

**LAPORAN HASIL PEMBUATAN  
APLIKASI  
(PROYEK AKHIR SEMESTER GASAL )**



Disusun Oleh:

Nama	: Yulmenina Rizqa Labibah
Kelas	: X PPLG 2
NIS	: 258776
Judul Proyek	: Sistem Pemantauan Kesehatan (VIRUS TRACKER)

**PENGEMBANGAN PERANGKAT LUNAK DAN GIM  
SMK NEGERI 1 KANDEMAN  
TAHUN PELAJARAN 2025/2026**

## 1. DESKRIPSI PROYEK

Proyek ini merupakan pembuatan **Aplikasi Sistem Pemantauan Kesehatan (Virus Tracker)** sederhana menggunakan **C# Windows Forms**. Aplikasi ini berfungsi untuk mencatat data pasien, memantau gejala, memilih obat sesuai penyakit, serta menghitung total biaya dan kembalian transaksi.

Aplikasi ini dibuat untuk mendukung simulasi pemantauan kesehatan di masa depan “**Warga Kota Bikini Bottom Tahun 2050**”, dengan tema futuristik dan tampilan ilustratif.

### A. Tujuan

Proyek ini bertujuan untuk membuat aplikasi pemantauan kesehatan warga menggunakan Windows Forms. Sistem ini mencatat data pasien seperti nama, usia, jenis kelamin, dan status penyakit. Setelah itu aplikasi juga dapat menampilkan rekomendasi obat, menghitung total biaya, kembalian serta menampilkan data pasien dalam tabel.

### B. Nama aplikasi

**“Sistem Pemantauan Kesehatan (VIRUS TRACKER) Warga Kota”**

### C. Fitur Utama Aplikasi

#### 1. Input Data Pasien

- 1) Pada kolom Nama, ketik nama pasien.
- 2) Pada kolom Usia, masukkan angka usia pasien.
- 3) Pada Jenis Kelamin, pilih *Laki-laki* atau *Perempuan*.
- 4) Pada kolom Status Penyakit, ketik penyakit pasien.
- 5) Setelah semua kolom terisi → tekan tombol Tambah Data.  
→ Setelah itu Informasi pasien langsung muncul di tabel DataGridView di bagian atas.

#### 2. Menambah Data pasien

- 1) Isi semua kolom (Nama, Usia, Jenis Kelamin, Penyakit).
- 2) Klik tombol Tambah Data.
- 3) Data otomatis masuk ke tabel pada bagian atas.
- 4) TextBox kembali kosong dan siap diisi data pasien berikutnya.

#### 3. Menghapus Data Pasien

- 1) Klik salah satu baris pada tabel pasien.

- 2) Tekan tombol Hapus (yang berwarna merah).
- 3) Akan muncul pertanyaan “apakah anda akan menghapus data ini” .
- 4) Jika memilih ya maka data di baris tersebut akan hilang.
- 5) Jika “tidak” maka data nya akan tetap tertinggal atau tidak hilang.
- 6) Ketika ingin menghapus semua data:
  - a. Klik tombol yang sama (hapus).
  - b. pilih “hapus semua data”.
  - c. jika ya “ data akan terhapus semua ”.
  - d. jika tidak “maka data akan masih tetap ”.

#### 4. Mencari data

- 1) Masukkan data yang ingin dicari.
- 2) Tekan tombol **Cari**.
- 3) Tabel akan menampilkan hanya data yang cocok dengan kata pencarian.
- 4) Jika tidak ada yang cocok → data tidak ditemukan.

#### 5. Memilih Obat berdasarkan penyakit

- 1) Setelah mengisi *Data*, lihat kolom **OBAT**.
- 2) Klik dropdown daftar obat.
- 3) Pilih obat yang sesuai untuk penyakit pasien.

#### 6. Menentukan jumlah obat

- 1) Pada kolom Jumlah, masukkan jumlah obat yang ingin dibeli pasien.
- 2) Kolom ini berupa angka.

#### 7. Menghitung harga Obat

- 1) Setelah memilih obat dan mengisi jumlah, klik tombol TOTAL  
(atau total akan muncul otomatis jika sudah diprogram begitu).
- 2) Total harga akan muncul di kotak total.

#### 8. Melakukan pembayaran dan menentukan kembalian

- 1) Masukkan uang pembayaran pada kolom **Bayar**.
- 2) Tekan tombol **BAYAR**.
- 3) Program otomatis menghitung:
  - Total harga obat
  - Kembalian uang

Hasilnya muncul di kolom **Kembalian**.

#### 9. Keluar aplikasi

- 1) Klik tombol keluar maka aplikasi akan tertutup.

## **2. LANDASAN TEORI C# YANG DIGUNAKAN**

### **A. Bahasa Pemrograman C#**

C# dibuat oleh Microsoft, mendukung OOP, dan digunakan dalam berbagai pengembangan perangkat lunak seperti aplikasi desktop, game (Unity), web backend, dan IoT.

### **B. Windows Formz**

Platform GUI yang menyediakan kontrol seperti:

- TextBox
- Label
- Button
- ComboBox
- DataGridView
- PictureBox

WinForms memberikan kemudahan drag-and-drop.

### **C. Konsep Pemrograman**

1. **Variabel dan tipe data** → untuk menyimpan seluruh informasi yang dibutuhkan program Pada sistem pemantauan kesehatan.
  - a. Tipe Data → Tipe data adalah jenis atau kategori nilai yang bisa disimpan dalam sebuah variabel pada bahasa pemrograman, seperti int untuk angka, string untuk nama, char bisa jadi untuk karakter atau jenis kelamin.
  - b. Variabel → Variabel adalah tempat atau wadah untuk menyimpan data di dalam program. Variabel berfungsi seperti kotak penyimpanan di mana kita bisa menyimpan nilai tertentu, memberi nama pada kotak tersebut, lalu menggunakan isinya kapan pun dalam program, seperti nama, usia, jenis kelamin.

```
string nama;  
int usia;  
string jenisKelamin;  
string penyakit;  
int jumlah;  
int hargaObat;  
int total;  
int bayar;  
int kembali;
```

2. **input dan Output** → untuk menerima data dari pengguna, sedangkan output digunakan untuk menampilkan hasil dari proses program.

**Input meliputi:**

- Mengisi nama pasien
- Mengisi usia
- Memilih jenis kelamin
- Mengisi status penyakit
- Memilih obat
- Mengisi jumlah obat
- Mengisi uang pembayaran

**Output meliputi:**

- Menampilkan data pasien pada tabel
- Menampilkan harga obat
- Menampilkan total harga
- Menampilkan kembalian
- Menampilkan pesan kesalahan atau petunjuk

Dengan konsep Input dan Output, program dapat menerima data dari pengguna dan mengolahnya menjadi informasi yang muncul di layar.

3. **Operator** → untuk melakukan perhitungan dan pengecekan kondisi dalam program.

**Operator yang digunakan meliputi:**

- **Operator aritmatika** untuk menghitung total harga obat dan kembalian
- **Operator perbandingan** untuk menentukan apakah uang pembayaran cukup atau tidak
- **Operator logika** untuk memeriksa beberapa syarat sekaligus

Dengan operator, program dapat melakukan perhitungan seperti “harga obat × jumlah”, atau menentukan apakah pembayaran sudah cukup.

4. **Percabangan (If - else)** → digunakan agar program dapat menentukan keputusan berdasarkan kondisi tertentu.

Digunakan pada fitur-fitur seperti:

- Memeriksa apakah usia valid
- Memeriksa apakah semua data sudah diisi sebelum ditambahkan
- Menentukan apakah uang pembayaran mencukupi
- Menentukan obat yang cocok berdasarkan nama penyakit
- Menampilkan pesan ketika data pencarian tidak ditemukan

Dengan percabangan, program mampu menyesuaikan tindakan yang harus dilakukan pada setiap kondisi.

5. **Array/List** → untuk menyimpan banyak data pasien dalam satu wadah.

List berfungsi untuk:

- Menyimpan seluruh data pasien
- Menghapus data pasien tertentu
- Menampilkan data pasien ke dalam tabel
- Menjadi sumber data saat pencarian dilakukan

Dengan List, program dapat menampung data pasien sebanyak apa pun sesuai kebutuhan pengguna.

6. **Perulangan / Looping** → digunakan ketika program perlu memproses banyak data secara berulang.

Digunakan untuk:

- Menampilkan semua pasien ke dalam tabel
- Mencari data berdasarkan kata kunci
- Memperbarui tabel setelah data dihapus atau ditambah

Dengan perulangan, program dapat memproses banyak baris data dengan efisien tanpa menulis perintah yang sama berulang kali.

7. **Fungsi / Method** → untuk memisahkan proses agar program lebih rapi dan mudah dipahami.

Contoh proses yang dibuat sebagai method:

- Menambah data pasien
- Menghapus data dari tabel
- Mengosongkan semua kolom input
- Menghitung total harga obat
- Menghitung uang kembalian
- Menampilkan kembali seluruh data ke tabel

Dengan method, setiap proses memiliki tugas masing-masing sehingga program lebih terstruktur.

8. **Event Handling** → Event handling adalah konsep pemrograman yang membuat program merespons tindakan pengguna.

Contoh event:

- Menekan tombol Tambah, Hapus, Bayar, atau Cari
- Mengubah pilihan pada daftar obat
- Mengklik baris pada tabel
- Mengubah nilai jumlah obat

Event membuat program “hidup” dan merespons secara langsung ketika pengguna berinteraksi dengan aplikasi.

```
0 references
private void pictureBox1_Click(object sender, EventArgs e)
{
}

1 reference
private void label3_Click(object sender, EventArgs e)
{
}

1 reference
private void label5_Click(object sender, EventArgs e)
{
}

1 reference
private void textBox2_TextChanged(object sender, EventArgs e)
{
}
```

9. **DataGridView** → untuk menampilkan banyak data dalam bentuk tabel agar lebih mudah dibaca.

Digunakan untuk:

- Menampilkan daftar pasien
- Menyimpan data yang sudah ditambahkan
- Menghapus data tertentu dengan memilih baris
- Menampilkan hasil pencarian

Dengan DataGridView, program dapat mengelola data pasien secara terstruktur.

### 3. PERANCANGAN PROGRAM

#### A. Kebutuhan dalam merancang

1) Kebutuhan fungsional:

- Input pasien



- Penyakit → obat otomatis
  - Hitung harga
  - Hitung kembalian
  - Simpan data
  - Hapus data
- 2) Kebutuhan non-fungsional:
- Tampilan mudah dipahami
  - Aplikasi ringan
  - Tidak lag.

## B. Struktur Data

Kolom	Deskripsi
1	Nama
2	Usia
3	Jenis Kelamin
4	Penyakit
5	Obat
6	Harga
7	Jumlah
8	Total
9	Bayar
10	Kembali

**C. Algoritma dan pengolahan data** → untuk menyelesaikan suatu masalah. Dalam konteks pembuatan aplikasi Virus Tracker, algoritma berperan sebagai dasar logika bagaimana aplikasi berjalan dari awal hingga akhir.

Algoritma utama menggambarkan alur kerja keseluruhan aplikasi, yaitu:

- Mulai program

- Input data pasien (Nama, Usia, Jenis Kelamin, Penyakit)
- Tentukan obat otomatis berdasarkan penyakit
- Tentukan harga obat
- Input jumlah obat yang ingin dibeli
- Lakukan perhitungan total harga
- Input jumlah uang pembayaran pasien
- Hitung kembalian
- Simpan data ke tabel DataGridView
- Jika user ingin menghapus data → hapus baris yang dipilih
- Jika user ingin keluar → tutup aplikasi
- Program selesai

#### 4. IMPLEMENTASI PROGRAM

##### A. Deklarasi Variabel

```
string nama;
int usia;
string jenisKelamin;
string penyakit;
int jumlah;
int hargaObat;
int total;
int bayar;
int kembali;
```

##### B. Input Data Pasien

Untuk me Input Data Pasien, tidak ada kode pemrogramannya karena menggunakan Textbox yang disatukan dalam groupBox.

```

0 references
private void button1_Click(object sender, EventArgs e)
{
}

1 reference
private void groupBox1_Enter(object sender, EventArgs e)
private void button3_Click(object sender, EventArgs e)
{
    dataGridView1.Rows.Add(txtNama.Text, txtUsia.Text, txtJK.Text, txtStatus.Text);
}

0 references
private void label13_Click(object sender, EventArgs e)
{
}

0 references
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
}

```

→ Dengan itu maka kita dapat mengisi data secara langsung dan otomatis.

### C. Menambahkan Data

```

private void button3_Click(object sender, EventArgs e)
{
    dataGridView1.Rows.Add(txtNama.Text, txtUsia.Text, txtJK.Text, txtStatus.Text);
}

```

→ Setelah berhasil menyelesaikan pengInputan data yang selanjutnya di lakukan adalah dengan klik “Tambah Data”.

### D. Mencari Data

```

1 reference
private void button1_Click_1(object sender, EventArgs e)
{
    // Ambil teks dari textbox nama
    string namaCari = txtNama.Text.Trim();

    // Pastikan input tidak kosong
    if (string.IsNullOrEmpty(namaCari))
    {
        MessageBox.Show("Masukkan nama pasien yang ingin dicari!", "peringatan", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    // Tandai apakah data di temukan
    bool ditemukan = false;

    // loop setiap baris di DataGridView
    foreach (DataGridViewRow row in dataGridView1.Rows)
    {
        if (row.IsNewRow) continue;
    }
}

```

```
// cek kolom pertama (nama pasien)
string namaPasien = row.Cells[0].Value?.ToString() ?? "";

if (namaPasien.Equals(namaCari, StringComparison.OrdinalIgnoreCase))
{
    row.Selected = true;
    dataGridView1.CurrentCell = row.Cells[0];
    MessageBox.Show($"Data pasien '{namaCari}' ditemukan!", "Informasi", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    ditemukan = true;
    break;
}

// jika tidak ditemukan
if (!ditemukan)
{
    MessageBox.Show($"data pasien '{namaCari}' tidak ditemukan.", "hasil pencarian", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
}
}
```

→ Jika ingin menemukan data secara cepat maka yang harus kita lakukan adalah dengan meng input data pasien tersebut lalu klik “Cari” maka selamat pasien yang anda cari telah ditemukan, sebaliknya jika salah meng input data pasien atau pasien tersebut tidak ada di dalam data maka “Data pasien tidak ditemukan”.

### E. Menghapus Data

```
private void button2_Click(object sender, EventArgs e)
{
    AllClear();
}
```

→ Setelah berhasil menambahkan data, maka kita harus menghapus Inputan data tersebut karena sudah otomatis akan tersimpan di DataGridView nya. Agar dapat meng Input data yang lain dan yang akan datang.

### F. Menghapus salah satu Data

```

1 reference
private void button4_Click(object sender, EventArgs e)
{
    // pastikan ada baris yang dipilih
    if (dataGridView1.SelectedRows.Count > 0)
    {
        // konfirmasi ke pengguna sebelum menghapus
        DialogResult result = MessageBox.Show("Apakah kamu yakin untuk menghapus data pasien yang kamu pilih?"
        , "konfirmasi hapus", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (result == DialogResult.Yes)
        {
            // Hapus baris yang dipilih (baris pertama yang disorot)
            foreach (DataGridViewRow row in dataGridView1.SelectedRows)
            {
                if (!row.IsNewRow)
                {
                    dataGridView1.Rows.Remove(row);
                }
            }
        }
    }
}

```

```

    MessageBox.Show("Data pasien berhasil dihapus.", "Informasi", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
}
else
{
    MessageBox.Show("Pilih terlebih dahulu data pasien yang ingin dihapus!", "peringatan",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
}

```

→ Jika ingin menghapus salah satu data kita bisa klik “Hapus” pada bagian kanan bawah di DataGridView, jika ada box pesan klik tombol “ya” maka data tersebut akan menghilang, sebaliknya jika tidak jadi untuk menghapus klik tombol “Tidak”.

## G. Menghapus semua data

```

projek_akhir_yulmenina.Form1
button4_Click(object sender, EventArgs e)
{
    if (dataGridView1.Rows.Count > 0)
    {
        DialogResult reultoi = MessageBox.Show("Apakah kamu yakin ingin menghapus semua data pasien?",
        "konfirmasi hapus semua",
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
        if (reultoi == DialogResult.Yes)
        {
            dataGridView1.Rows.Clear();

            MessageBox.Show("Semua data pasien telah dihapus.", "informasi", MessageBoxButtons.OK
            , MessageBoxIcon.Information);
        }
    }
    else
    {
        // jika memilih no
        MessageBox.Show("PENGHAPUSAN DIBATALKAN.", "dibatalkan", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
    }
}

```

→ Setelah berhasil menghapus salah satu data maka akan ada box pesan yang menawarkan “Apakah anda ingin menghapus semua data?” jika “tidak” data yang

lain akan tetap berada di DataGridView, akan tetapi jika “ya” otomatis semua data akan menghilang.

#### **H. Menentukan jenis Obat dengan harga yang sudah tertera**

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
```

```
{
    if (cmbObat.Text == "Paracetamol")
    {
        txtHarga.Text = "15000"; // Rp.15.000
    }
    else if (cmbObat.Text == "Amlodipin")
    {
        txtHarga.Text = "20000"; // Rp.20.000
    }
    else if (cmbObat.Text == "Simvastatin")
    {
        txtHarga.Text = "45000"; // Rp.45.000
    }
    else if (cmbObat.Text == "flunarizine")
    {
        txtHarga.Text = "80000"; // Rp.80.000
    }
    else if (cmbObat.Text == "Amoxilin")
    {
        txtHarga.Text = "30000"; // Rp.30.000
    }
}
```

→ Setelah menambahkan data kita bisa memilih jenis obat yang sesuai dengan penyakit dan otomatis harganya akan masuk ke dalam textbox.

#### **I. Menentukan jumlah obat dan menghitung totalnya**

```

1 reference
private void txtJumlah_TextChanged(object sender, EventArgs e)
{
    // jika user mengisi menggunakan huruf maka program akan memperingatkan
    if (txtHarga.Text != "" && txtJumlah.Text != "")
    {
        if (int.TryParse(txtJumlah.Text, out int jumlah))
        {
            int harga = int.Parse(txtHarga.Text);
            int total = harga * jumlah;
            txtTotal.Text = $"{total}";
        }
        else
        {
            MessageBox.Show("Di isi pakai angka ya!");
        }
    }
}

```

→ Setelah memilih obat kita harus menentukan jumlah obatnya. Setelah menentukan jumlah obat yang kita perlukan, maka otomatis total harga nya akan muncul.

#### **J. Membayar obat dan menghitung kembalian**

```

private void btnByr_Click(object sender, EventArgs e)
{
    // Jika user sudah bertansaksi maka akan muncul peringatan

    int total = Convert.ToInt32(txtTotal.Text);
    int bayar = Convert.ToInt32(txtbayar.Text);

    if (total > bayar)
    {
        MessageBox.Show("uang anda kurang!");
        return;
    }

    else
    {
        MessageBox.Show("Transaksi anda berhasil");
    }
}

```

```

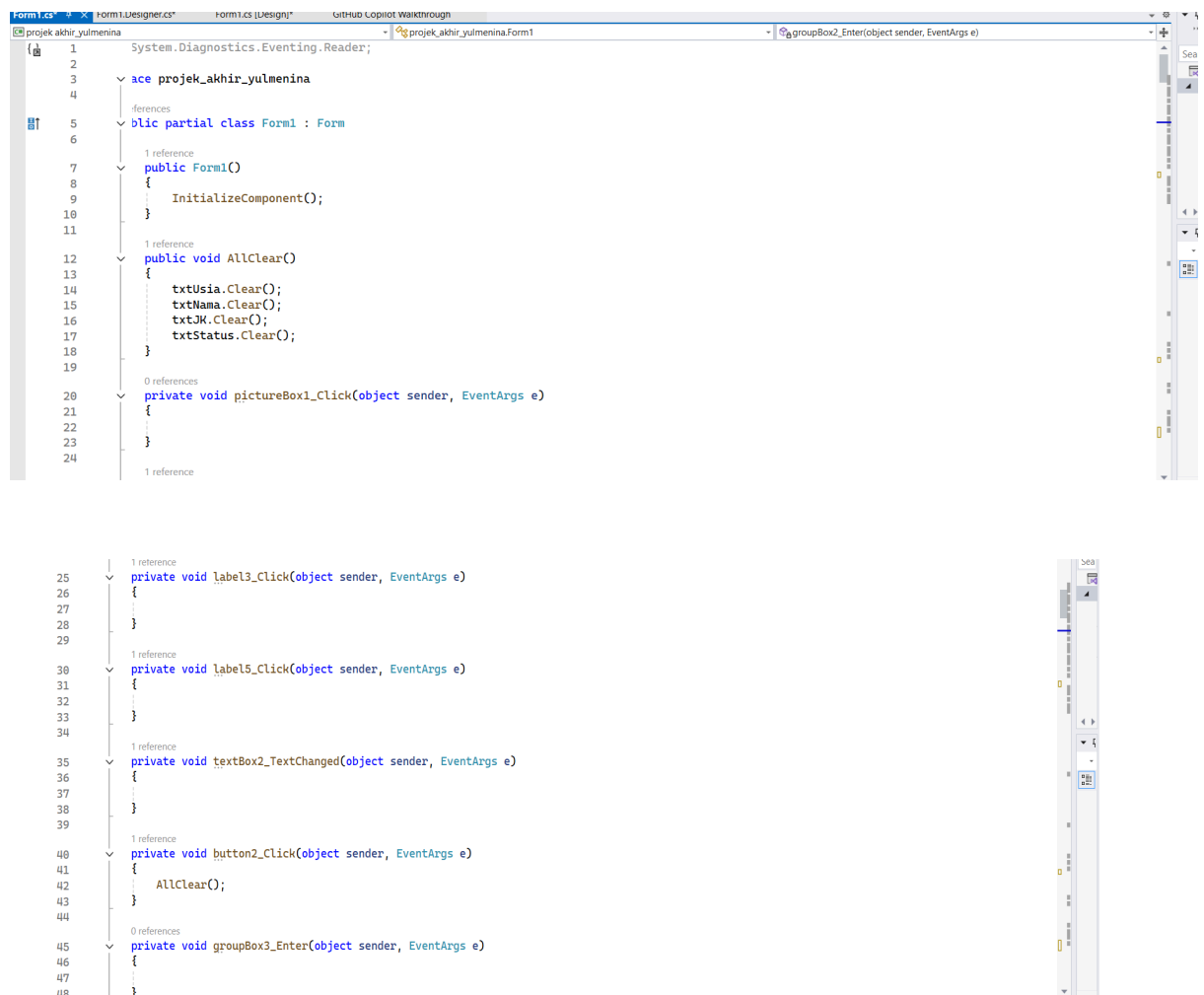
double kembalian = bayar - total;
txtKembali.Text = kembalian.ToString();

}

```

→ Selanjutnya langkah yang kita lakukan adalah membayar obatnya dengan klik “BAYAR”, jika uangnya kurang maka “Mohon maaf uang anda kurang”, sebaliknya jika uang anda lebih dari total harga obat tersebut, anda akan menerima kembaliannya.

## 5. LAMPIRAN





```

50 0 references
51 private void label10_Click(object sender, EventArgs e)
52 {
53 }
54
55 0 references
56 private void label8_Click(object sender, EventArgs e)
57 {
58 }
59
60 0 references
61 private void groupBox2_Enter(object sender, EventArgs e)
62 {
63 }
64
65 0 references
66 private void textBox8_TextChanged(object sender, EventArgs e)
67 {
68 }
69
70 0 references
71 private void button1_Click(object sender, EventArgs e)
72 {
73 }
74
75 1 reference
76 private void groupBox1_Enter(object sender, EventArgs e)
77 {
78 }
79
80 0 references
81 private void label13_Click(object sender, EventArgs e)
82 {
83 }
84
85 0 references
86 private void checkBox1_CheckedChanged(object sender, EventArgs e)
87 {
88 }
89
90 1 reference
91 private void button3_Click(object sender, EventArgs e)
92 {
93     dataGridView1.Rows.Add(txtNama.Text, txtUsia.Text, txtJK.Text, txtStatus.Text);
94 }
95
96 1 reference
97 private void button5_Click(object sender, EventArgs e)
98 {
99     DialogResult result = MessageBox.Show("Apakah kamu yakin untuk keluar dari aplikasi?"
100     , "konfirmasi keluar", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
101     if (result == DialogResult.Yes)
102     {
103         Application.Exit();
104     }
105
106 1 reference
107 private void txtUsia_TextChanged(object sender, EventArgs e)
108 {
109 }
110
111 1 reference
112 private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
113 {
114 }
115
116 1 reference
117 private void txtJK_TextChanged(object sender, EventArgs e)
118 {
119 }
120
121 0 references

```

```

95 private void button5_Click(object sender, EventArgs e)
96 {
97     DialogResult result = MessageBox.Show("Apakah kamu yakin untuk keluar dari aplikasi?"
98     , "konfirmasi keluar", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
99     if (result == DialogResult.Yes)
100     {
101         Application.Exit();
102     }
103 }
104
105 1 reference
106 private void txtUsia_TextChanged(object sender, EventArgs e)
107 {
108 }
109
110 1 reference
111 private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
112 {
113 }
114
115 1 reference
116 private void txtJK_TextChanged(object sender, EventArgs e)
117 {
118 }
119
120 0 references

```

```

120 private void txtStatus_TextChanged(object sender, EventArgs e)
121 {
122 }
123
124
125 1 reference
126 private void button4_Click(object sender, EventArgs e)
127 {
128     // pastikan ada baris yang dipilih
129     if (dataGridView1.SelectedRows.Count > 0)
130     {
131         // konfirmasi ke pengguna sebelum menghapus
132         DialogResult result = MessageBox.Show("Apakah kamu yakin untuk menghapus data pasien yang kamu pilih?"
133         , "Konfirmasi hapus", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
134         if (result == DialogResult.Yes)
135         {
136             // Hapus baris yang dipilih (baris pertama yang disorot)
137             foreach (DataGridViewRow row in dataGridView1.SelectedRows)
138             {
139                 if (!row.IsNewRow)
140                 {
141                     dataGridView1.Rows.Remove(row);
142                 }
143             }
144             MessageBox.Show("Data pasien berhasil dihapus.", "Informasi", MessageBoxButtons.OK,
145             MessageBoxIcon.Information);
146         }
147     }
148 }

```

```

145 }
146 }
147 else
148 {
149     MessageBox.Show("Pilih terlebih dahulu data pasien yang ingin dihapus!", "peringatan",
150     MessageBoxButtons.OK, MessageBoxIcon.Warning);
151 }
152
153 if (dataGridView1.Rows.Count > 0)
154 {
155     DialogResult resultoi = MessageBox.Show("Apakah kamu yakin ingin menghapus semua data pasien?",
156     "konfirmasi hapus semua",
157     MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
158     if (resultoi == DialogResult.Yes)
159     {
160         dataGridView1.Rows.Clear();
161         MessageBox.Show("Semua data pasien telah dihapus.", "informasi", MessageBoxButtons.OK
162         , MessageBoxIcon.Information);
163     }
164 }
165 else
166 {
167     // jika memilih no
168     MessageBox.Show("PENGAHAPUSAN DIBATALKAN.", "dibatalkan", MessageBoxButtons.OK,
169     MessageBoxIcon.Information);
170 }
171
172
173

```

```

173 }
174 }
175
176
177 1 reference
178 private void button1_Click_1(object sender, EventArgs e)
179 {
180     // Ambil teks dari textbox nama
181     string namaCari = txtNama.Text.Trim();
182
183     // Pastikan input tidak kosong
184     if (string.IsNullOrEmpty(namaCari))
185     {
186         MessageBox.Show("Masukkan nama pasien yang ingin dicari!", "peringatan", MessageBoxButtons.OK, MessageBoxIcon.Warning);
187         return;
188     }
189     // Tandai apakah data di temukan
190     bool ditemukan = false;
191
192     // loop setiap baris di DataGridView
193     foreach (DataGridViewRow row in dataGridView1.Rows)
194     {
195         if (row.IsNewRow) continue;
196
197         // cek kolom pertama (nama pasien)
198         string namaPasien = row.Cells[0].Value?.ToString() ?? "";
199
200         if (namaPasien.Equals(namaCari, StringComparison.OrdinalIgnoreCase))
201         {
202             // Data ditemukan
203         }
204     }
205 }

```

```

200     {
201         row.Selected = true;
202         dataGridView1.CurrentCell = row.Cells[0];
203         MessageBox.Show($"Data pasien '{namaCari}' ditemukan!", "Informasi", MessageBoxButtons.OK
204             , MessageBoxIcon.Information);
205         ditemukan = true;
206         break;
207     }
208
209     // jika tidak ditemukan
210     if (!ditemukan)
211     {
212         MessageBox.Show($"data pasien '{namaCari}' tidak ditemukan.", "hasil pencarian", MessageBoxButtons.OK,
213             MessageBoxIcon.Information);
214     }
215 }
216
217
218
219 1 reference
220 private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
221 {
222     if (cmbObat.Text == "Paracetamol")
223     {
224         txtHarga.Text = "15000"; // Rp.15.000
225     }
226     else if (cmbObat.Text == "Amlodipin")
227     {
228         txtHarga.Text = "20000"; // Rp.20.000

```

```

228     }
229     else if (cmbObat.Text == "Simvastatin")
230     {
231         txtHarga.Text = "45000"; // Rp.45.000
232     }
233     else if (cmbObat.Text == "Flunarizine")
234     {
235         txtHarga.Text = "80000"; // Rp.80.000
236     }
237     else if (cmbObat.Text == "Amoxilin")
238     {
239         txtHarga.Text = "30000"; // Rp.30.000
240     }
241 }
242
243
244
245 1 reference
246 private void label7_Click(object sender, EventArgs e)
247 {
248 }
249
250 1 reference
251 private void textBox1_TextChanged(object sender, EventArgs e)
252 {
253 }
254

```

```

254
255 1 reference
256 private void btnByr_Click(object sender, EventArgs e)
257 {
258     // Jika user sudah bertansaksi maka akan muncul peringatan
259
260     int total = Convert.ToInt32(txtTotal.Text);
261     int bayar = Convert.ToInt32(txtbayar.Text);
262
263     if (total > bayar)
264     {
265         MessageBox.Show("uang anda kurang!");
266         return;
267     }
268     else
269     {
270         MessageBox.Show("Transaksi anda berhasil");
271     }
272
273     double kembalian = bayar - total;
274     txtKembali.Text = kembalian.ToString();
275
276 }
277
278
279 1 reference
280 private void label6_Click(object sender, EventArgs e)
281 {

```

```

280     private void txtJumlah_TextChanged(object sender, EventArgs e)
281     {
282     }
283
284     1 reference
285     private void txtJumlah_TextChanged(object sender, EventArgs e)
286     {
287         // jika user mengisi menggunakan huruf maka program akan memperingatkan
288         if (txtHarga.Text != "" && txtJumlah.Text != "")
289         {
290             if (int.TryParse(txtJumlah.Text, out int jumlah))
291             {
292                 int harga = int.Parse(txtHarga.Text);
293                 int total = harga + jumlah;
294                 txtTotal.Text = $"{total}";
295             }
296             else
297             {
298                 MessageBox.Show("Di isi pakai angka ya!");
299             }
300         }
301     }
302
303     1 reference
304     private void textBox2_TextChanged_1(object sender, EventArgs e)
305     {
306     }
307
308     1 reference
309     private void txtTotal_TextChanged(object sender, EventArgs e)
310     {
311     }
312
313     1 reference
314     private void txtHarga_TextChanged(object sender, EventArgs e)
315     {
316     }
317
318     1 reference
319     private void txtKembali_TextChanged(object sender, EventArgs e)
320     {
321     }
322
323

```



MITRA GREEN LIFE  
MEDICAL CENTER

## SISTEM PEMANTAUAN KESEHATAN (VIRUS TRACKER)

### WARGA KOTA BIKINI BOTTOM TAHUN 2050

#### INPUT DATA PASIEN

Nama   
Usia   
Jenis kelamin   
Status penyakit

	Nama Pasien	Usia	Jenis Kelamin	penyakit
•				

note: setelah diketahui apa penyakitnya silahkan pilih obat yang sesuai

OBAT:

Harga: 
Jumlah

TOTAL:

