Pengembangan Sistem Pencatatan Pasien Terintegrasi Berbasis C di Klinik X

Subtitle as needed ***(paper subtitle)***

Rafi Ananta Alden

Muhammad Zaki Fazansyah

Program Studi Teknik Elektro

Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung (STEI ITB)

Bandung, Indonesia

[13222087@mahasiswa.itb.ac.id](mailto:13222087@mahasiswa.itb.ac.id)

18322018@mahasiswa.itb.ac.id

Authors Name/s per 2nd Affiliation (*Author*)

line 1 (of *Affiliation*): dept. name of organization

line 2-name of organization, acronyms acceptable

line 3-City, Country

line 4-e-mail address if desired

**Font size 10 (Times New Roman)**

*Abstract:* This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. *\*CRITICAL: Do Not Use Symbols, Special Characters, or Math in Paper Title or Abstract*.

*Keywords*: Component; formatting; style; styling; insert (key words)

# Pendahuluan

## Latar Belakang

Pada era digital saat ini, efisiensi dan akurasi dalam pengelolaan data kesehatan sangatlah penting. Klinik-klinik kesehatan, baik kecil maupun besar, menghadapi tantangan dalam mengelola informasi pasien secara efektif. Penggunaan sistem manual atau berbasis kertas sering kali menyebabkan kesalahan, duplikasi data, dan kesulitan dalam pengaksesan informasi yang cepat dan tepat.

Klinik X, sebagai salah satu penyedia layanan kesehatan, mengalami permasalahan terkait. Pengelolaan data pasien yang mencakup informasi pribadi, riwayat medis, diagnosis, dan catatan perawatan sering kali memakan waktu dan rawan menghadapi kesalahan jika tidak didukung oleh sistem yang terkomputerisasi. Oleh karena itu, diperlukan sebuah aplikasi yang dapat membantu mempermudah pengelolaan data pasien secara lebih efisien dan akurat.

## Tujuan

Tujuan dari dibuatnya aplikasi pencatatan pasien ini adalah:

1. **Meningkatkan Efisiensi Pengelolaan Data**: Mempermudah proses pencatatan, pengubahan, pencarian, dan penghapusan data pasien.
2. **Meminimalisasi *Human Error***: Mengurangi risiko kesalahan dalam pencatatan data yang sering terjadi pada sistem manual.
3. **Meningkatkan Aksesibilitas Informasi**: Memastikan data pasien dapat diakses dengan cepat dan mudah oleh petugas medis yang berwenang.
4. **Menyediakan Analisis Data**: Memfasilitasi analisis data pasien untuk membantu pengambilan keputusan medis dan manajerial.
5. **Menyimpan Data Secara Terpusat**: Mengintegrasikan semua data pasien dalam satu sistem terpusat yang dapat diakses oleh berbagai departemen di klinik.

## Pentingnya Aplikasi

Pengembangan aplikasi pencatatan pasien berbasis bahasa pemrograman C ini penting bagi Klinik X karena beberapa alasan:

* **Efisiensi Operasional**: Aplikasi ini akan mengurangi beban administrasi dan meningkatkan produktivitas staf klinik dengan mengotomatisasi banyak tugas rutin.
* **Keakuratan Data**: Dengan menggunakan sistem terkomputerisasi, risiko kesalahan data dapat diminimalisasi yang pada akhirnya akan meningkatkan kualitas pelayanan kepada pasien.
* **Keamanan Data**: Sistem yang terkomputerisasi dengan baik dapat menyediakan mekanisme keamanan yang lebih baik untuk melindungi data pasien dari akses yang tidak sah.
* **Kepuasan Pasien**: Dengan pengelolaan data yang lebih baik, klinik dapat memberikan layanan yang lebih cepat dan akurat kepada pasien sehingga dapat meningkatkan kepuasan pasien.
* **Dasar untuk Pengembangan Lebih Lanjut**: Aplikasi ini dapat menjadi dasar bagi pengembangan lebih lanjut menuju sistem yang lebih kompleks dan terintegrasi seperti rekam medis elektronik (EMR) yang lebih komprehensif.

Dengan latar belakang, tujuan, dan pentingnya aplikasi ini, diharapkan proyek pengembangan sistem pencatatan pasien di Klinik X dapat berjalan dengan baik dan memberikan manfaat yang signifikan bagi klinik dan pasien.

# Rumusan Masalah

# Studi Literatur

# Metodologi Pembuatan Aplikasi

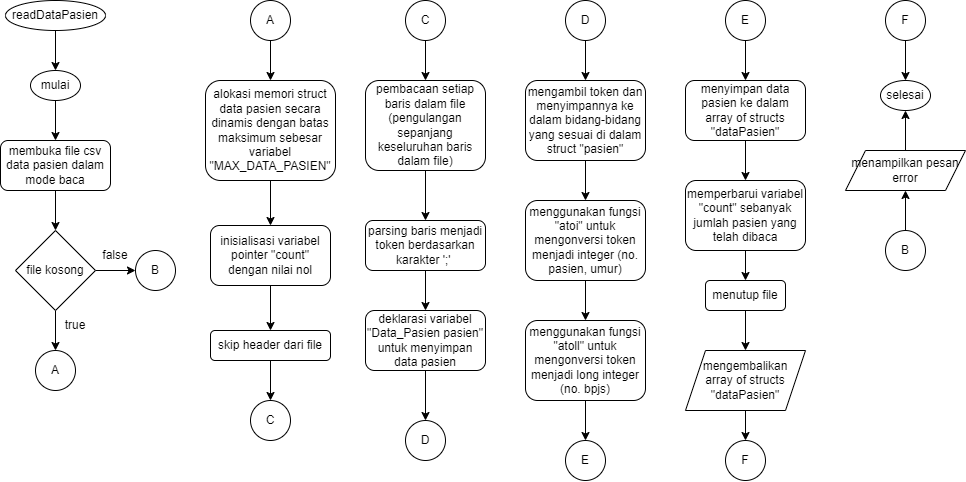
Bagian ini berisi *flowchart*, *Data Flow Diagram* (DFD), dan pemaparan mengenai *Graphical User Interface* (GUI) yang diintegrasikan ke dalam program yang dibuat.

1. *Flowchart*

### Program Utama dan Beberapa Fungsi Penunjang

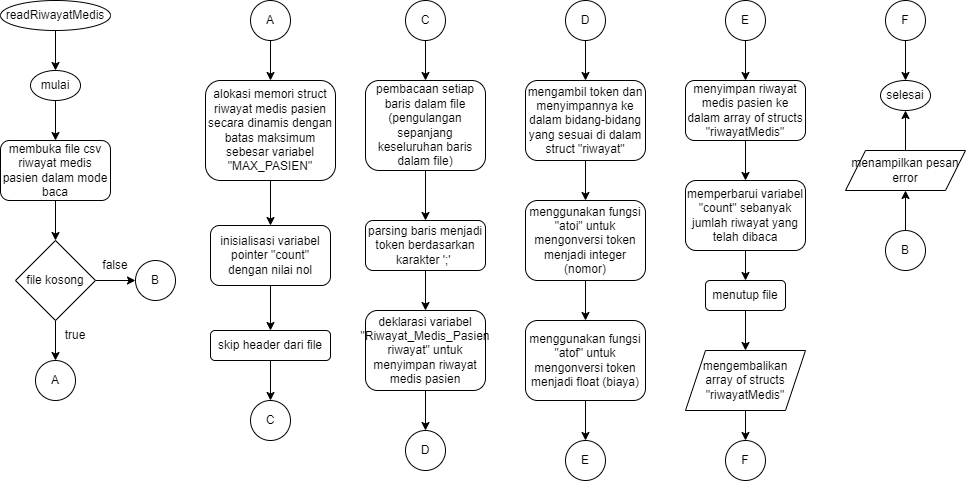
#### Program Utama

#### Fitur Baca Data Pasien



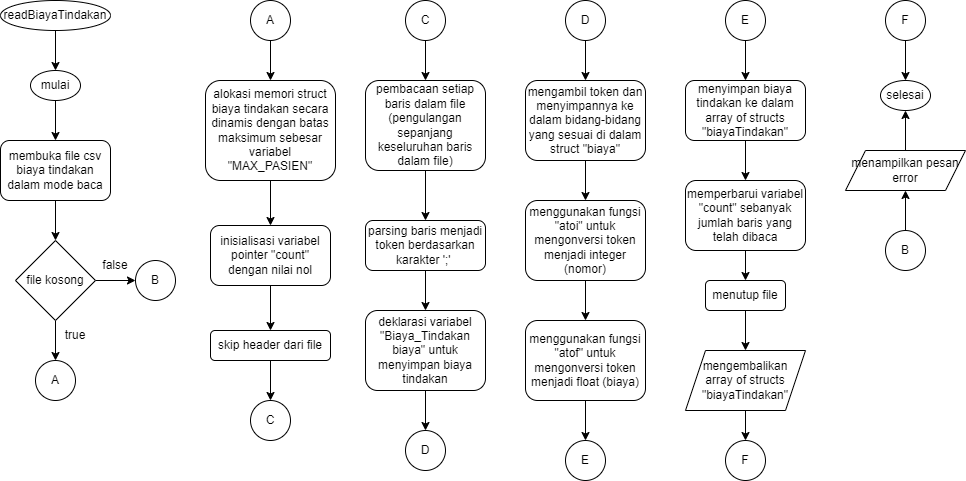
Gambar Flowchart Fitur Baca Data Pasien

#### Fitur Baca Riwayat Medis Pasien



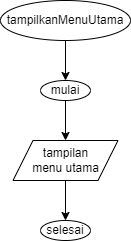
Gambar Flowchart Fitur Baca Riwayat Medis Pasien

#### Fitur Baca Biaya Tindakan



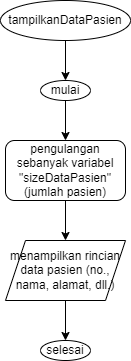
Gambar Flowchart Fitur Baca Biaya Tindakan

#### Fitur Tampilkan Menu Utama



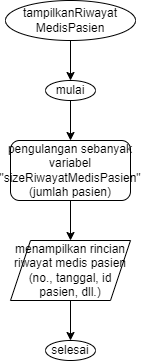
Gambar Flowchart Fitur Tampilkan Menu Utama

#### Fitur Tampilkan Data Pasien



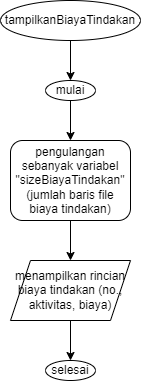
Gambar Flowchart Fitur Tampilkan Data Pasien

#### Fitur Tampilkan Riwayat Medis Pasien



Gambar Flowchart Fitur Tampilkan Riwayat Medis Pasien

#### Fitur Tampilkan Biaya Tindakan



Gambar Flowchart Fitur Tampilkan Biaya Tindakan

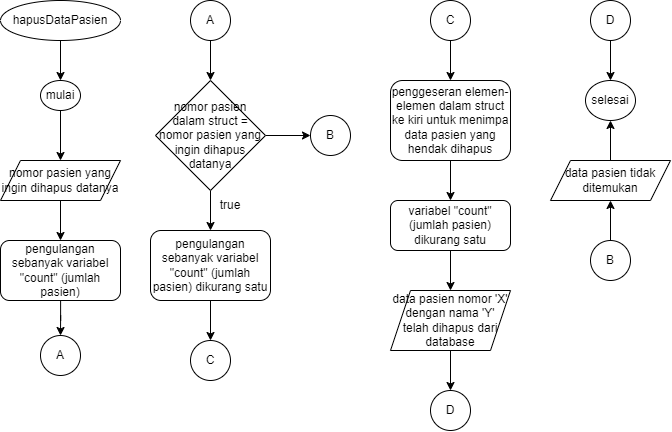
#### Fitur Pembersihan Buffer Input

### Manajemen Data Pasien

#### Fitur Tambah Data Pasien

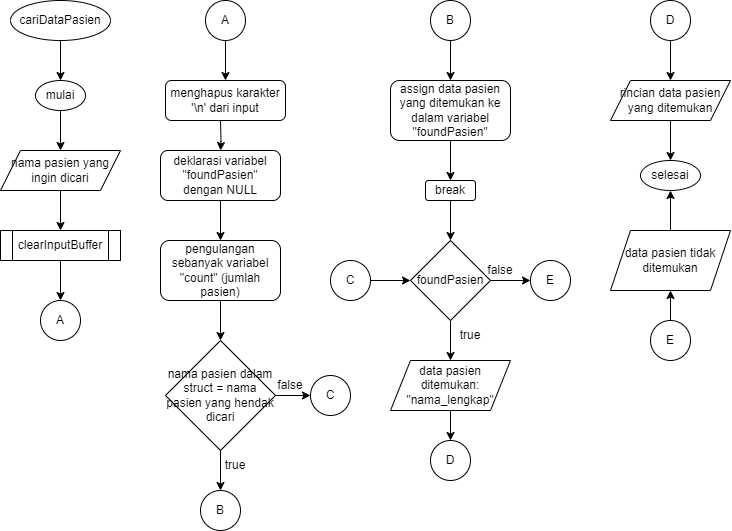
#### Fitur Ubah Data Pasien

#### Fitur Hapus Data Pasien



Gambar 4-A-2-c Flowchart Fungsi “hapusDataPasien”

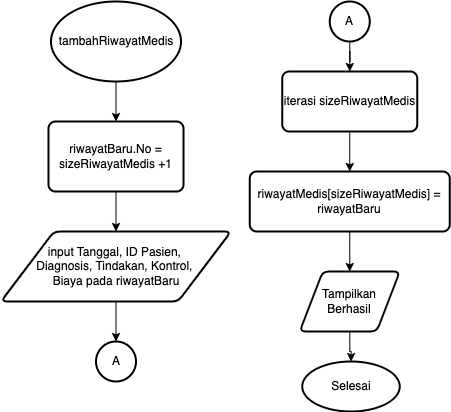
#### Fitur Cari Data Pasien



Gambar 4-A-2-d Flowchart Fungsi “cariDataPasien”

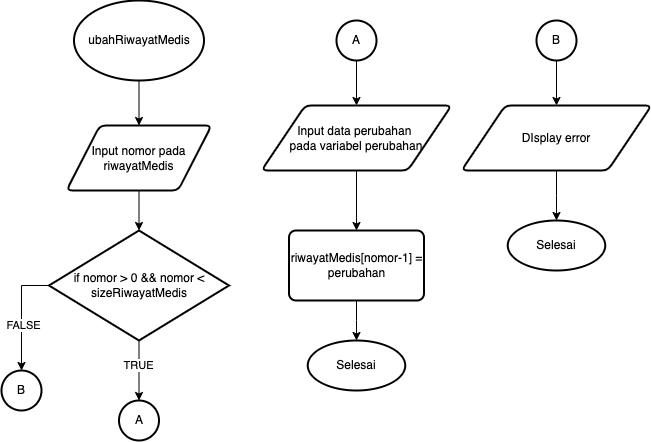
### Riwayat Medis dan Tindakan

#### Fitur Tambah Riwayat Keterangan, Diagnosis, dan Tindakan



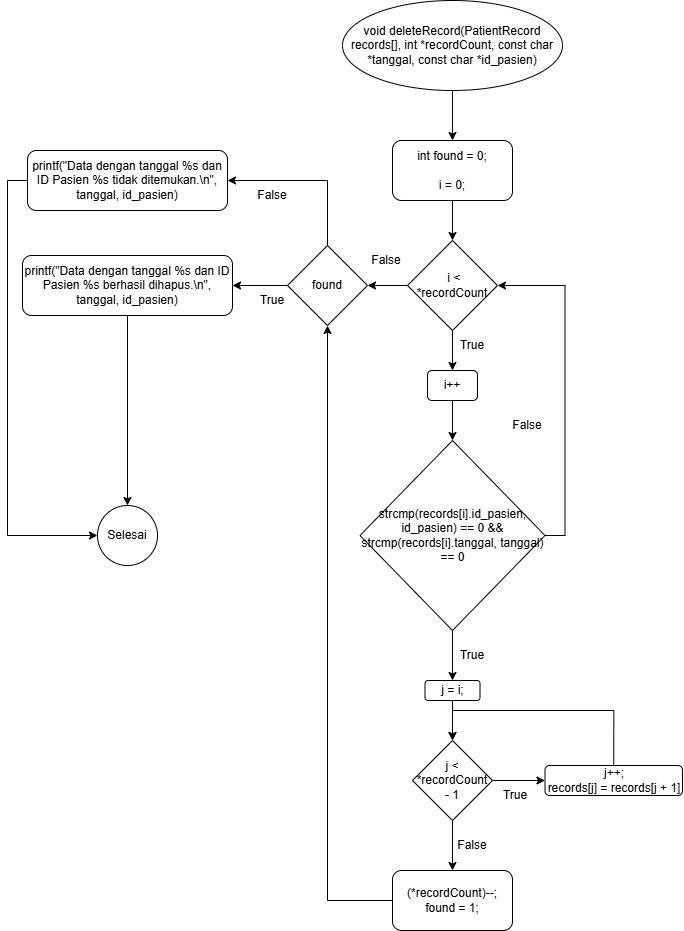
Gambar 4-A-3-a Flowchart Fungsi “tambahRiwayatMedis”

#### Fitur Ubah Riwayat Keterangan, Diagnosis, dan Tindakan

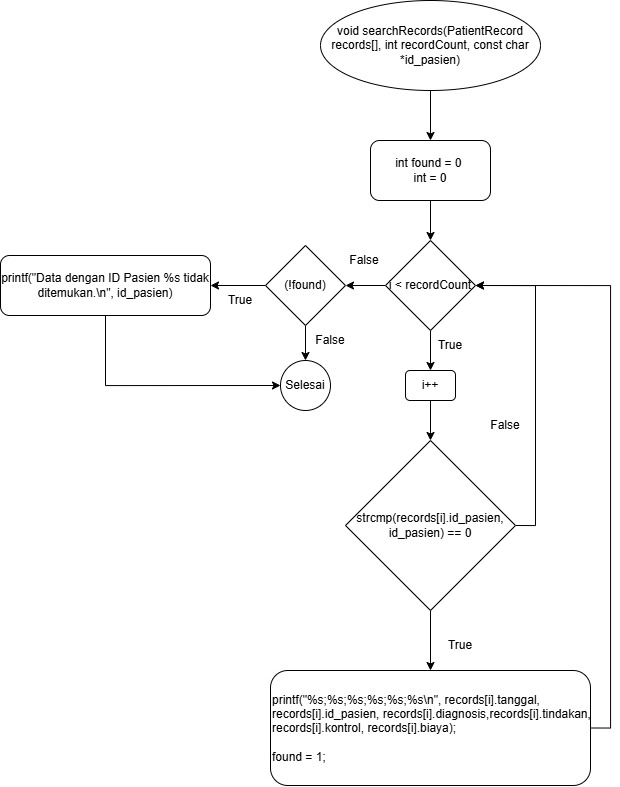


Gambar 4-A-3-b Flowchart Fungsi “ubahRiwayatMedis”

#### Fitur Hapus Riwayat Keterangan, Diagnosis, dan Tindakan



#### Fitur Cari Riwayat Keterangan, Diagnosis, dan Tindakan



### Laporan Keuangan, Analisis Penyakit, dan Informasi Kontrol Pasien

#### Fitur Laporan Keuangan (Bulanan, Tahunan, Rata-rata Tahun

#### Fitur Analisis Penyakit Pasien

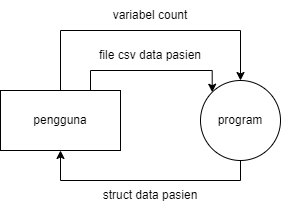
#### Fitur Informasi Kontrol Pasien

1. *Data Flow Diagram (DFD)*

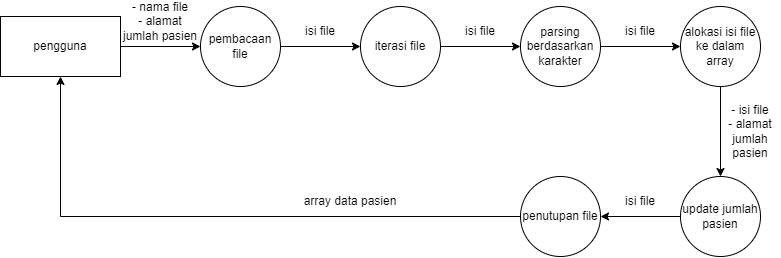
### Program Utama dan Beberapa Fungsi Penunjang

#### Program Utama

#### Fitur Baca Data Pasien

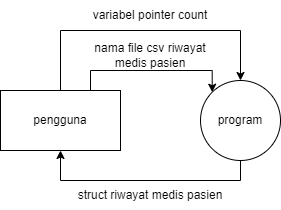


Gambar 4-B-1-b DFD Level 0 Fitur Baca Data Pasien



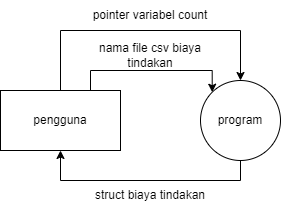
Gambar DFD Level 1 Fitur Baca Data Pasien

#### Fitur Baca Riwayat Medis Pasien



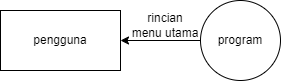
Gambar 4-B-1-c DFD Level 0 Fitur Baca Riwayat Medis Pasien

#### Fitur Baca Biaya Tindakan



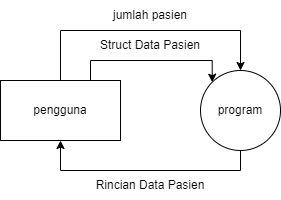
Gambar 4-B-1-d DFD Level 0 Fitur Baca Biaya Tindakan

#### Fitur Tampilkan Menu Utama



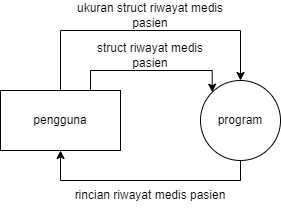
Gambar 4-B-1-e DFD Fitur Tampilkan Menu Utama

#### Fitur Tampilkan Data Pasien



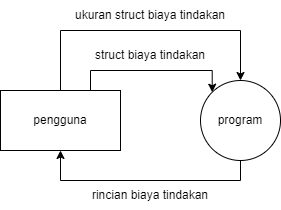
Gambar 4-B-1-f DFD Level 0 Fitur Tampilkan Data Pasien

#### Fitur Tampilkan Riwayat Medis Pasien



Gambar 4-B-1-g DFD Level 0 Fitur Tampilkan Riwayat Medis Pasien

#### Fitur Tampilkan Biaya Tindakan



Gambar 4-B-1-h DFD Level 0 Fitur Tampilkan Biaya Tindakan

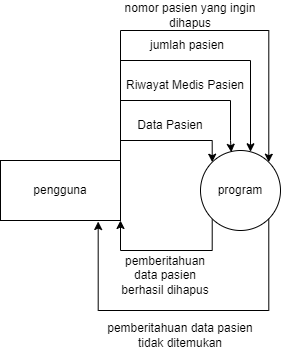
#### Fitur Pembersihan Buffer Input

### Manajemen Data Pasien

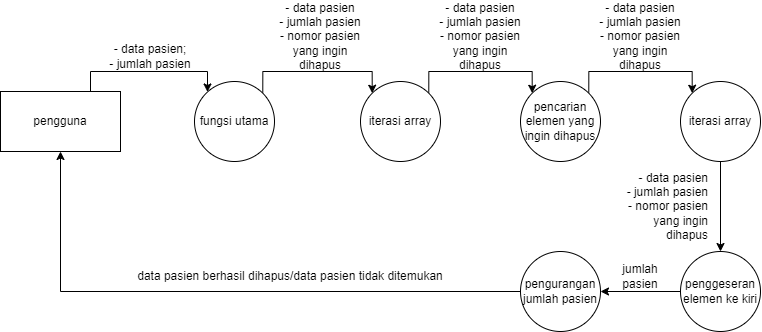
#### Fitur Tambah Data Pasien

#### Fitur Ubah Data Pasien

#### Fitur Hapus Data Pasien

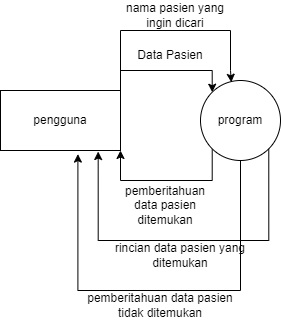


Gambar 4-B-1-c DFD Level 0 Fitur Hapus Data Pasien

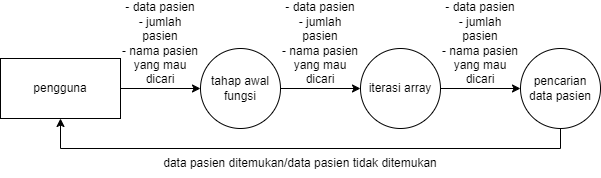


Gambar DFD Level 1 Fitur Hapus Data Pasien

#### Fitur Cari Data Pasien



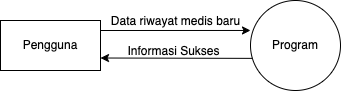
Gambar 4-B-1-d DFD Level 0 Fitur Cari Data Pasien



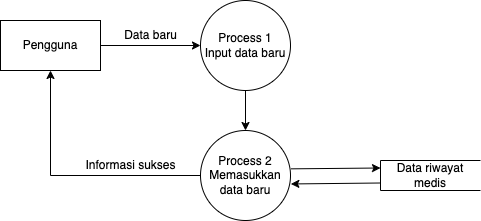
Gambar DFD Level 1 Fitur Cari Data Pasien

### Riwayat Medis dan Tindakan

#### Fitur Tambah Riwayat Keterangan, Diagnosis, dan Tindakan

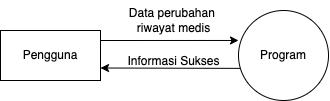


Gambar 4-B-2-a.1 DFD Level 0 Fitur tambah riwayat medis

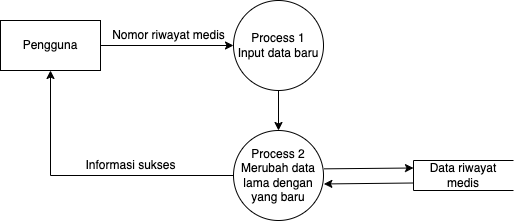


Gambar 4-B-2-a.2 DFD Level 1 Fitur tambah riwayat medis

#### Fitur Ubah Riwayat Keterangan, Diagnosis, dan Tindakan

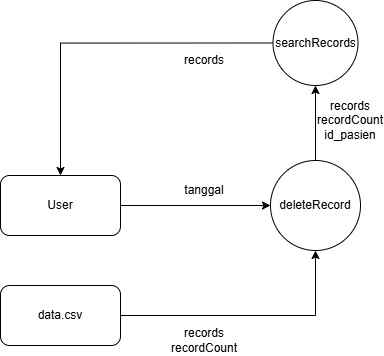


Gambar 4-B-2-b.1 DFD Level 0 Fitur ubah riwayat medis

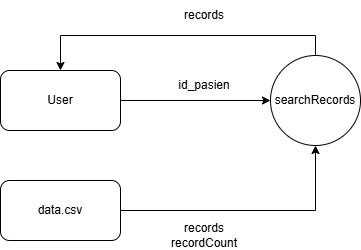


Gambar 4-B-2-b.2 DFD Level 1 Fitur ubah riwayat medis

#### Fitur Hapus Riwayat Keterangan, Diagnosis, dan Tindakan



#### Fitur Cari Riwayat Keterangan, Diagnosis, dan Tindakan



### Laporan Keuangan, Analisis Penyakit, dan Informasi Kontrol Pasien

#### Fitur Laporan Keuangan (Bulanan, Tahunan, Rata-rata Tahun

#### Fitur Analisis Penyakit Pasien

#### Fitur Informasi Kontrol Pasien

1. *Graphical User Interface (GUI)*

# Implementasi dan Analisis

### Manajemen Data Pasien

#### Fitur Tambah Data Pasien

#### Fitur Ubah Data Pasien

#### Fitur Hapus Data Pasien

Fungsi hapusDataPasien digunakan untuk menghapus data pasien dari array dataPasien dan riwayatMedisPasien berdasarkan nomor pasien yang dimasukkan oleh pengguna. Berikut adalah penjelasan implementasi fungsi tersebut:

* 1. Pertama, fungsi meminta pengguna untuk memasukkan nomor pasien yang ingin dihapus datanya menggunakan printf dan scanf.
  2. Fungsi kemudian mencari data pasien dengan nomor yang sesuai dalam array dataPasien menggunakan perulangan for. Jika data pasien ditemukan, fungsi melakukan langkah-langkah berikut: a. Menyimpan nama lengkap pasien dalam variabel nama\_pasien menggunakan strcpy. b. Menggeser elemen-elemen setelah posisi data pasien yang dihapus ke kiri menggunakan perulangan for bersarang. Pergeseran ini dilakukan untuk mengisi kekosongan yang ditimbulkan setelah penghapusan data pasien. c. Mengurangi nilai count (jumlah elemen dalam array) dengan 1 karena satu data pasien telah dihapus. d. Menampilkan pesan konfirmasi bahwa data pasien telah dihapus menggunakan printf. e. Keluar dari fungsi menggunakan return.
  3. Jika data pasien tidak ditemukan setelah perulangan selesai, fungsi akan menampilkan pesan "Data pasien tidak ditemukan" menggunakan printf.

Fungsi ini memiliki kompleksitas waktu `O(n)`, di mana `n` adalah jumlah data pasien dalam array. Hal ini disebabkan oleh perulangan `for` yang digunakan untuk mencari data pasien yang ingin dihapus dan pergeseran elemen-elemen setelah penghapusan.

Lebih lanjut, fungsi ini memiliki kompleksitas ruang (*space complexity*) sebesar `O(1)`. Hal ini disebabkan fungsi ini tidak mengalokasikan memori secara dinamis. Hal ini didukung dengan hanya digunakannya variabel-variabel statis seperti `no\_pasien`, `i`, dan `j` bertipe integer, serta `nama\_pasien` bersifat string. Di sisi lain, dapat dilihat bahwa *array* `dataPasien` dan `riwayatMedisPasien` sudah dialokasikan sebelumnya di luar fungsi.

Secara keseluruhan, fungsi ini cukup efisien dalam menghapus data pasien dari array karena tidak perlu membuat array baru atau mengalokasikan memori tambahan.

#### Fitur Cari Data Pasien

Fungsi cariDataPasien digunakan untuk mencari data pasien dalam *array* dataPasien berdasarkan nama pasien yang dimasukkan oleh pengguna. Berikut adalah penjelasan implementasi fungsi tersebut:

1. Fungsi meminta pengguna untuk memasukkan nama pasien yang ingin dicari menggunakan printf dan fgets. Nama pasien yang dimasukkan disimpan dalam variabel nama\_pasien.
2. Fungsi memanggil fungsi clearInputBuffer untuk membersihkan buffer input sebelum membaca input dari pengguna.
3. Karakter newline (\n) pada akhir input yang dimasukkan oleh pengguna dihapus menggunakan strcspn dan assignment karakter \0.
4. Fungsi menginisialisasi pointer foundPasien dengan nilai NULL.
5. Fungsi melakukan pencarian data pasien dengan nama yang sesuai dalam array dataPasien menggunakan perulangan for.
6. Dalam perulangan, fungsi membandingkan nama pasien dalam setiap elemen array dataPasien dengan nama\_pasien yang dimasukkan oleh pengguna menggunakan strcmp.
7. Jika ditemukan kecocokan nama, pointer foundPasien akan diisi dengan alamat memori dari elemen array dataPasien yang cocok, dan perulangan akan dihentikan dengan break.
8. Setelah perulangan selesai, fungsi akan memeriksa apakah foundPasien bernilai NULL atau tidak.
9. Jika foundPasien tidak NULL, fungsi akan menampilkan data pasien yang ditemukan menggunakan printf.
10. Jika foundPasien bernilai NULL, fungsi akan menampilkan pesan "Data pasien tidak ditemukan" menggunakan printf.

Kompleksitas waktu dari fungsi cariDataPasien adalah `O(n)`, di mana `n` adalah jumlah data pasien dalam array dataPasien. Hal ini disebabkan oleh perulangan for yang digunakan untuk mencari data pasien dengan nama yang sesuai.

Di sisi lain, kompleksitas ruang dari fungsi ini adalah `O(1)`. Hal ini disebabkan tidak ada memori dinamis yang dialokasikan. Hal ini dibuktikan dengan dipakainya variabel-variabel statis, di antaranya nama\_pasien (tipe data `char` dengan ukuran sebesar lima puluh) dan foundPasien (tipe data *pointer* ke Data\_Pasien).

Ada pun kinerja dari fungsi ini dipengaruhi oleh beberapa faktor. Faktor yang pertama adalah kompleksitas waktu yang mana waktu pencarian dapat menjadi lambat untuk kasus dengan jumlah data pasien yang besar. Sejumlah operasi *string* juga menjadi faktor karena operasi-operasi ini dapat menjadi lambat jika panjang *string* nama pasien cukup besar.

Secara keseluruhan, kinerja dari fungsi cariDataPasien cukup baik untuk kasus dengan jumlah data pasien yang kecil hingga sedang. Namun, untuk kasus dengan jumlah data pasien yang besar, kinerja dapat menjadi lambat karena kompleksitas waktu yang linear.

### Riwayat Medis dan Tindakan

#### Fitur Tambah Riwayat Keterangan, Diagnosis, dan Tindakan

Fungsi tambahRiwayatMedis menambahkan data riwayat medis pada *array* riwayatMedis. Berikut adalah penjelasan implementasi fungsi tersebut:

* 1. Fungsi membuat variabel riwayatBaru dengan tipe data struct Riwayat\_Medis\_Pasien.
  2. Fungsi menerima input berupa Tanggal, ID Pasien, Diagnosis, Tindakan, dan Kontrol. Nomor riwayat medis baru berupa sizeRiwayatMedis yang ditambah 1. Input dilakukan dengan fungsi fgets.
  3. Karakter newline (\n) pada akhir input yang dimasukkan oleh pengguna dihapus menggunakan strcspn dan assignment karakter \0.
  4. Fungsi memanggil fungsi clearInputBuffer untuk membersihkan buffer input sebelum membaca input dari pengguna.
  5. Dilakukan iterasi pada sizeRiwayatMedis untuk menunjukkan data riwayat medis bertambah satu.
  6. Lalu, elemen terakhir pada array riwayatMedis di-*assign* dengan riwayatBaru.
  7. Ketika semua langkah telah dijalankan, fungsi akan menampilkan informasi bahwa penambahan riwayat medis telah berhasil.

Setiap baris pada fungsi ini dilakukan tepat satu kali tanpa dilakukannya perulangan. Maka dari itu, kompleksitas waktu dari fungsi ini adalah O(1). Di sisi lain, variabel yang digunakan pada fungsi ini juga merupakan variabel statis, yaitu tidak dilakukan alokasi memori. Kompleksitas ruang pada fungsi ini juga O(1).

Kinerja dari fungsi ini sudah baik karena tidak bergantung dengan variabel apa pun. Fungsi ini akan bekerja dengan waktu yang sama pada ukuran data yang besar maupun kecil.

#### Fitur Ubah Riwayat Keterangan, Diagnosis, dan Tindakan

Fungsi tambahRiwayatMedis menambahkan data riwayat medis pada *array* riwayatMedis. Berikut adalah penjelasan implementasi fungsi tersebut:

* 1. Fungsi membuat variabel perubahan dengan tipe data struct Riwayat\_Medis\_Pasien.
  2. Fungsi menerima input nomor riwayat medis dari pengguna.
  3. Jika nomor tidak berada pada data, maka fungsi akan mencetak error, lalu fungsi selesai.
  4. Dilakukan assignment pada perubahan.No sesuai input nomor pasien
  5. Fungsi menerima input berupa Tanggal, ID Pasien, Diagnosis, Tindakan, dan Kontrol. Input dilakukan dengan fungsi fgets. Input tersebut disimpan pada variabel perubahan.
  6. Karakter newline (\n) pada akhir input yang dimasukkan oleh pengguna dihapus menggunakan strcspn dan assignment karakter \0.
  7. Fungsi memanggil fungsi clearInputBuffer untuk membersihkan buffer input sebelum membaca input dari pengguna.
  8. Elemen ke no-1 pada *array* riwayatMedis di-*assign* dengan variabel perubahan.
  9. Ketika semua langkah telah dijalankan, fungsi akan menampilkan informasi bahwa perubahan riwayat medis telah berhasil.

Kompleksitas waktu pada fungsi ini adalah O(1) karena setiap baris dijalankan tepat satu kali tanpa adanya perulangan. Kompleksitas ruang pada fungsi ini adalah O(1) karena tidak ada alokasi memori pada variabel manapun. Kinerja fungsi ini sudah baik karena waktu ataupun memori yang digunakan tidak bergantung dengan variabel apapun.

#### Fitur Hapus Riwayat Keterangan, Diagnosis, dan Tindakan

#### Fitur Cari Riwayat Keterangan, Diagnosis, dan Tindakan

### Laporan Keuangan, Analisis Penyakit, dan Informasi Kontrol Pasien

#### Fitur Laporan Keuangan (Bulanan, Tahunan, Rata-rata Tahun

#### Fitur Analisis Penyakit Pasien

#### Fitur Informasi Kontrol Pasien

# Kesimpulan

Pembuatan aplikasi pencatatan pasien dilakukan untuk memudahkan klinik X dalam perekaman pasien-pasien yang datang. Aplikasi ini dimulai dengan membuka tiga file csv yang sudah berisi data pasien, data riwayat pasien, dan data biaya tindakan. Pembacaan file disimpan dalam tiga array berbeda: array data pasien, array riwayat medis, dan array biaya tindakan. Pengelolaan data seperti penambahan, pencarian, penghapusan, dan pemberian informasi dilakukan pada ketiga array tersebut.

Setelah membuat aplikasi pencatatan pasien dan melakukan analisis, didapatkan kesimpulan seperti berikut.

1. Pengelolaan data seperti pencatatan, pengubahan, pencarian, dan penghapusan data pasien tidak perlu lagi dilakukan pada sebuah buku karena aplikasi ini sudah memuat semua fitur yang dibutuhkan
2. Masih terdapat kemungkinan human error pada aplikasi ini karena proses pencatatan data masih dilakukan oleh pengguna. Namun, human error dapat terminimalisasikan karena tampilan dari aplikasi yang lebih mudah dibaca dibandingkan dengan buku.
3. Pengaksesan data dilakukan dengan memilih fitur yang diinginkan, lalu memilih data yang ingin diakses. Hal ini membuat pengaksesan data menjadi jauh lebih cepat.
4. Fitur analisis data meliputi laporan keuangan klinik, laporan jumlah pasien dan penyakitnya, serta informasi mengenai pasien yang perlu kembali kontrol sehingga klinik dapat membuat keputusan yang tepat.
5. Seluruh data akan tersimpan pada aplikasi ini selama aplikasi tidak ditutup.

# Referensi

1. G. Eason, B. Noble, and I.N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.
3. I.S. Jacobs and C.P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

Author 1 short CV and phorograph

Author 2 short CV and phorograph