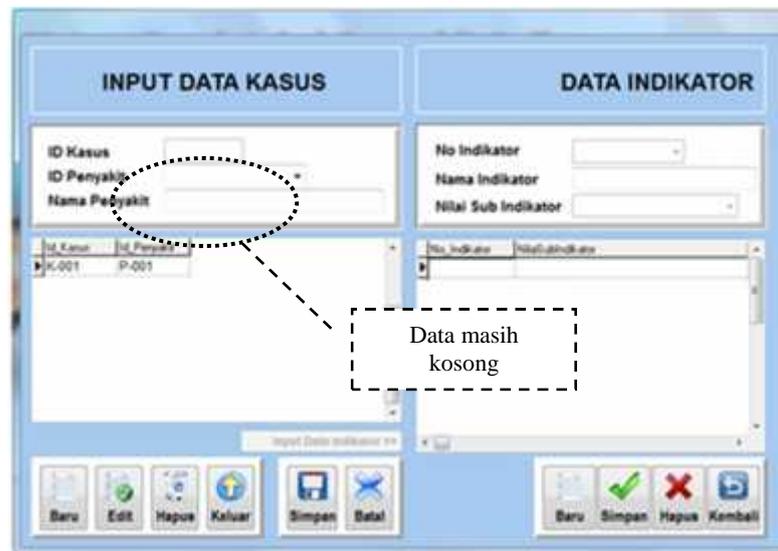
Gambar 4.16 Tampilan Setelah Klik Tombol Batal

Hasil pengujian :

Tombol Batal melakukan fungsinya seperti yang diharapkan.

d. Penggunaan Tombol *Edit*

Tombol *edit* merupakan tombol yang digunakan ketika akan merubah data yang sudah tersimpan pada *database*. Tahap awal, belum terdapat data yang akan diubah. Seperti pada gambar 4.17 di bawah ini :



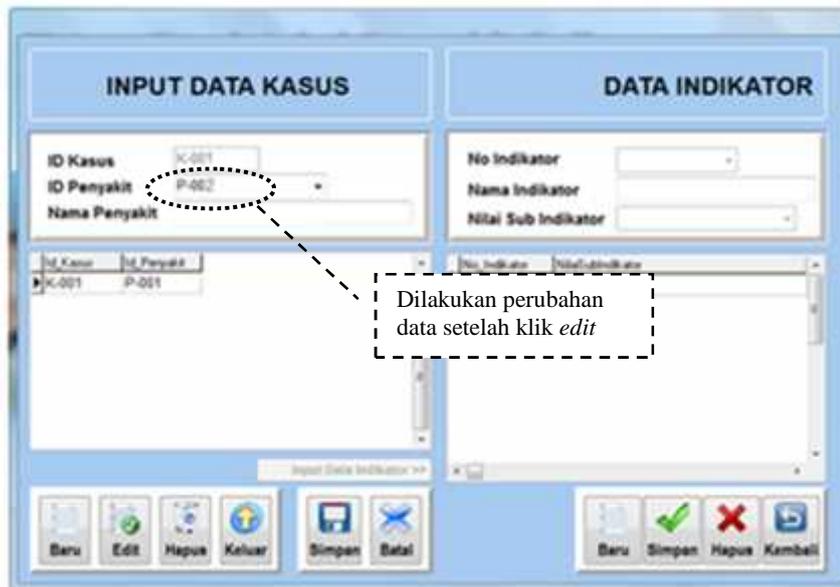
Gambar 4.17 Tampilan Pengujian Fungsionalitas Tombol *Edit1*

Tahap Selanjutnya, dilakukan pemilihan data terlebih dahulu yaitu dengan melakukan pencarian menggunakan tombol navigasi atau klik data secara langsung pada *dbgrid*. Seperti terlihat pada gambar 4.18 di bawah ini :



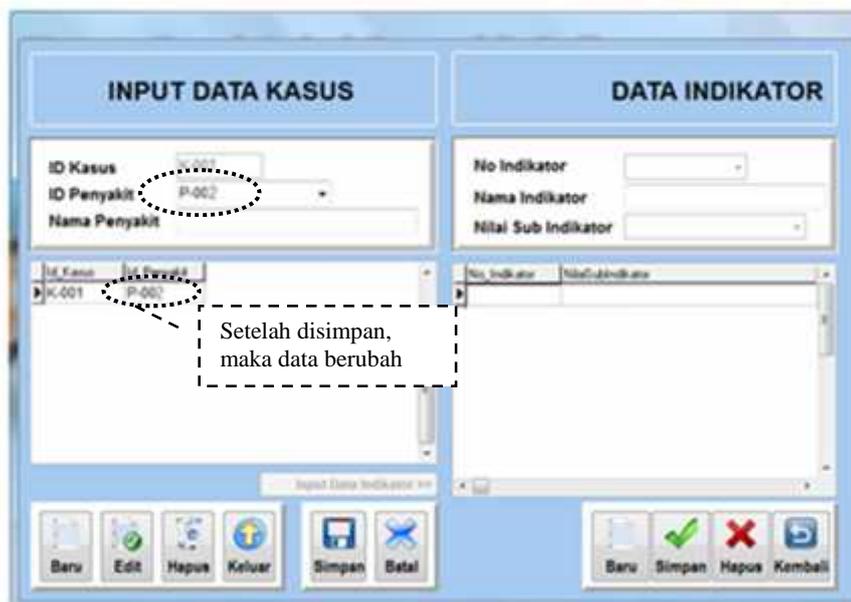
Gambar 4.18 Tampilan Pengujian Fungsionalitas Tombol *Edit2*

Selanjutnya, Setelah data tampil, maka dilakukan peng-*edit*-an data sesuai dengan data yang sudah sesuai. Untuk lebih jelasnya, dapat dilihat pada gambar 4.19 sebagai berikut :



Gambar 4.19 Tampilan Pengujian Fungsionalitas Tombol *Edit*2

Tahap pengujian yang terakhir adalah proses simpan yaitu dilakukan klik pada tombol simpan. Untuk lebih jelasnya dapat dilihat pada gambar 4.20 di bawah ini :



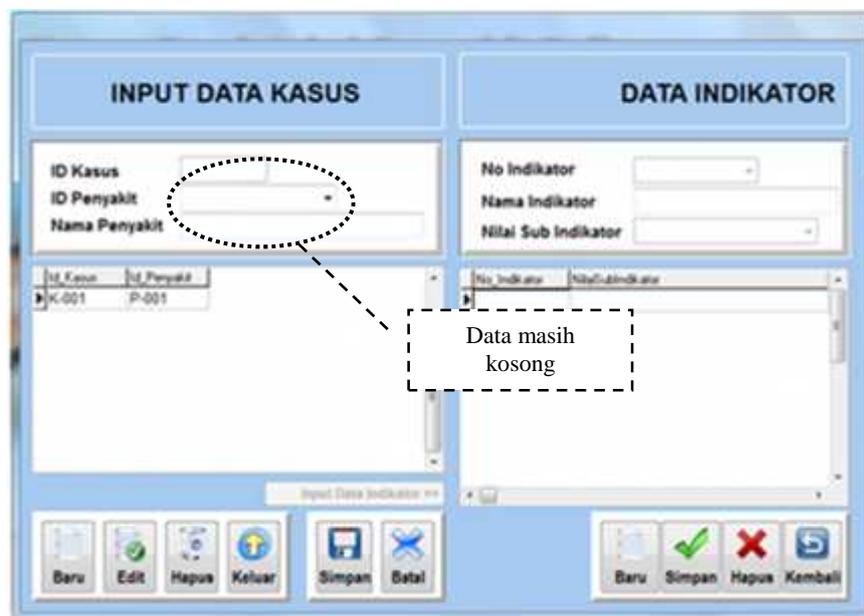
Gambar 4.20 Tampilan Pengujian Fungsionalitas Tombol *Edit*4

Hasil pengujian :

Tombol *edit* melakukan fungsinya seperti yang diharapkan.

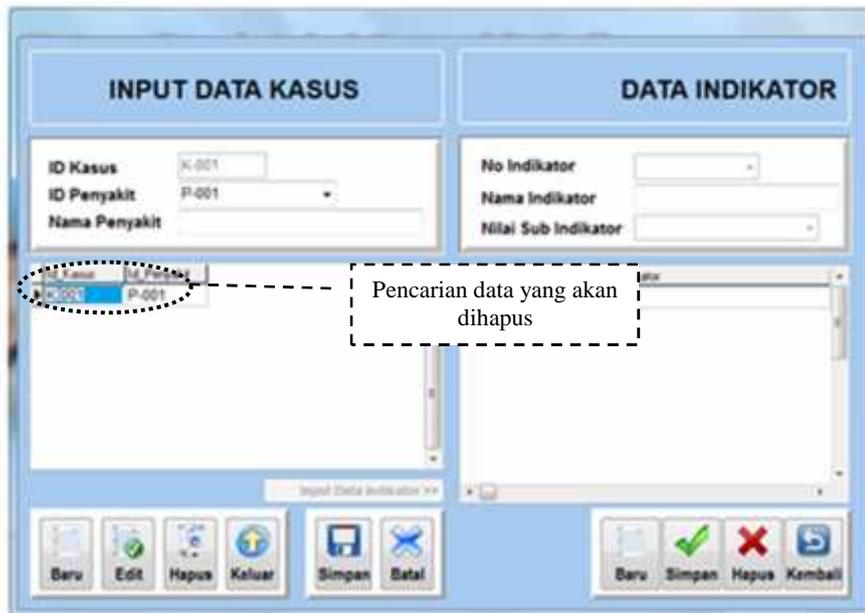
e. Penggunaan Tombol Hapus

Hampir sama seperti proses mengubah data, hanya saja tidak ada proses penyimpanan melainkan penghapusan data dari *database*. Tahap awal, belum terdapat data yang akan diubah. Seperti pada gambar 4.21 seperti berikut ini :



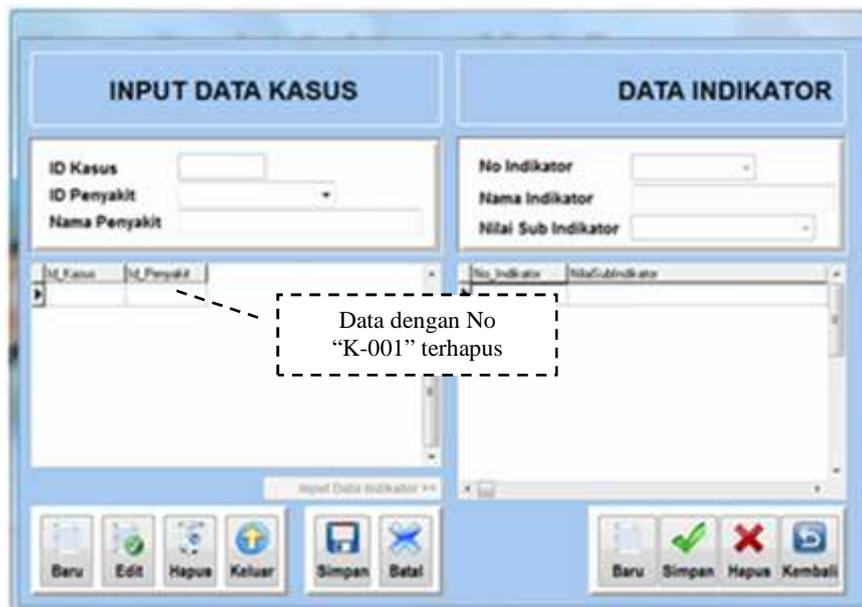
Gambar 4.21 Tampilan Pengujian Fungsionalitas Tombol Hapus1

Tahap Selanjutnya, dilakukan pemilihan data terlebih dahulu yaitu dengan melakukan pencarian menggunakan tombol navigasi atau klik data secara langsung pada *dbgrid*. Seperti terlihat pada gambar 4.22 berikut ini :



Gambar 4.22 Tampilan Pengujian Fungsionalitas Tombol Hapus2

Setelah data tampil, tahap pengujian yang terakhir yaitu dilakukan klik pada tombol hapus, maka data yang dimaksud akan terhapus. Untuk lebih jelasnya dapat dilihat pada gambar 4.23 sebagai di bawah ini :



Gambar 4.23 Tampilan Pengujian Fungsionalitas Tombol Hapus3

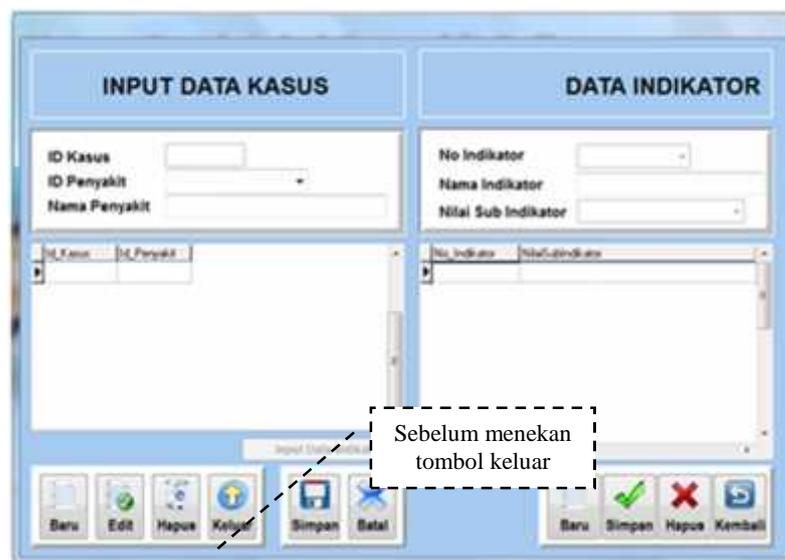
Hasil pengujian:

Tombol hapus melakukan fungsinya seperti yang diharapkan.

f. Penggunaan Tombol Keluar

Tombol keluar berfungsi untuk menutup *form input* Penerimaan.

Selanjutnya kembali ke tampilan menu utama. Tampilan sebelum ditekan tombol keluar dapat dilihat pada gambar 4.24 di bawah ini :



Gambar 4.24 Tampilan Pengujian Fungsionalitas Tombol Keluar1

Sedangkan setelah ditekan tombol keluar, maka *form input* Penerimaan akan ditutup, sebagai gantinya ditampilkan *form* menu utama. Seperti pada gambar 4.25 di bawah ini :



Gambar 4.25 Tampilan Pengujian Fungsionalitas Tombol Keluar2

Hasil pengujian :

Tombol Keluar melakukan fungsinya seperti yang diharapkan.

4.4.1.2 Tes stress

Tes ini dilaksanakan untuk mengetahui sampai dimana kemampuan sistem dalam mengatasi kesalahan. Pengujian ini dilakukan masih pada form input Penerimaan. Pengujian dilakukan dengan melakukan *input* data yang tidak valid.

Test case :

- Jumlah diisi data bertipe string.
- Jumlah diisi data negatif.

Pengujian dilakukan dengan menginput data pada jumlah dengan tipe yang tidak seharusnya, seperti data string dan penggunaan simbol minus. Pengujian dapat dilihat pada gambar 4.26 di bawah ini :

Gambar 4.26 Tampilan Pengujian Tes *Stress*

Tiap pemberian nilai bobot ataupun nilai kedekatan, tidak diinputkan manual menggunakan *keyboard* melainkan menggunakan nilai yang sudah terdapat pada *combobox*, maka tidak akan terjadi kesalahan dalam menentukan nilai angka.

Hasil pengujian :

Program ini sudah memiliki kontrol terhadap *human error* sehingga kesalahan kesalahan yang mungkin terjadi akibat penginputan data bisa dihindari.

4.4.2 Uji Pengguna

Evaluasi Pengguna yang dilakukan peneliti yaitu dengan memberikan kuisisioner kepada Dokter, Asisten Dokter. Kuisisioner dibuat untuk Pengguna agar peneliti dapat mengetahui apakah sistem layak digunakan atau perlu dilakukan perbaikan sistem. Pada kuisisioner yang dibuat peneliti memberikan lima pertanyaan yang sama kepada tiap pengguna. Setelah itu dilakukan proses perhitungan apakah sistem layak, perlu perbaikan kembali atau tidak layak digunakan. Pada kuisisioner yang dibuat, peneliti menggunakan rumus dengan menghitung nilai dari masing-masing pertanyaan. Setiap pertanyaan masing-masing terdapat *range* nilai atau frekuensi antara 0-4, dari hasil nilai tersebut dijumlahkan kemudian dikalikan dengan jumlah pertanyaan atau jumlah kategori. setelah itu didapat hasil dari nilai tiap responden, maka dari nilai tersebut dihitung nilai akhir dengan rumus sigma dari jumlah frekuensi dan jumlah kategori dibagi jumlah responden, maka didapat hasil antara 50-100. Dari nilai tersebut terdapat keterangan bahwa sistem layak, perlu perbaikan atau tidak layak untuk digunakan.

Pertanyaan-pertanyaan dapat dilihat pada tabel 4.1 seperti di bawah ini :

Pegguna : Dokter						
No	Kategori Penilaian	Sangat Baik	Baik	Cukup	Kurang	Sangat Kurang
1	Tampilan aplikasi apakah sudah memenuhi kebutuhan <i>user</i> ?					
2	Proses penginputan data kasus, indikator, dan kasus baru memenuhi permintaan <i>user</i> ?					
3	Proses perhitungan CBR sudah memenuhi permintaan <i>user</i> ?					
4	Proses pencarian data dan pencetakan laporan sudah dapat dilakukan dengan baik oleh <i>user</i> ?					
5	Hasil laporan memenuhi permintaan <i>user</i> ?					
Pegguna: Asisten Dokter						
No	Kategori Penilaian	Sangat Baik	Baik	Cukup	Kurang	Sangat Kurang
1	Tampilan aplikasi apakah sudah memenuhi kebutuhan <i>user</i> ?					
2	Proses penginputan data kasus, indikator, dan kasus baru memenuhi permintaan <i>user</i> ?					

3	Proses perhitungan CBR sudah memenuhi permintaan <i>user</i> ?					
4	Proses pencarian data dan pencetakan laporan sudah dapat dilakukan dengan baik oleh <i>user</i> ?					
5	Hasil laporan memenuhi permintaan <i>user</i> ?					

Tabel 4.1 Kuisisioner Evaluasi Pengguna

Penilaian mempunyai *range* yang digunakan untuk mewakili keterangan penilaian. *Range* penilaian dapat dilihat pada tabel 4.2 di bawah ini :

Keterangan Penilaian	Range Nilai
Sangat Baik	4
Baik	3
Cukup	2
Kurang	1
Sangat Kurang	0

Tabel 4.2 Range Penilaian

Nilai akhir dihitung dengan rumus = $\frac{(fxi)}{N}$

Ket :

f = jumlah frekuensi untuk setiap variabel

i = kategori dalam variabel

N = jumlah responden

Setelah nilai akhir dihitung, selanjutnya dilakukan penilaian berdasarkan keterangan pada tabel 4.3 sebagai berikut :

Nilai Akhir	Ket
85-100	Layak
70-84	Perbaikan
50-69	Tidak Layak

Tabel 4.3 Peringkat Kelayakan Sistem

Semua nilai dihitung dan dikelompokkan berdasarkan keterangan pada masing-masing responden. Untuk lebih jelasnya, hasil penilaian dapat dilihat pada tabel 4.4 di bawah ini :

Responden	Sangat baik	Baik	Cukup	Kurang	Sangat Kurang	Jumlah Frekuensi	Nilai
Dokter	16	3	0	0	0	19	95
Asisten Dokter	16	3	0	0	0	19	95

Tabel 4.4 Hasil Penilaian Variabel Responden

Ket : Nilai Responden (f_{xi}) = jumlah frekuensi x jumlah kategori

$$\text{Nilai Akhir} = \frac{(f_{xi})}{N} = \frac{95+95}{2} = \frac{190}{2} = \mathbf{95}$$

Hasil Akhir =

Layak

Berdasarkan hasil evaluasi yang telah dilakukan peneliti dengan memberikan kuisioner yang digunakan untuk mengetahui penilaian aplikasi sistem yang dibuat. Maka dapat disimpulkan bahwa sistem yang dibuat sudah memenuhi standar kelayakan.

4.4.3 Solusi dan Penanganan Stres

Dari hasil pengujian menggunakan sistem ini, peneliti memberikan solusi atau penanganan sesuai dengan tingkatan stres pada siswa seperti berikut ini :

a) Stres Ringan

Apabila didiagnosa mengalami stres ringan, hanya diperlukan relaksasi karena pada dasarnya stres ringan dapat memberikan stimulasi positif bagi penderitanya untuk lebih berkembang.

b) Stres Sedang

Apabila mengalami stres sedang, dapat ditangani dengan 3 cara dilihat dari faktor penyebab stres yang dialami oleh remaja tersebut. Penanganannya antara lain sebagai berikut :

1. Faktor Biologis, dapat diukur dengan *neurotransmitter*, perlu diberikan obat – obatan seperti psikofarmaka yang tentunya diberikan oleh tenaga profesional.
2. Faktor Pola Asuh, untuk pertahanan mental pada remaja atau pelajar sebaiknya orang tua melakukan pengasuhan yang lebih sehat yaitu dengan komunikasi adult to adult (dewasa ke dewasa) agar respon yang diberikan oleh orang tua lebih dapat diterima.
3. Faktor Lingkungan, berkaitan dengan pelajaran, ekonomi dan lain – lain. Hal ini dapat ditangani dengan cara sederhana yaitu menggeser persepsi tersebut terhadap stresor itu sendiri apabila lingkungan itu tidak dapat kita kelola.

c) Stres Berat

Apabila mengalami Stres Berat, penanganan memerlukan tenaga ahli/profesional agar dapat diberikan tindakan lanjut secara medis atau direhabilitasi jika dirasa perlu.