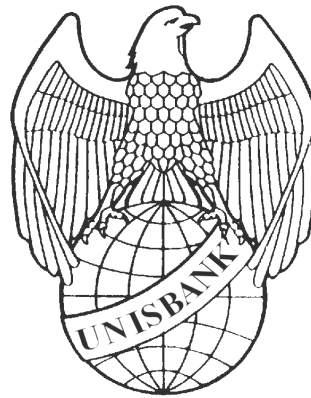


**RANCANG BANGUN SISTEM INFORMASI PENYAKIT KELAMIN
DENGAN MENGGUNAKAN JAVA**

Tugas Akhir disusun untuk memenuhi syarat
mencapai gelar Kesarjanaan Komputer pada
Program Studi Sistem Informasi
Jenjang Program Strata – 1



Oleh :

IKHFAN SUBANDI

06.01.55.0152

5769

**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS STIKUBANK (UNISBANK)**

SEMARANG

2010

PERNYATAAN KESIAPAN UJIAN TUGAS AKHIR

Saya, Ikhfan Subandi, dengan ini menyatakan bahwa Laporan Tugas Akhir yang berjudul :

RANCANG BANGUN SISTEM INFORMASI PENYAKIT KELAMIN DENGAN MENGGUNAKAN JAVA

adalah benar hasil karya saya dan belum pernah diajukan sebagai karya ilmiah, sebagian atau seluruhnya, atas nama saya atau pihak lain.

(Ikhfan Subandi)

NIM : 06.01.55.0152

Disetujui oleh Pembimbing

Kami setuju Laporan tersebut diajukan untuk Ujian Tugas Akhir

(R.Soelistijadi, S.Sos, M.Kom)

Pembimbing I

Semarang.....2010

(Ir.Zuly Budiarmo M,Cs)

Pembimbing II

Semarang.....2010

**RANCANG BANGUN SISTEM INFORMASI PENYAKIT
KELAMIN DENGAN MENGGUNAKAN JAVA**

HALAMAN PENGESAHAN

Telah dipertahankan di depan tim dosen penguji Tugas Akhir Fakultas Teknologi Informasi UNIVERSITAS STIKUBANK (UNISBANK) Semarang dan diterima sebagai salah satu syarat guna menyelesaikan jenjang Program Strata 1, Program Studi : Sistem Informasi

Semarang, September

2010

Ketua

(R.Soelistijadi, S.Sos,M.Kom)
Sekretaris

(Ir.Zuly Budiarmo, M.Cs)
Anggota

(Dwi Agus Diartono, S.Kom, M.Kom)

MENGETAHUI:
UNIVERSITAS STIKUBANK (UNISBANK) SEMARANG

Fakultas Teknologi Informasi
Dekan

(Dwi Agus Diartono, S.Kom, M.Kom)

MOTTO dan PERSEMBAHAN

Motto

- ✓ *Bahaya yang lebih besar bagi kebanyakan dari kita bukanlah gagal meraih tujuan yang terlalu tinggi, melainkan berhasil mencapai tujuan yang terlalu rendah.*
- ✓ *Semangat adalah salah satu mesin terkuat kesuksesan. Ketika kau melakukan sesuatu, lakukan dengan sekuat tenaga. Curahkan seluruh jiwamu. Tandai dengan kepribadianmu sendiri. Jadilah aktif, jadilah enerjik, jadilah bersemangat dan setia, dan kau akan mencapai tujuanmu. Tak ada hal besar yang dicapai tanpa semangat.*
- ✓ *I find that the harder I work, the more luck I seem to have.*
- ✓ *Kegagalan adalah sukses yang tertunda*

Persembahan

- ✓ *Allah S.W.T.yang telah memberikan kekuatan agar penulis bisa menyelesaikan Skripsi ini*
- ✓ *Nabi Muhammad SAW sebagai nabi junjunganku dan suri tauladan bagi seluruh umat Islam di dunia*
- ✓ *Kedua orang tua, kakak dan juga seluruh keluarga penulis*
- ✓ *Calon pendamping hidupku ukhti Kustina (uus)*
- ✓ *Sahabat-sahabat di Universitas Stikubank (Febry, Afrill, Tri, Agung, arya, Bambang, Krisna, Ghoval ,anggoro, cahya, arista, andika, dewi,azrul dan juga teman-teman lainnya yang tidak dapat disebutkan satu persatu).*
- ✓ *Serta semua pihak yang telah banyak memberikan bantuannya dalam penyusunan Tugas Akhir ini.*

UNIVERSITAS STIKUBANK SEMARANG
FAKULTAS TEKNOLOGI INFORMASI
PROGRAM STUDI SISTEM INFORMASI

Tugas Akhir Sarjana Komputer
Semester 8 tahun 2010

RANCANG BANGUN SISTEM INFORMASI PENYAKIT KELAMIN
DENGAN MENGGUNAKAN JAVA

Ikhfan Subandi
06.01.55.0152

Abstraksi

Pada saat ini penyakit kelamin semakin populer di masyarakat Indonesia. Penyakit kelamin adalah jenis penyakit yang disebabkan oleh kuman. Dan biasanya ditularkan melalui hubungan seksual. Penyakit kelamin kebanyakan sudah bisa disembuhkan namun masih ada juga penyakit kelamin yang belum bisa disembuhkan.

Dalam membangun sistem aplikasi ini permasalahan yang muncul adalah bagaimana mengembangkan suatu aplikasi yang dapat menyajikan informasi penyakit kelamin, sehingga dapat membantu masyarakat untuk mempelajari penyakit kelamin.

Tujuan dari penelitian ini adalah menciptakan suatu aplikasi yang dapat memberikan informasi mengenai penyakit kelamin kepada user melalui media handphone.

Alat-alat analisis yang digunakan dalam membangun aplikasi ini diantaranya, flowchart (diagram alur) yang digunakan untuk menerangkan logika program dan langkah-langkah dari proses Bahasa pemrograman yang digunakan adalah menggunakan Java Platform 2 Micro Edition (J2ME).

Hasil penelitian ini adalah suatu aplikasi yang memberikan sistem informasi menggunakan bahasa pemrograman Java Platform 2 Micro Edition (J2ME). yang dapat menyajikan informasi mengenai berbagai macam penyakit kelamin.

Kesimpulan penelitian ini adalah menghasilkan aplikasi yang dapat memberikan pengetahuan kepada user mengenai penyakit kelamin. Saran yang diberikan untuk pengembangan aplikasi ini di masa mendatang adalah dimungkinkannya pengembangan Sistem Informasi yang berbasis web, sehingga dapat mempermudah user untuk mengaksesnya dari berbagai tempat.

Kata kunci: Sistem Informasi, penyakit kelamin.

Mengetahui,

Pembimbing I

Pembimbing II

(R.Soelistijadi, S.Sos, M.Kom)

(Ir.Zuly Budiarmo, M.Cs)

KATA PENGANTAR

Puji syukur penulis limpahkan hanya kepada Allah SWT yang telah memberikan kemampuan, sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul “Rancang Bangun Sistem informasi Penyakit Kelamin dengan Java” ini dengan baik, sebagai salah satu persyaratan untuk memperoleh gelar sarjana pada program studi Strata Satu (S-1) Sistem Informasi, di Universitas Stikubank (UNISBANK) Semarang.

Selama penulisan Tugas Akhir ini, penulis mendapatkan banyak dorongan yang membantu penulis mengatasi berbagai hambatan yang didapat. Oleh karena itu, penulis mengucapkan banyak terima kasih kepada pihak-pihak yang telah membantu kelancaran penulis dalam menyusun Tugas Akhir ini, yaitu:

1. Bapak DR. Bambang Suko Priyono, M.M selaku Rektor Universitas Stikubank (UNISBANK) Semarang.
2. Bapak Kristophorus Hadiono, S.Kom, M.Cs selaku Ketua Program Studi Sistem Informasi yang telah membantu penulis dalam memberikan saran dan konsultasi dalam menentukan judul Tugas Akhir, sehingga skripsi ini dapat terselesaikan dengan baik.
3. Bapak R.Soelistijadi, S.sos M.Kom selaku pembimbing satu yang telah meluangkan banyak waktu untuk membimbing dan memberikan saran bagi penulis dalam proses penyusunan Tugas Akhir.
4. Bapak Ir.Zuly Budiarmo, M.Cs selaku pembimbing dua yang telah membimbing dan memeriksa laporan Tugas Akhir yang disusun oleh penulis.

5. Seluruh dosen Fakultas Teknologi Informasi yang telah mengajarkan ilmunya kepada penulis.
6. Kedua orang tua, kakak dan seluruh keluarga yang telah banyak memberikan saran, dukungan dan doa kepada penulis selama penyusunan Tugas Akhir.
7. Teman-teman di UNISBANK yaitu Agung, Febry, Afrilla, Arya, Krisna, Tri, Ghoval dan juga teman-teman lainnya, yang telah banyak membantu dalam memberikan inspirasi, dukungan, doa dan berbagi ilmu kepada penulis. Sehingga penulisan Tugas Akhir ini dapat terselesaikan dengan baik.
8. Tak terkecuali semua pihak yang telah banyak memberikan bantuan baik moril maupun materiil dalam penyusunan Tugas Akhir ini.

Penulis menyadari bahwa pembuatan Tugas Akhir ini masih jauh dari sempurna, karena itu penulis mengharapkan kritik dan saran yang bersifat membangun dari pembaca demi kesempurnaan laporan Tugas Akhir ini di masa mendatang. Akhir kata, semoga laporan Tugas Akhir ini dapat bermanfaat bagi semua pihak. Terima Kasih.

Semarang, Juni 2010

Penulis

DAFTAR ISI

	Halaman
Halaman Judul	i
Halaman Persetujuan	ii
Halaman Pengesahan	iii
Halaman Motto/Persembahan	iv
Abstrak	v
Kata Pengantar	vi
Daftar Isi	viii
Daftar Gambar	xii
Daftar Tabel	xiii
BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan dan Batasan Masalah	3
1.2.1. Rumusan Masalah.....	3
1.2.2. Batasan Masalah	3
1.3. Tujuan dan Manfaat Penelitian.....	4
1.3.1. Tujuan Penelitian	4
1.3.2. Manfaat Penelitian	4
1.4. Metodologi Penelitian	5
1.4.1. Obyek Penelitian	5

1.4.2. Jenis dan sumber Data	5
1.4.3. Metode Pengumpulan Data	6
1.4.4. Metode Pengembangan Sistem.....	6
1.6. Sistematika Penulisan	8

BAB II LANDASAN TEORI

2.1. Pengertian Sistem	10
2.2. Pengertian Informasi.....	10
2.3. Pengertian Sistem Informasi	12
2.4. Pengertian Penyakit Kelamin.....	15
2.4.1. Urethritis Gonore.....	15
2.4.2. Tysonitis	17
2.4.3. Paraurethritis	17
2.4.4. Cowperitis	18
2.5. Diagram Arus Data(DFD).....	18
2.6. Entity Relationship (E-R Diagram).....	20
2.7. Flowchart	23
2.8. Java Platform 2 Micro Editon(J2ME)	25
2.8.1. Pengertian Java	25
2.8.2. Kelebihan Java.....	27
2.8.3. Karakteristik Java	28
2.8.4. Pengertian Platform Java.....	31
2.8.5. Arsitektur J2ME	32
2.9. Perangkat Mobile	34

BAB III ANALISA DAN PERANCANGAN SISTEM

3.1. Analisa Permasalahan	36
3.2. Deskripsi Sistem Yang Akan Dibuat	41
3.3. Analisa Kebutuhan Sistem	42
3.3.1 Kebutuhan Hardware	42
3.3.2 Kebutuhan Software.....	42
3.3.3 Kebutuhan Brainware	43
3.4. Perancangan Sistem	43
3.5. Flowchart.....	44
3.6. Desain Sistem.....	46
3.6.1 Data Flow Diagram (DFD).....	46
3.7. Struktur Program	49
3.8. Perancangan Halaman Input dan Output	49
3.8.1. Halaman Pembuka	49
3.8.2. Halaman Menu Utama	50
3.8.3. Halaman Menu Penyakit Kelamin Pada Pria	51
3.8.4. Halaman Submenu Penyakit Kelamin Pada Pria	52
3.8.5. Halaman Menu Penyakit Kelamin Pada Wanita.....	53
3.8.6. Halaman Submenu Penyakit Kelamin Pada wanita	54

BAB IV IMPLEMENTASI SISTEM

4.1. Implementasi Sistem	55
4.1.1. Impementasi Hardware	55
4.1.2. Implementasi Software.....	56
4.1.3. Pemllihan Brainware	57
4.2. Pengujian Sistem	58
4.2.1. Implementasi Tampilan Pembuka	58
4.2.2. Implementasi Tampilan Menu Utama.....	59
4.2.3. Implementasi Menu Penyakit Kelamin Pada Pria	60
4.2.4. Implementasi Submenu Penyakit Kelamin Pada Pria.....	61
4.2.5. Implementasi Menu Penyakit Kelamin Pada Wanita.....	62
4.2.6. Implementasi Submenu Penyakit Kelamin Pada Wanita.....	63

BAB V PENUTUP

5.1. Kesimpulan	64
5.2. Saran	64

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

	Halaman
Gambar 2.1.	Siklus Pengolahan Data 13
Gambar 2.2.	Lima Komponen Sistem Informasi..... 13
Gambar 2.3.	Keunggulan –Keunggulan Java 26
Gambar 2.4.	Platform Java 31
Gambar 3.1.	Flowchart Menu Utama 44
Gambar 3.2.	Data Context Diagram 46
Gambar 3.3.	DFD Level 1 47
Gambar 3.4.	E-R Diagram 48
Gambar 3.5.	Struktur Program Sistem Informasi Penyakit Kelamin ... 49
Gambar 3.6.	Perancangan Halaman Pembuka 50
Gambar 3.7.	Perancangan Menu Utama 50
Gambar 3.8.	Perancangan Menu Penyakit Kelamin Pada Pria 51
Gambar 3.9.	Perancangan Submenu Penyakit Kelamin Pada Pria 52
Gambar 3.10.	Perancangan Menu Penyakit Kelamin Pada Wanita 53
Gambar 3.11.	Perancangan Submenu Penyakit Kelamin Pada Wanita 54
Gambar 4.1.	Tampilan Layar Pembuka 58
Gambar 4.2.	Tampilan Menu Utama 59
Gambar 4.3.	Tampilan Menu Penyakit Kelamin Pada Pria 60
Gambar 4.4.	Tampilan Submenu Penyakit Kelamin Pada Pria..... 61
Gambar 4.5.	Tampilan Menu Penyakit Kelamin Pada Wanita 62
Gambar 4.6.	Tampilan Submenu Penyakit Kelamin Pada Wanita..... 63

DAFTAR TABEL

	Halaman
Tabel 2.1. Simbol DFD	19
Tabel 2.2. Simbol ER Diagram.....	21
Tabel 2.3. Simbol Flowchart	25
Tabel 3.1. Data Jenis Penyakit Kelamin	36
Tabel 4.1. Spesifikasi Minimal Hardware	56

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada saat ini penyakit kelamin semakin populer di masyarakat Indonesia. Penyakit kelamin adalah jenis penyakit yang disebabkan oleh kuman. Dan biasanya ditularkan melalui hubungan seksual. Penyakit kelamin kebanyakan sudah bisa disembuhkan namun masih ada juga penyakit kelamin yang belum bisa disembuhkan

Penyakit kelamin ini disebabkan bakteri. Biasanya terjadi beberapa hari hingga beberapa minggu setelah berhubungan dengan orang yang sudah terjangkit penyakit ini. Jika terjadi pada pria, penyakit ini mengeluarkan cairan dari kemaluan. Namun kadang gejala penyakit ini bias terasa berat atau bahkan tidak terasa sama sekali. sedang pada wanita gejala ini biasa tidak nampak atau hanya sangat ringan. Namun jika tidak diobati bisa membuat si wanita menjadi mandul.

Penyakit kelamin bisa disebabkan oleh Infeksi Jamur Namanya juga infeksi jamur jadi penyebabnya tentu saja jamur. Jika terjadi pada pria, gejalanya adalah adanya rasa gatal dan berwarna merah pada bawah kulit kemaluan pria yang tidak disunat. Sedang jika terjadi pada wanita, biasanya penyakit ini akan mengeluarkan cairan putih kental yang membuat gatal. Penyakit ini bisa disembuhkan dengan menggunakan krim anti jamur

Namun dalam era modern seperti sekarang ini masyarakat memiliki pengetahuan yang terbatas bahkan cenderung kurang mengetahui tentang perkembangan penyakit kelamin. Sehingga masyarakat pada umumnya kurang mengetahui tipe dan jenis penyakit kelamin bagaimana cara mengantisipasi dan mencegah penyakit kelamin. Untuk itu dengan adanya perkembangan teknologi komputer yang pesat seperti sekarang ini, perlu dimanfaatkan untuk menyebarkan dan mendayagunakan hasil penelitian

Biasanya orang awam banyak yang menemui kesulitan dalam mengenali penyakit kelamin. Hal ini di karenakan kurangnya pengetahuan yang dimiliki tentang gejala-gejala awal suatu penyakit kelamin. Mendeteksi suatu penyakit kelamin lebih awal kemungkinan tidak berlanjut kearah yang lebih parah bahkan bisa menyebabkan kematian. Dari gejala-gejala yang di timbulkan dapat ditarik sebuah kesimpulan.

Dengan bantuan J2ME dapat mengenali gejala-gejala dan cirri-ciri yang ditimbulkan oleh penyakit kelamin. Aplikasi J2ME ini membantu menganalisis gejala awal penyakit kelamin.

Dengan adanya pemanfaatan teknologi komputer dapat menciptakan suatu aplikasi berbentuk J2ME yang dapat membantu masyarakat dalam mempelajari dan mendalami lebih jauh mengenai perkembangan penyakit kelamin.

Aplikasi J2ME ini hanya di buat untuk membantu mengenali gejala dan ciri penyakit kelamin bukan membuat sebuah alat yang bisa mendeteksi semua penyakit kelamin.

Berdasarkan penjelasan uraian di atas maka peneliti mengajukan skripsi dengan judul **“RANCANG BANGUN SISTEM INFORMASI PENYAKIT KELAMIN DENGAN MENGGUNAKAN JAVA”**.

1.2 Perumusan dan Batasan Masalah

1.2.1 Perumusan Masalah

Dari latar belakang masalah yang telah diuraikan di atas dapat diambil rumusan yang akan menjadi pembahasan penelitian ini yaitu bagaimana Rancang bangun sistem informasi penyakit kelamin dengan menggunakan JAVA.

1.2.2 Pembatasan Masalah

Untuk mengarahkan penelitian sesuai spesifikasi yang ditentukan maka diberikan batasan masalah sebagai berikut :

- 1 penyakit yang di teliti hanya penyakit kelamin yang terdapat pada alat kelamin.
- 2 Perangkat mobile yang digunakan adalah *handphone* atau PDA dimana di dalamnya terdapat fasilitas Java platform (J2ME) yang mendukung MIDP 2.0 dan menyediakan API CLDC 1.0 (minimal)
- 3 Bahasa pemrograman yang digunakan adalah menggunakan Java Platform 2 Micro Edition (J2ME).

Untuk mencapai sasaran yang sesuai dengan permasalahan yang diuraikan di atas, maka peneliti memberikan batasan permasalahan dalam pembuatan penelitian ini hanya sebatas pada pengembangan aplikasi J2ME yang dapat memberikan informasi yang berguna bagi masyarakat dalam mempelajari penyakit kelamin

1.3 Tujuan dan Manfaat Penelitian

1.3.1 Tujuan Penelitian

Tujuan dari pengembangan teknologi yang hendak dicapai antara lain sebagai berikut:

- 1 Merancang bangun sistem informasi penyakit kelamin dengan menggunakan J2ME.
- 2 Memberikan pengetahuan tentang bahayanya Penyakit Kelamin.

1.3.2 Manfaat Penelitian

Manfaat dari pengembangan teknologi ini antara lain sebagai berikut:

a. Bagi Masyarakat

1. Menambah pengetahuan masyarakat tentang Penyakit Kelamin..
2. Membantu masyarakat mengenali ciri-ciri penyakit kelamin

b. Bagi Pembaca

1. Menjadi referensi bagi pembaca untuk mengembangkan atau menciptakan teknologi baru yang sejenis ataupun yang lain.

c. Bagi Penyusun

1. Menambah wawasan mengenai penyakit kelamin.
2. Dapat menerapkan dan memanfaatkan disiplin ilmu yang dimiliki.

1.4 Metodologi Penelitian

Dalam penyusunan tugas akhir ini diperlukan informasi yang penting dan akurat sehingga dapat menunjang proses pembuatan tugas akhir. Untuk mendapatkan informasi tersebut penulis menggunakan metode sebagai berikut :

1.4.1 Objek Penelitian

Dalam penyusunan laporan tugas akhir ini yang menjadi objek penelitian adalah dari buku-buku dan literatur yang mencakup Penyakit kelamin

1.4.2 Jenis dan Sumber Data

Guna menunjang pembuatan aplikasi menentukan penyakit kelamin ini, peneliti memakai jenis data dalam perolehannya adalah sebagai berikut :

- 1 Data Primer adalah data yang diperoleh secara langsung dari objek yang diteliti baik yang digunakan melalui pengamatan, pencatatan atau penelitian
- 2 Data Sekunder adalah data yang diperoleh secara tidak langsung yang berasal dari buku-buku dan literature lainnya sebagai dasar teori serta pelengkap data primer yang meliputi pendapat dan penjelasan

yang dikemukakan oleh para ahli maupun tokoh-tokoh sehingga akan membantu dalam penyusunan tugas akhir.

Contohnya : Analisa dari buku, internet dan media lainnya.

1.4.3 Metode Pengumpulan Data

Sesuai dengan sumber data serta tujuan penyusunan tugas akhir ini, maka dalam pengumpulan data ini menggunakan beberapa metode penyimpulan data antara lain sebagai berikut :

Studi Kepustakaan

Yaitu metode pengumpulan data dengan cara membaca dan mempelajari literatur, majalah, buku yang berhubungan dengan materi yang sedang diteliti saat ini untuk mendapatkan dasar-dasar teori dari data-data yang diperoleh.

1.4.4 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah metode prototype. Prototype adalah pembuatan model dari perangkat lunak yang akan dibuat atau dikerjakan sehingga pemakai dapat mengetahui hasil yang akan di dapat (Tauri. D. Mahyuzi, 2003 : 5). Proses yang terjadi pada prototype adalah :

1. Tahapan Analisa

Tahapan dimana harus dilakukan analisis seluruh kebutuhan yang diperlukan dalam perancangan ini yang meliputi :

- a. Mempelajari tentang segala hal yang berhubungan erat dengan penyakit kelamin
- b. Mengumpulkan dan menganalisa karakter penyakit kelamin dari berbagai macam literatur, buku-buku, artikel dan segala informasi yang berhubungan dengan penyakit kelamin.

2. Tahap Desain.

Tahap ini dimaksudkan untuk melakukan pengembangan secara lebih detail, antara lain :

- a. Menyusun sistem flowchart
- b. Melakukan desain tampilan program pada mobile
- c. Mengumpulkan data-data yang akan dijadikan hasil dari pencarian.

3. Building Prototyping

Merupakan pembuatan aplikasi secara keseluruhan dari rencana pemecah masalah.

4. Evaluasi

Sasarannya adalah melakukan evaluasi terhadap program yang sudah dibuat meliputi :

- a. Menguji program yang telah dibuat.

b. Menilai sistem yang telah dibuat, apakah sudah layak dipakai atau belum.

5. Hasil

Merupakan hasil dari prototype yang telah dibuat dan telah disetujui oleh user.

1.5 Sistematika Penulisan

Dalam penyusunan laporan skripsi ini peneliti menggunakan sistematika penulisan sebagai berikut:

BAB I. PENDAHULUAN

Bab ini menjelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, metode penelitian dan sistematika penulisan.

BAB II. LANDASAN TEORI

Bab ini menjelaskan tentang teori – teori yang mendukung dan mendasari peneliti dalam melakukan penelitian dan perancangan program.

BAB III. ANALISA DAN PERANCANGAN SISTEM

Bab ini akan membahas tentang tahapan perancangan dan desain yang akan dilakukan peneliti dalam merancang aplikasi J2ME tersebut.

BAB IV. IMPLEMENTASI

Bab ini menjelaskan tentang tahap – tahap implementasi program, dan hasil penelitian serta contoh tampilan dari setiap program, serta bagaimana cara pengoperasian dari aplikasi tersebut.

BAB V. PENUTUP

Bab ini berisi tentang kesimpulan dan saran dari hasil penelitian yang telah dilakukan.

BAB II

LANDASAN TEORI

2.1 PENGERTIAN SISTEM

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu (Jogiyanto HM, 1998 :1)

Sistem dengan pendekatan prosedur, yaitu suatu urutan kegiatan yang saling berhubungan, berkumpul bersama-sama untuk mencapai tujuan tertentu. Sedangkan sistem dengan pendekatan komponen / elemen, yaitu kumpulan komponen yang saling berkaitan dan bekerja sama untuk mencapai suatu tujuan tertentu. Suatu sistem dapat terdiri dari beberapa subsistem. Subsistem-subsistem tersebut dapat pula terdiri dari beberapa subsistem yang lebih kecil.

Dari sekian banyaknya arti kata sistem, kita akan mengambil pengertian bahwa sistem adalah suatu keutuhan utuh yang terdiri dari beberapa bagian yang saling berhubungan dan berinteraksi untuk mencapai tujuan tertentu.

2.2 PENGERTIAN INFORMASI

Informasi adalah data yang diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun yang akan datang. Informasi mempunyai ciri benar atau salah, baru, tambahan dan korektif. (Gordon. B. Davis : 1985). Menurut Raymond

McLeod (1995) mendefinisikan informasi sebagai data yang telah diolah menjadi bentuk yang lebih berarti bagi penerimanya. Alat pengolahan informasi dapat meliputi elemen komputer, elemen non komputer atau kombinasinya.

Sumber informasi adalah data. Data adalah kenyataan yang menggambarkan kejadian-kejadian dan kesatuan data. Kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu. Informasi diperoleh setelah data-data mentah diproses atau diolah. Menurut *John Burch* dan *Gary Grudnitski*, agar informasi dihasilkan lebih berharga, maka informasi harus memenuhi kriteria sebagai berikut :

- Informasi harus akurat, sehingga mendukung pihak manajemen dalam mengambil keputusan.
- Informasi harus relevan, benar-benar terasa manfaatnya bagi yang membutuhkan.
- Informasi harus tepat waktu, sehingga tidak ada keterlambatan pada saat yang dibutuhkan.

Kegunaan informasi adalah untuk mengurangi ketidakpastian di dalam proses pengambilan keputusan tentang suatu keadaan. Informasi yang digunakan di dalam suatu sistem informasi umumnya digunakan untuk nenerapa kegunaan. Informasi digunakan tidak hanya oleh satu orang pihak di dalam organisasi. Nilai sebuah informasi ditentukan dari dua hal yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan

bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkan informasi tersebut.

2.3 PENGERTIAN SISTEM INFORMASI

Sistem Informasi dapat didefinisikan sebagai berikut :

- a. Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.
- b. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan atau untuk mengendalikan organisasi.
- c. Suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

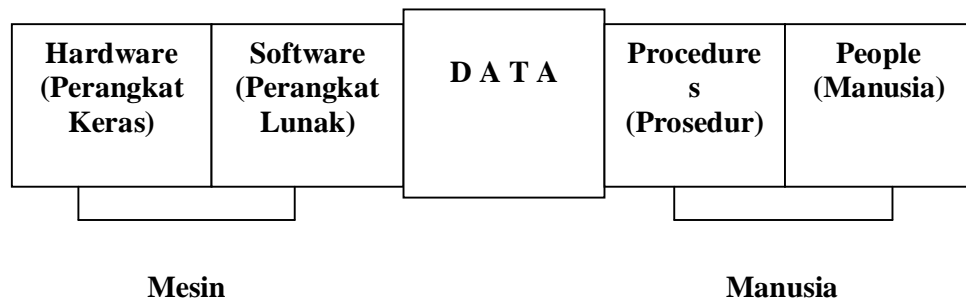
Sistem Informasi adalah kombinasi antara prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi. Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, strategi suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto H.M, 1998 : 11)

Bentuk dasar suatu sistem sangat sederhana, terdiri dari masukan, pengolahan dan keluaran.



Gambar 2.1 Siklus Pengolahan Data

Tiga tahap dasar dari siklus pengolahan data tersebut dapat dikembangkan lebih lanjut. Siklus pengolahan data yang dikembangkan (*Expanded Data Processing Cycle*) dapat ditambahkan tiga atau lebih tahapan lagi, yakni *organization*, *storage* dan *distribution*.



Gambar 2.2 Lima Komponen Sistem Informasi.

Keterangan :

1. *Hardware* ;

a. Peralatan penyimpanan data

Disk merupakan salah satu perangkat penyimpanan data yang paling sering digunakan. Disk diorganisasikan berupa silinder-silinder dengan tiap permukaan terdapat head yang ditumpuk secara

vertikal. *Head* terdiri dari beberapa *Track*. *Track* teragi menjadi sector-sektor.

b. Peralatan komunikasi data

Komunikasi data adalah satu bagian dari ilmu komunikasi yang mengkhususkan diri pada penyampaian informasi yang berupa teks dan gambar.

2. *Software*

Software merupakan kumpulan dari perintah atau fungsi yang ditulis dengan aturan tertentu unuk memerintahkan komputer melaksanakan tugas tertentu.

3. Data

Data merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi. Himpunan data akan memiliki sifat yang unik, antara lain sebagai berikut :

- a. Saling berkaitan (*inherrelated*); data-data tersebut akan saling berkaitan atau terintegrasi dan tesimpan secara terorganisir di dalam suatu media penyimpanan.
- b. Kebersamaan (*shared*); data yang terintegrasi tersebut dapat diakses oleh berbgai macam pengguna atau orang tetapi hnaya satu yang dapat merubahnya yaitu *Database Administrator (DBA)*.

4. Prosedur

Dokumentasi prosedur atau proses sistem, buku penuntun operasional (aplikasi) dan teknis. Prosedur menghubungkan berbagai perintah dan aturan yang akan menentukan rancangan dan penggunaan sistem informasi.

5. Manusia

Manusia adalah mereka yang terlibat dalam kegiatan sistem informasi seperti operator, pemimoin sistem informasi dan sebagainya. (Al-Fatta, Haniff, 2005 : 15-21)

2.4 PENGERTIAN PENYAKIT KELAMIN

Penyakit kelamin adalah jenis penyakit yang disebabkan oleh [kuman](#) yang ditularkan melalui hubungan [seks oral](#) maupun melalui [hubungan kelamin](#). (Djuanda,2004)

Macam-macam penyakit kelamin antara lain adalah sebagai berikut:

2.4.1 Uretritis Gonore

Gonore merupakan penyakit yang mempunyai insidens yang tinggi di antara P.M.S. Pada pengobatannya terjadi pula perubahan karena sebagian disebabkan oleh Neisseria gonorrhoeae yang telah resisten terhadap penisilin dan disebut Penicilinase Producing Neisseria gonorrhoeae. Kuman ini meningkat di banyak negeri termasuk Indonesia. Gonoro dalam arti luas mencakup semua penyakit yang disebabkan oleh Neisseria gonorrhoeae.

Pada umumnya penularannya melalui hubungan kelamin yaitu secara genito-genital, oro-genital, ano-genital. Tetapi, di samping itu dapat juga terjadi secara manual melalui alat-alat, pakaian, handuk, termometer, dan sebagainya. Oleh karena itu secara garis besar dikenal gonore genital, dan gonore ekstra genital

Penyebab gonoro adalah gonokok yang ditemukan oleh NEISSER pada tahun 1879 dan baru diumumkan pada tahun 1882. Kuman termasuk dalam grup neisseria dan dikenal ada 4 spesies, yaitu *N. gonorrhoeae* dan *N. meningitidis* yang bersifat patogen serta *N. catarrhalis* dan *N. pharyngis sicca* yang bersifat komensa. Keempat spesies ini sukar dibedakan kecuali dengan tes fermentasi. (Sjaiful Fahmi Daili)

Rasa gatal, panas dibagian distal uretra disekitar orifisium uretra eksternum, kemudian disusul disuria, pola kisuria keluar duh tubuh dari ujung uretra yang kadang-kadang disertai darah, dan disertai perasaan nyeri pada waktu ereksi. Pada pemeriksaan tampak orifisium uretra eksternum eritematosa, edematosa, dan ektropion. Tampak duh tubuh yang mukopurulen, dan pada beberapa kasus dapat terjadi pembesaran kelenjar getah bening inguinal unilateral atau bilateral.

2.4.2 Tysonitis

kelenjar tyson ialah kelenjar yang menghasilkan smegma. Infeksi biasanya terjadi pada penderita dengan preputium yang sangat panjang dan kebersihan yang kurang baik. Diagnosis dibuat berdasarkan ditemukannya butir pus atau pembengkakan pada daerah frenulum yang nyeri tekan. Bila duktus tertutup akan timbul abses dan merupakan sumber infeksi laten.

Rasa gatal, panas dibagian distal uretra disekitar orifisium uretra eksternum, kemudian disusul disuria, pola kisuria keluar duh tubuh dari ujung uretra yang kadang-kadang disertai darah,dan disertai perasaan nyeri pada waktu ereksi. Pada pemeriksaan tampak orifisium uretra eksternum eritematosa, edematosa,dan ektropion. Tampak duh tubuh yang mukopurulen,dan pada beberapa kasus dapat terjadi pembesaran kelenjar getah bening inguinal unilateral atau bilateral.

2.4.3 Paraurethritis

sering pada orang dengan orifisium uretra ekstemum terbuka atau hipospadia. Infeksi pada duktus ditandai dengan butir pus pada kedua muara parauretra.

Rasa gatal, panas dibagian distal uretra disekitar orifisium uretra eksternum, kemudian disusul disuria, pola kisuria keluar duh tubuh dari ujung uretra yang kadang-kadang disertai darah,dan disertai perasaan nyeri pada waktu ereksi.

2.4.4 Cowperitis

bila hanya duktus yang terkena biasanya tanpa gejala. Kalau infeksi terjadi pada kelenjar Cowper dapat terjadi abses.

Keluhan berupa nyeri dan adanya benjolan pada daerah perineum disertai rasa penuh dan panas, nyeri pada waktu defekasi, dan disuria. Jika tidak diobati abses akan pecah melalui kulit perineum, uretra, atau rektum dan mengakibatkan proktitis

2.5 Diagram Arus Data (DFD)


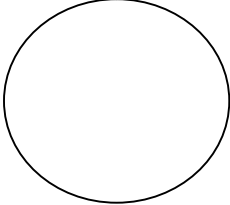

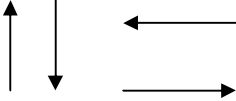
Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

Simbol-simbol dalam DFD yaitu :

Tabel 2.1 Simbol DFD

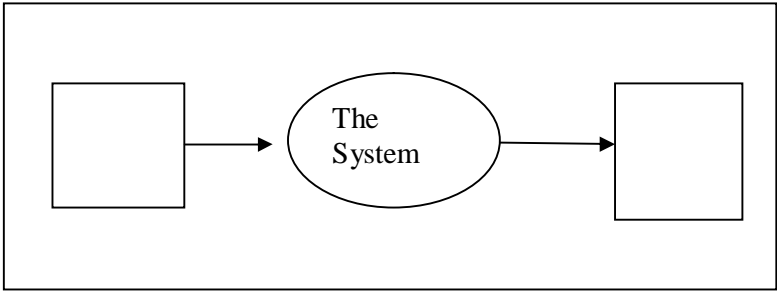
SIMBOL	KETERANGAN
	Terminator mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya terminator dikenal dengan nama entitas luar (external entity).
	Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input menjadi output.
	Data store ini biasanya berkaitan dengan penyimpanan-penyimpanan, seperti file atau database yang berkaitan dengan penyimpanan secara komputerisasi, misalnya file disket, file harddisk, file pita magnetik. Data store juga berkaitan dengan penyimpanan secara manual seperti buku alamat, file folder, dan agenda.
	Data flow / alur data digambarkan dengan anak panah, yang menunjukkan arah menuju ke dan keluar dari suatu proses. Alur data ini digunakan untuk menerangkan perpindahan data atau paket data/informasi dari satu bagian sistem

	ke bagian lainnya.
--	--------------------

Pelevelan pada DFD

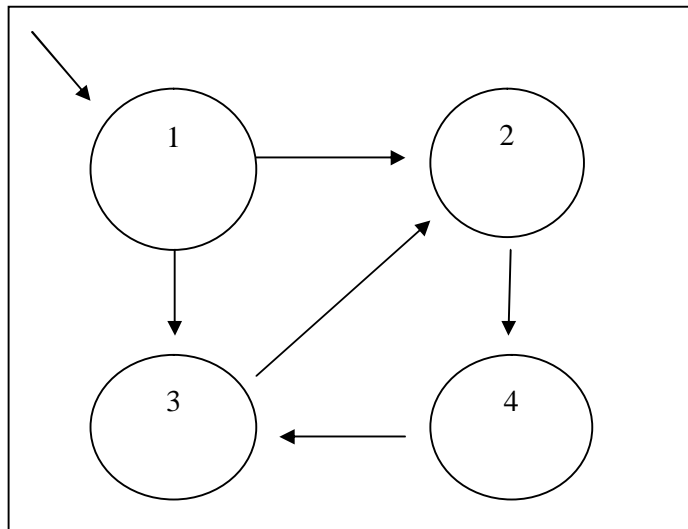
1. Context Diagram / DFD Level 0

Context Diagram (CD) merupakan data arus awal , juga disebut DFD level 0. CD ini memiliki satu Externa entity yaitu user dan admin dan sebuah proses yaitu sistem informasi penyakit penyakit kelamin.



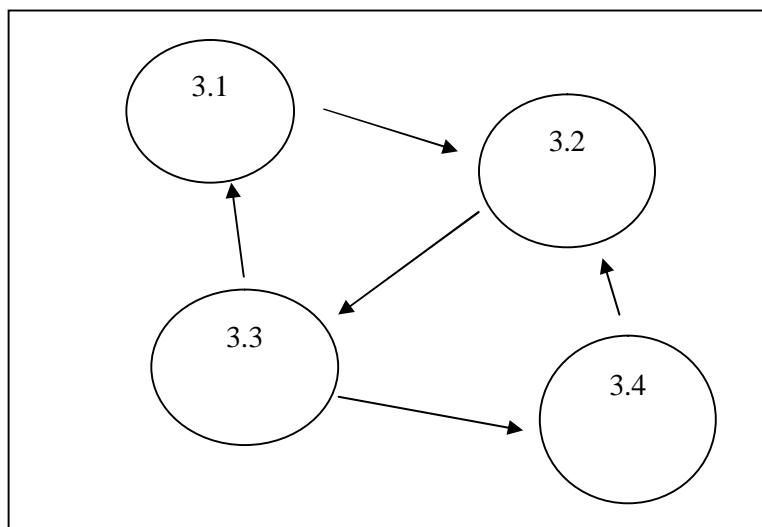
2.DFD level 1

DFD level 1 ini merupakan penjabaran darri proses CD . pada DFD level 1 ini mempunyai dua proses yaitu proses penelusuran dan proses manipulasi data.



3.DFD level 2

DFD level 2 ini merupakan penjabaran dari proses DFD level 1 pada DFD level 2 ini merupakan proses penjabaran dari proses penelusuran dan proses manipulasi data.




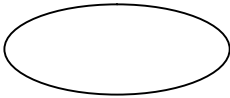
2.6 Entity Relationship (E-R Diagram)

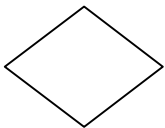

E-R Diagram adalah model konseptual hubungan antar penyimpanan. E-R Diagram digunakan untuk memodelkan struktur data dan hubungan antar data. Dengan E-R Diagram kita dapat menguji model dengan menyampaikan proses yang harus dilakukan.

E-R Diagram pertama kali dideskripsikan oleh Peter Chen (The Entity Relationship Model Toward a Unified of Data, March 1976). Dalam buku ini chen mencoba merumuskan dasar-dasar model ERD menggunakan struktur dan hubungan antar data. Ada empat simbol yang digunakan antara lain:

Simbol-simbol dalam ERD yaitu :

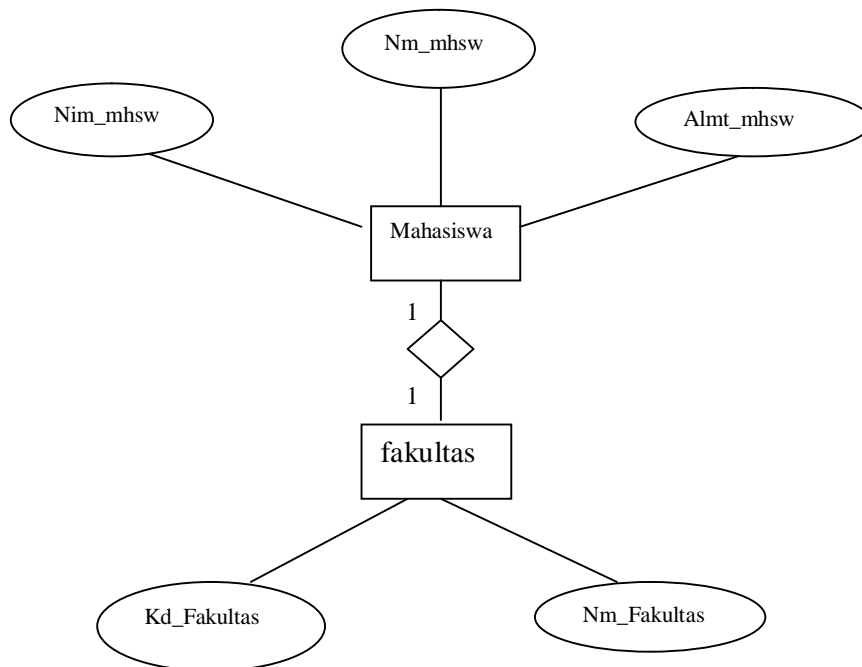
Tabel 2.2 Simbol ERD

SIMBOL	KETERANGAN
	Entity adalah suatu obyek yang dapat didefinisikan dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang dibuat.
	Atribut berfungsi mendeskripsikan karakter entity

	Simbol yang dapat menghubungkan entity satu dengan yang lain
	Garis alir menunjukkan keterkaitan antar entitas

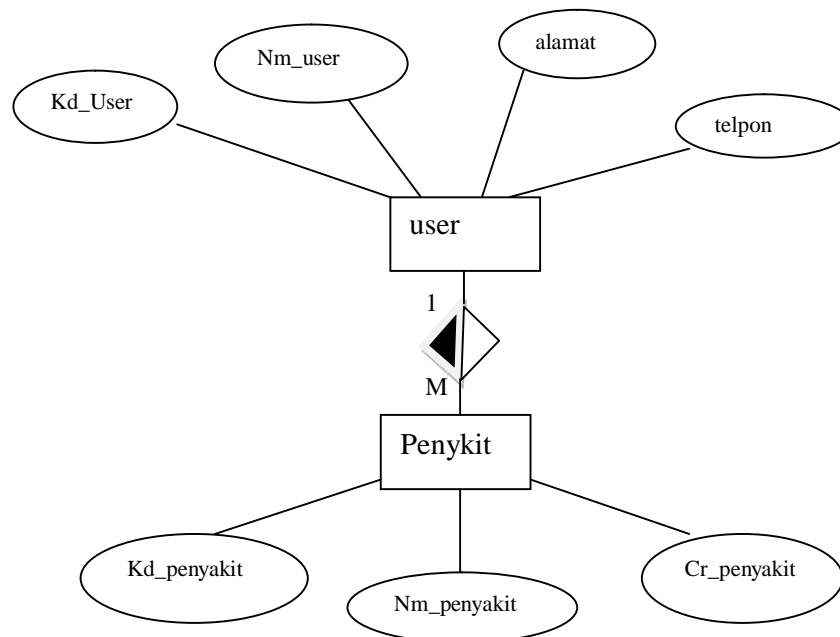
1. ER-D 1 to 1 (One to One)

Satu entitas pada tipe entitas A berhubungan dengan paling banyak satu entitas pada tipe entitas B dan juga sebaliknya.



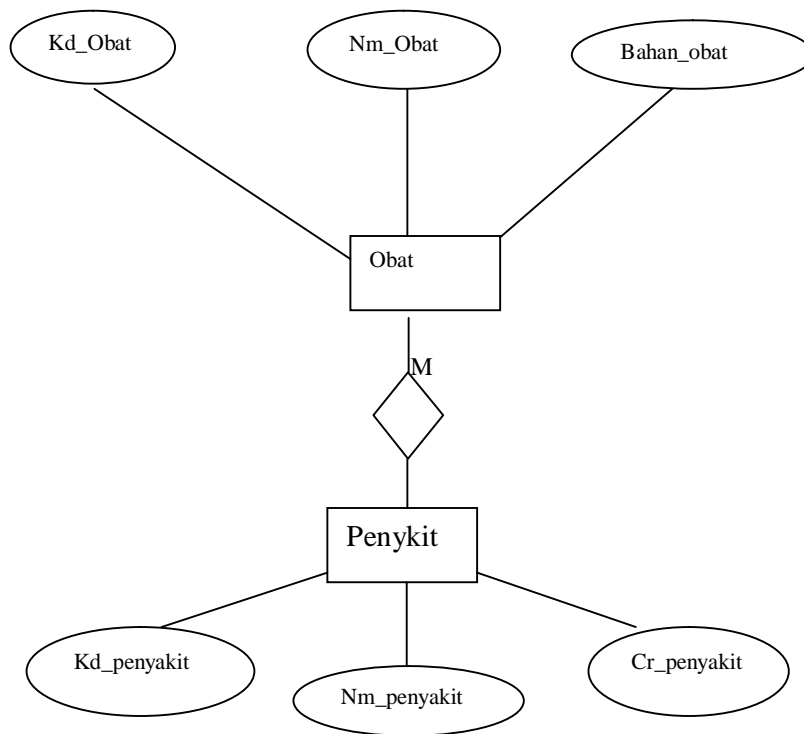
2. ER-D 1 to M (One to Many)

Suatu entitas di A dihubungkan dengan sejumlah entitas di B.



3. ER-D M to M (Many to Many)

Setiap entitas A dapat berhubungan dengan banyak entitas B dan sebaliknya setiap entitas B juga dapat berhubungan dengan banyak entitas A.



Transformasi ER-D Ke Tabel

Tahap-Tahap Transformasi :

1. Entity-Relationship Diagram menjadi basis data.
2. Entity menjadi tabel dan atribut menjadi kolom/field dari tabel.
3. Entitas lemah \diamond key dari "owner" (entitas kuat) ke tabel entitas lemah.
4. Setiap tipe entity dibuat suatu tabel yang memuat semua atribut simple, sedangkan untuk atribut komposit hanya dimuat komponen-komponennya saja.
5. Setiap tabel yang mempunyai atribut multivalued, buatlah tabel baru dimana primary key-nya merupakan gabungan dari primary key dari tabel tersebut dengan atribut multivalued.
6. Setiap unary relationship 1:N, selain membuat tabel baru berdasarkan entity, buat juga tabel baru berdasarkan relationship-nya dengan atribut kunci tamu

(foreign key) berdasarkan atribut kunci dari entity tersebut dan atribut kunci alternatif sebagai primary key-nya.

7. Untuk CR 1:1 dengan atau tanpa total participation maka akan dibuat tabel baru berdasarkan relationship, dimana kolom-kolomnya terdiri dari alternate key, dan primary key dari masing-masing entity.
8. Untuk CR 1:N dengan atau tanpa total participation maka primary key dari sisi 1 masuk ke sisi N.
9. Untuk CR M:N → dibuat tabel tersendiri berdasarkan relationshipnya dengan kolom-kolomnya terdiri dari alternate key dan primary key dari masing-masing entity.

([http://tkj.polnep.ac.id/courses/BASISDATA2cabe/document/Materi/Teori/3 - Entity Relationship Model.ppt?cidReq=BASISDATA2fc00](http://tkj.polnep.ac.id/courses/BASISDATA2cabe/document/Materi/Teori/3_-_Entity_Relationship_Model.ppt?cidReq=BASISDATA2fc00)) diakses September 2010.



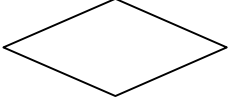


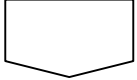

2.7 FLOWCHART

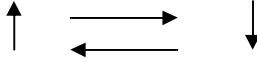


Flowchart adalah bagian yang menggambarkan urutan instruksi untuk proses dengan komputer dan hubungan antara proses lainnya dengan menggunakan symbol-simbol flowchart. Terdapat dua macam flowchart :

1. Sistem flowchart : bagan yang memperlihatkan urutan prosedur dan proses dari beberapa *file* di dalam media tertentu.
2. Program flowchart : bagan yang memperlihatkan urutan dan hubungan proses dalam suatu program. Flowchart merupakan

langkah awal dalam pembuatan program, dengan adanya flowchart maka urutan proses di proram menjadi lebih jelas.

Simbol-simbol dalam flowchart yaitu :

Simbol	Keterangan
	Menunjukkan proses pengolahan dan perubahan harga.
	<i>Manual operation</i> , suatu symbol yang menunjukkan setiap pengolahan yang tidak dilakukan oleh komputer.
	<i>Decission</i> , suatu kondisi yang akan menghasilkan kemungkinan jawaban atau aksi.
	<i>Prefined process</i> , suatu symbol untuk menyediakan tempat pengolahan dalam storage.
	<i>Terminal</i> , untuk memulai atau mengakhiri suatu program atau juga sebagai interupsi dalam program.
	Sambungan lain halaman
	Sambungan dalam satu halaman

	Arah arus
	Input-input setiap peralatan (I/O) dapat menggunakan symbol ini tanpa harus mengetahui jenis peralatan baik input atau output.
	Simbol dokumen, untuk data yang berbentuk kertas atau informasi

Tabel 2.3 Simbol – symbol flowchart

2.8 JAVA PLATFORM 2 MICRO EDITION (J2ME)

2.8.1 Pengertian java

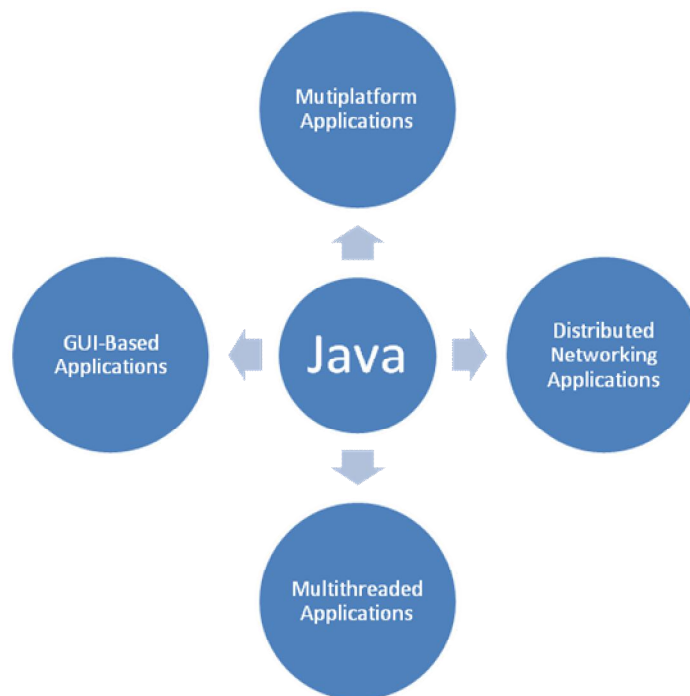
Java diciptakan oleh suatu tim yang dipimpin oleh *Patrick Naughton* dan *James Gosling* dalam suatu proyek dari *Sun Microsystem* yang memiliki kode *Green* dengan tujuan untuk menghasilkan bahasa komputer sederhana yang dapat dijalankan di peralatan sederhana dengan tidak terikat pada arsitektur tertentu. (Isak Rikcyanto, 2003 : 2)

Akhirnya *Sun* merintis versi awal Java secara resmi pada awal 1996 yang kemudian terus berkembang hingga muncul JDK 1.1 kemudian JDK 1.2 yang mengandung banyak peningkatan dan perbaikan sehingga mulai versi ini Java disebut Java2. Perubahan yang utama adalah adanya *Swing* yang merupakan teknologi GUI (*Graphical User Interface*) yang mampu menghasilkan aplikasi window yang benar-benar portabel.

Pada tahun-tahun berikutnya (1998-1999) lahirlah teknologi J2EE (*Java 2 Enterprise Edition*) yang berbasis J2SE yang diawali dengan servlet dan EJB kemudian diikuti JSP.

Terakhir teknologi Java melahirkan J2ME yang sudah diadopsi oleh Nokia, Siemens, Sony Ericsson, Motorola, Samsung untuk menghasilkan aplikasi mobile baik *games* maupun *software* bisnis dan berbagai jenis *software* lain yang dapat dijalankan di peralatan *mobile* seperti ponsel. (Isak Rickyanto, 2003)

Java merupakan bahasa pemrograman berorientasi objek dan bebas *platform*, dikembangkan oleh SUN *Micro System*. Gambar di bawah ini memperlihatkan keunggulan-keunggulan Java. .



Gambar 2.3 Keunggulan-keunggulan Java

2.8.2 Kelebihan Java

Java merupakan teknologi yang unik dan revolusioner dan merupakan teknologi pertama di dunia *software* yang memiliki semboyan “*write once, run anywhere*”. Semboyan tersebut telah terbukti karena banyak program Java dapat dijalankan di berbagai *platform* Sistem Operasi, seperti Linux, Windows maupun Unix.

Java telah mengatasi masalah portabilitas yang sering menjadi kendala dan hambatan dalam pembuatan suatu aplikasi *software*. Mengapa? Karena *software developer* harus mengeluarkan banyak tenaga, pikiran dan waktu untuk menghasilkan aplikasi yang dapat berjalan di *operating system* atau *platform* lain.

Java juga di desain untuk menghasilkan program dengan seminimal mungkin *bug* karena kemampuan sebagai berikut :

- a. Java didesain untuk menghilangkan alokasi memori dan dealokasi memori secara manual. Java memiliki *garbage collection* otomatis yang mencegah adanya *memory leak*. Memory leak adalah masalah yang sering dihadapi programmer C dan C++ di mana memori yang digunakan untuk objek atau variabel yang sudah tidak digunakan tidak didealokasikan sehingga menyebabkan kehabisan memori karena proses alokasi maupun dealokasi yang tidak diatur dengan baik.

- b. Java memiliki *array* yang tidak memerlukan *pointer* sehingga memudahkan para *programmer*.
- c. Java menghilangkan banyak kebingungan apabila terjadi proses assignment (pemberian nilai) pada *statement* kondisial seperti berikut : `if (varnya = 5)`

Kode di atas menyebabkan program Java tidak dapat dikompilasi karena Java membedakan tanda = yang digunakan untuk pemberian nilai dan untuk pengecekan kondisi *True* atau *False* yang harus menggunakan tanda = ganda (`= =`).
- d. Java menghilangkan multiple *inheritance* pada C++ dan menggunakan *interface* yang memiliki kemampuan yang sama tetapi lebih sederhana.

2.8.3 Karakteristik java

Secara singkat, karakteristik Java adalah :

- a. Sederhana

Java tidak memiliki sintaks yang aneh tetapi banyak menggunakan sintaks C++ yang sudah banyak dikenal sehingga java tidak menyulitkan bagi para programmer. Bahkan Java memberikan banyak keunggulan dan kemudahan dibanding C++.

b. Berorientasi Objek

Java merupakan pemrograman berorientasi objek yang murni. Dalam pemrograman Java semua adalah objek, terkecuali tipe data primitive.

c. Dapat didistribusikan dengan mudah

Sifat terdistribusi dari Java sangat tampak sebagai *applet* dan *library* yang mampu bekerja dalam jaringan dan bekerja dengan objek terdistribusi (RMI) dengan sangat baik. Saat ini Java juga memiliki kemampuan untuk *server side processing*, yaitu teknologi *Java Servlet* dan *Java Sever Pages*.

d. Aman

Aman karena program Java memiliki *library security* serta *policy* yang membatasi akses applet di komputer *client*.

e. Diinterpretasi oleh interpreter

Java memerlukan *virtual machine* yang bertindak sebagai interpreter yang menterjemahkan bytecode (*file class*) menjadi bahasa mesin yang dimengerti oleh komputer *host*.

f. Portabel

Portabel karena Java dapat dijalankan di berbagai *platform* tanpa perubahan kode sama sekali.

g. Multithreading

Java memiliki kemampuan untuk menangani dan menjalankan banyak *thread* sekaligus.

h. Dinamis

Java merupakan teknologi yang terus berkembang dan hal ini tampak nyata sekali dengan *library* yang terus ditingkatkan kemampuan dan kelengkapannya. Bahkan Java saat ini telah menjadi teknologi handal untuk aplikasi mobile dengan adanya *Java 2 Micro Edition* (J2ME).

i. Netral terhadap arsitektur hardware

Java dapat dijalankan dengan baik pada komputer yang memiliki arsitektur berbeda-beda.

j. Robust

Java merupakan teknologi yang mampu menolong programmer untuk menghasilkan program secara cepat dan handal karena Java mencegah adanya *memory leaking*, meniadakan *pointer* (yang merupakan keharusan pada programmer C dan C++) serta mencegah berbagai *error* yang mungkin terjadi dengan adanya berbagai proses pengecekan awal pada kompilasi. (Isak Rickyanto, 2003 : 5 - 6)

2.8.4 Pengertian platform java

Platform merupakan lingkungan perangkat keras atau perangkat lunak di mana suatu program dijalankan. *Platform* Java berbeda

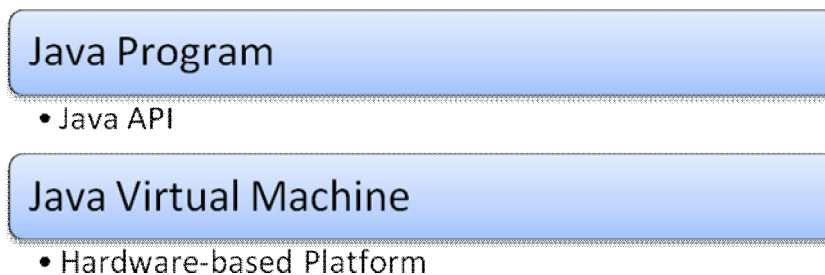
dengan banyak *platform* yang lain. Dalam *platform* Java, platform perangkat lunak berjalan di atas *platform* berbasis perangkat keras. Kebanyakan *platform* yang merupakan kombinasi antara perangkat keras dan sistem operasi.

Platform Java memiliki dua komponen :

- a. Java Virtual Machine (JVM)
- b. Java Application Programming Interface (Java API)

Java API menggunakan kumpulan komponen perangkat lunak yang siap buat yang menyediakan berbagai fasilitas, seperti GUI *widget*. Java API dikelompokkan dalam paket (*package*) komponen-komponen yang berkaitan.

(Benyamin L Sinaga, 2005 : 10)



Gambar 2.4 Platform Java (sumber: Java Tutorial)

Dari gambar di atas terlihat, dalam setiap program Java berjalan di atas platform Java. Platform Java mengisolasi program Java dengan perangkat keras, sehingga program Java tidak bergantung dengan perangkat keras yang digunakan (*hardware independent*).

Berbeda dengan program dalam bahasa lain, program Java menggunakan *compiler* dan *interpreter*. Dengan menggunakan *compiler*

program Java akan diubah menjadi kode antara (*intermediate language*) disebut dengan Java *bytecode*. *Byte code* ini merupakan kode yang tidak bergantung pada *platform*, yang oleh interpreter akan dijalankan pada komputer. Java *bytecode* ini merupakan kode instruksi bagi Java *Virtual Machine*. Interpreter merupakan implementasi dari Java *Virtual Machine*. Dengan menggunakan Java *bytecode* yang tidak bergantung pada platform tersebut, Java dikenal memiliki jargon “**Write Once Run Anywhere**”. (Sigit, W Aloysius, 2008 : 24)

2.8.5 Arsitektur J2ME

Platform J2ME menyediakan sebuah platform bersama (*common*) untuk pemakai dan peralatan-peralatan tertanam (*embedded devices*) seperti *mobile phone*, PDA, *TV set-top boxes*, *game consoles*, dan *point of sale (POS) terminals*, maupun sejumlah besar jenis peralatan tertanam yang lain. Arsitektur J2ME mendefinisikan sekumpulan standar tentang konfigurasi, profil, dan paket-paket pilihan yang dirancang untuk memenuhi kebutuhan dari peralatan-peralatan yang disebutkan di atas. Himpunan (atau kumpulan) standar yang dimaksud memampukan user untuk menulis aplikasi-aplikasi untuk sejumlah besar jenis-jenis peralatan. Selanjutnya aplikasi-aplikasi ini membangkitkan kemampuan-kemampuan masing-masing peralatan.

Platform J2ME disebarluaskan ke jutaan pemakai peralatan di seluruh dunia. Ketika *user* mengembangkan aplikasi-aplikasi untuk platform

J2ME, *user* harus memahami tidak hanya pada arsitektur platform, akan tetapi juga terhadap peralatan-peralatan yang disupport oleh arsitektur ini.

Untuk memastikan bahwa platform J2ME berfungsi pada bermacam-macam peralatan, *user* sebaiknya memperhatikan persyaratan-persyaratan berikut ini :

- Kebutuhan – kebutuhan pengguna (bukan kreatifitas pengguna) harus menggerakkan fasilitas-fasilitas, fitur-fitur dan kebutuhan-kebutuhan sumberdaya untuk peralatan.
- Penjual (*vendor*) harus mempunyai fleksibilitas untuk menambah fungsionalitas baru sebagaimana yang diperlukan.
- J2ME harus menyediakan konfigurasi-konfigurasi yang ditargetkan pada kebutuhan-kebutuhan dari keluarga peralatan-peralatan yang berbeda.
- J2ME harus berbasis pada proses-proses standar terbuka untuk memastikan kesepakatan antar semua kelompok (*parties*).

2.9 PERANGKAT MOBILE

Perangkat *mobile* memiliki banyak jenis dalam hal ukuran, desain dan *layout*. Tetapi mereka memiliki kesamaan karakteristik yang sangat berbeda dari sistem *desktop* yaitu :

1. Ukuran yang kecil

Perangkat *mobile* memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.

2. Memori yang terbatas

Perangkat *mobile* juga memiliki memori yang kecil, yaitu *primary* (RAM) dan *secondary* (disk). Pembatasan ini adalah salah satu faktor yang mempengaruhi penulisan program untuk berbagai jenis dari perangkat ini. Dengan pembatasan jumlah dari memori, pertimbangan-pertimbangan khusus yang harus diambil untuk memelihara pemakaian dari sumber daya yang mahal.

3. Daya proses yang terbatas.

Sistem *mobile* tidak setangguh *desktop*. Ukuran teknologi dan biaya adalah faktor yang mempengaruhi status dari sumber daya ini. Seperti *hard disk* dan RAM, anda dapat menemukan mereka dalam ukuran yang pas dengan kemasan kecil.

4. Mengonsumsi daya yang rendah

Perangkat *mobile* menghasilkan sedikit daya dibandingkan dengan mesin *desktop*. Perangkat ini harus menghemat daya karena mereka berjalan dalam keadaan dimana daya yang disediakan dibatasi oleh baterai-baterai.

5. Kuat dan dapat diandalkan.

Karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat menghadapi benturan-benturan, gerakan dan sesekali tetesan-tetesan air.

6. Konektivitas yang terbatas.

Perangkat *mobile* memiliki *bandwith-bandwtih* rendah, beberapa dari mereka bahkan tidak tersambung. Kebanyakan dari mereka menggunakan koneksi *wireless*.

7. Masa hidup yang pendek.

Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala. Coba ambil kasus sebuah *handphone*, mereka *booting* dalam hitungan detik, dan kebanyakan mereka tidak mematikan *handphone*.

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1. Analisa Permasalahan

Analisa permasalahan merupakan pembahasan tentang masalah apa yang akan dibahas oleh peneliti. Masyarakat ingin mengetahui lebih banyak dan sebanyak mungkin tentang masalah penyakit kelamin, namun dengan berkembangnya teknologi informasi di era modernisasi saat ini, masyarakat ingin segala hal yang serba instan. Tidak ada banyak waktu bagi mereka untuk sekedar ke toko buku hanya untuk sekedar mendapatkan informasi yang ingin didapatkan.

Untuk memenuhi kebutuhan masyarakat yang mana rasa keingintahuannya lebih besar dan ingin serba cepat, mudah, efektif dan efisien serta *fleksibel* dalam penggunaannya, maka penulis bermaksud membuat suatu sistem aplikasi yang dapat di *install* melalui *mobile* berbasis Java. Sehingga dapat membantu masyarakat untuk dapat mengetahui jenis penyakit kelamin berdasarkan gejala dan penyebabnya.

Tabel 3.1. Data Jenis Penyakit Kelamin

No	Jenis Kelamin	Nama Penyakit	Ciri-ciri Penyakit
1	Pria	a.Uretritis : yang paling sering dijumpai adalah uretritis anterior akuta dan dapat menjalar ke proksimal,selanjutnya mengakibatkan komplikasi	Rasa gatal, panas dibagian distral uretra disekitar orifisium uretra eksternum, kemudian disusul disuria, pola kisuria keluar duh tubuh dari ujung uretra yang kadang-kadang

	<p>lokal, asendens, dan diseminata.</p>	<p>disertai darah, dan disertai perasaan nyeri pada waktu ereksi. Pada pemeriksaan tampak orifisium uretra eksternum eritematosa, edematosa, dan ektropion. Tampak duh tubuh yang mukopurulen, dan pada beberapa kasus dapat terjadi pembesaran kelenjar getah bening inguinal unilateral atau bilateral.</p>
	<p>b. Tysonitis : kelenjar tyson ialah kelenjar yang menghasilkan smegma. Infeksi biasanya terjadi pada penderita dengan preputium yang sangat panjang dan kebersihan yang kurang baik. Diagnosis dibuat berdasarkan ditemukannya butir pus atau pembengkakan pada daerah frenulum yang nyeri tekan. Bila duktus tertutup akan timbul abses dan merupakan sumber infeksi laten.</p>	<p>Rasa gatal, panas dibagian distal uretra disekitar orifisium uretra eksternum, kemudian disusul disuria, pola kisuria keluar duh tubuh dari ujung uretra yang kadang-kadang disertai darah, dan disertai perasaan nyeri pada waktu ereksi. Pada pemeriksaan tampak orifisium uretra eksternum eritematosa, edematosa, dan ektropion. Tampak duh tubuh yang mukopurulen, dan pada beberapa kasus dapat terjadi pembesaran kelenjar getah bening inguinal unilateral atau bilateral.</p>
	<p>c. Paraurethritis : sering pada orang dengan orifisium uretra eksternum terbuka atau hipospadia. Infeksi pada duktus ditandai dengan butir pus pada kedua muara parauretra.</p>	<p>Rasa gatal, panas dibagian distal uretra disekitar orifisium uretra eksternum, kemudian disusul disuria, pola kisuria keluar duh tubuh dari ujung uretra yang kadang-kadang disertai darah, dan disertai perasaan nyeri pada waktu</p>

			ereksi.
		d. Littritis : Infeksi pada duktus ditandai dengan butir pus pada kedua muara parauretra.	Tidak ada gejala khusus, hanya pada urin ditemukan benang-benang atau butir-butir. Bila salah satu saluran tersumbat, dapat terjadi abses folikular. Didiagnosis dengan uretroskopi.
		e. Cowperitis : bila hanya duktus yang terkena biasanya tanpa gejala. Kalau infeksi terjadi pada kelenjar Cowper dapat terjadi abses.	Keluhan berupa nyeri dan adanya benjolan pada daerah perineum disertai rasa penuh dan panas, nyeri pada waktu defekasi, dan disuria. Jika tidak diobati abses akan pecah melalui kulit perineum, uretra, atau rektum dan mengakibatkan proktitis.
		f. Prostatitis : pembesaran prostat dengan konsistensi kenyal, berbentuk nodus, nyeri tekan, dan didapatkan fluktuasi bila telah terjadi abses.	Ditandai dengan perasaan tidak enak pada daerah perineum dan suprapubis, malse, demam, nyeri kencing sampai hematuria, spasme otot uretra sehingga terjadi retensi urin, tenesmus ani, sulit buang air besar, dan obstipasi.
		g. Vesikulitis : radang akut yang mengenai vesikula seminalis dan duktus ejakulatoris, dapat timbul menyertai prostatitis akut atau epididimitis akut.	Demam, polakisuria, hematuria terminal, nyeri pada waktu ereksi atau ejakulasi, dan spasme mengandung darah. Vesikula/seminalis yang membengkak dan keras seperti sosis, memanjang diatas prostat.
		h. Vas deferentitis/funikulitis	Perasaan nyeri pada daerah abdomen bagian bawah pada sisi yang sama.

		i. Epididimitis : keadaan yang mempermudah timbulnya epididimitis ini adalah trauma pada uretra posterior yang disebabkan oleh salah penanganan atau kelalaian penderita sendiri. Faktor yang mempengaruhi keadaan ini antara lain irigasi yang terlalu sering dilakukan. Pengurutan prostat yang berlebihan, atau aktivitas seksual dan jasmani yang berlebihan.	Cairan irigator terlalu panas atau terlalu pekat, instrumentasi yang kasar, epididimitis dan tali spermatis membengkak dan teraba panas, juga testis sehingga menyerupai hidrokel sekunder. Pada penekanan terasa nyeri sekali, bila mengenai kedua epididimitis dapat mengakibatkan sterilitas.
		j. Trigonitis : infeksi ascendens dari uretra posterior dapat mengenai trigonum vesicae urinaria.	Trigonitis menimbulkan gejala poliuria, disuria terminal, dan hematuria.
2	Wanita	a. Uretritis	Gejala utama ialah disuria, kadang-kadang poliuria. orifisium uretra eksternum tampak merah, edematosa dan ada sekret mukopurulen.
		b. Parauretritis/Skenitis	Kelenjar parauretra dapat terkena, tetapi abses jarang terjadi.
		c. Servisititis	Rasa nyeri pada punggung bawah, serviks tampak merah dengan erosi dan sekret mukopurulen. Duh tubuh akan terlihat lebih banyak.
		d. Bartholinitis : bila saluran kelenjar tersumbat dapat timbul abses dan dapat pecah melalui mukosa atau kulit. Kalau tidak diobati dapat menjadi rekuren atau kista.	Labium mayoral pada sisi yang terkena membengkak, merah dan nyeri tekan. Kelenjar bartholin membengkak, terasa nyeri sekali ketika penderita berjalan, penderita sukar duduk

		e. Salpingitis : peradangan dapat bersifat akut, subakut atau kronis. Cara infeksi langsung dari serviks melalui tuba fallopi sampai pada daerah salping dan ovarium sehingga dapat menimbulkan penyakit radang panggul. Infeksi ini dapat menimbulkan kehamilan ektopik dan sterilitas.	Nyeri pada daerah abdomen bawah, duh tubuh vagina, disuria, dan menstruasi yang tidak teratur atau abnormal.
		f. Proktitis : dapat terjadi karena kontaminasi dari vagina dan kadang-kadang karena hubungan genitoanal.	Terasa seperti terbakar pada daerah anus dan pada pemeriksaan tampak mukosa eritematosa, edematosa, dan tertutup pus mukopurulen.
		g. Orofaringitis : cara infeksi melalui kontak secara orogenital.	Keluhan sering bersifat asimtomatik, orofaring tampak eksudat mukopurulen yang ringan atau sedang.
		h. Konjungtivitis : penyakit ini dapat terjadi pada bayi yang baru lahir dari ibu yang menderita servisititis gonore. Pada orang dewasa infeksi terjadi karena penularan pada konjungtiva melalui tangan atau alat-alat.	Keluhannya berupa fotofobi, konjungtiva bengkak dan merah dan keluar eksudat mukopurulen. Bila tidak diobati dapat berakibat terjadinya ulkus kornea, panoftalmitis sampai timbul kebutaan.
		i. Gonore diseminata : kira-kira 1% kasus gonore akan berlanjut menjadi gonore diseminata. Penyakit ini banyak didapat pada penderita dengan gonore asimtomatik sebelumnya, terutama pada wanita.	Gejala yang timbul dapat berupa artritis (terutama mono artritis) miokarditis, endokarditis, perikarditis, meningitis, dan dermatitis.

Sumber : Djuanda, Ilmu Penyakit Kulit dan Kelamin, Fakultas Kedokteran

Universitas Indonesia (2007)

3.2. Deskripsi Sistem Yang Akan Dibuat

Aplikasi yang akan dibuat ini adalah sebuah sistem dimana *user* dapat mengetahui informasi penyakit kelamin berdasarkan jenis kelamin, meliputi nama penyakit kelamin, gejala penyakit, penyebab penyakit, dan cara penyembuhan penyakit. Untuk dapat mengetahui penyakit kelamin tersebut, maka sebelumnya *user* diharuskan untuk menginstall aplikasi ini pada *handphone* yang mana di dalamnya terdapat fasilitas berbasis Java.

Setelah aplikasi di *install* ke dalam *mobile*, maka *user* dapat membuka sistem tersebut dengan meng-klik pada gambar (*icon*) yang terdapat dalam *handphone* tersebut. Selanjutnya *user* akan menemukan suatu tampilan awal berupa menu utama, *user* cukup menekan tombol **GO**. Dimana di dalam **menu utama** tersebut terdapat beberapa pilihan yang dapat di pilih sesuai keinginan *user* yaitu jenis penyakit kelamin pada pria, jenis penyakit kelamin pada wanita, gejala penyakit kelamin pada pria, gejala penyakit kelamin pada wanita, penyebab penyakit kelamin pada pria, penyebab penyakit kelamin pada wanita, cara penyembuhan penyakit kelamin pada pria, dan cara penyembuhan penyakit kelamin pada wanita.

Untuk melanjutkan pada proses selanjutnya, maka *user* dapat menekan tombol **OK**. Selanjutnya *user* akan dapat mengetahui informasi dari menu yang dipilih.

Apabila *user* menekan tombol **kembali** pada pilihan yang ada pada *mobile*, maka akan tampil menu sebelumnya. Untuk perintah keluar dari aplikasi ini maka *user* dapat memilih *icon* dengan tulisan keluar.

3.3. Analisa Kebutuhan Sistem

3.3.1. Kebutuhan *Hardware*

Hardware merupakan perangkat keras dari suatu sistem komputer. Adapun kebutuhan *hardware* yang akan digunakan untuk membangun sistem tersebut memiliki spesifikasi minimal :

- a. Processor Pentium IV 1.5 GHZ
- b. Memori VGA 32 MB
- c. RAM 512 MB
- d. Harddisk 80 MB
- e. Keyboard, mouse standar
- f. Handphone yang digunakan pada penelitian ini adalah handphone yang berfasilitas Java.

3.3.2. Kebutuhan *Software*

Software adalah perangkat yang memungkinkan komputer mengerjakan perintah-perintah yang kita berikan. Disebut dengan istilah lunak karena sifatnya yang tidak berupa peralatan fisik. *Software* yang diperlukan oleh penulis adalah :

- a. Sistem Operasi Windows XP 1 sebagai sistem operasi untuk menjalankan semua program yang ada di komputer.
- b. Java Platform Micro Edition (J2ME) sebagai aplikasi yang akan dipasang pada *mobile*.
- c. NetBeans IDE 6.8 sebagai compiler

3.3.3. Kebutuhan *Brainware*

Brainware adalah orang yang akan mengoperasikan sistem komputer. Dalam sistem ini yang menjadi *brainware* adalah siapa saja yang memakai sistem ini serta pembuat sistem itu sendiri. Untuk pembuat sistem ini diperlukan keahlian di bidang bahasa Java. Sedangkan bagi pengguna tidak memerlukan keahlian tertentu melainkan hanya mempunyai *mobile (handphone)* berbasis Java.

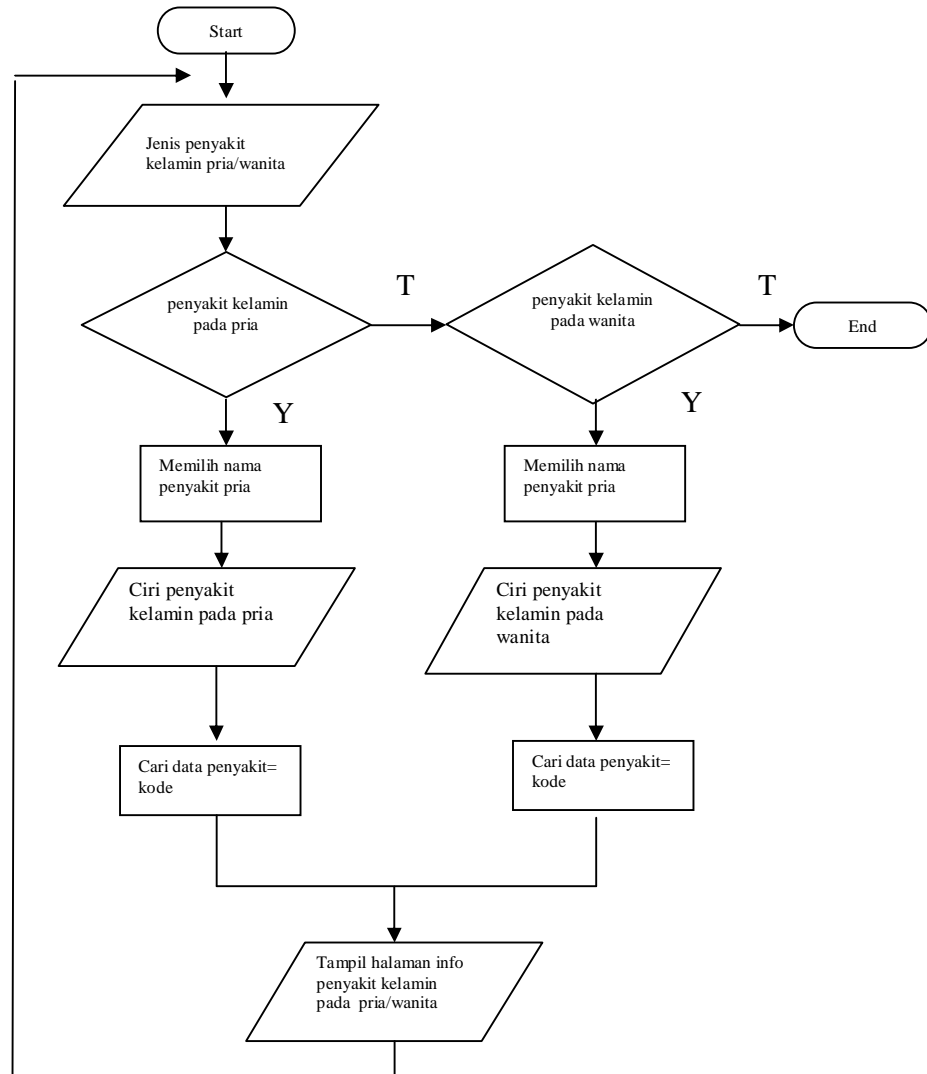
3.4. Perancangan Sistem

Setelah dilakukan tahap penganalisaan, maka tahap berikutnya adalah tahap perancangan. Proses perancangan akan memberikan gambaran tentang perangkat lunak yang akan dibuat nantinya atau yang akan dikembangkan. Pada bagian perancangan ini akan dijelaskan komponen-komponen pembangun atau *tools* yang ada dalam rangka pembangunan aplikasi sistem informasi penyakit kelamin.

Perancangan sistem menggunakan alat bantu atau tools *flow chart*. Flow Chart adalah suatu bagan yang menggambarkan arus logika dari data yang akan diproses dalam suatu program dari awal sampai akhir. Bagan ini merupakan alat yang berguna untuk mempersiapkan program sistem informasi penyakit kelamin.

3.5. Flowchart

Menu Utama



Gambar 3.1 Flowchart Menu Utama

Keterangan gambar 3.1

1. Mulai
2. Tampil jenis penyakit kelamin pria / wanita
3. Masuk ke pilihan kondisi pertama, apabila ya, memilih nama penyakit kelamin pada pria, tampil ciri-ciri penyakit kelamin pria,tampil halaman informasi penyakit kelamin pada pria. Apabila tidak, maka
4. Masuk ke pilihan kondisi kedua, apabila ya, memilih nama penyakit kelamin pada wanita, tampil ciri-ciri penyakit kelamin wanita,tampil halaman informasi penyakit kelamin pada wanita. Apabila tidak, maka
5. Selesai, bila ya, maka masuk ke jenis penyakit kelamin pria/wanita

.

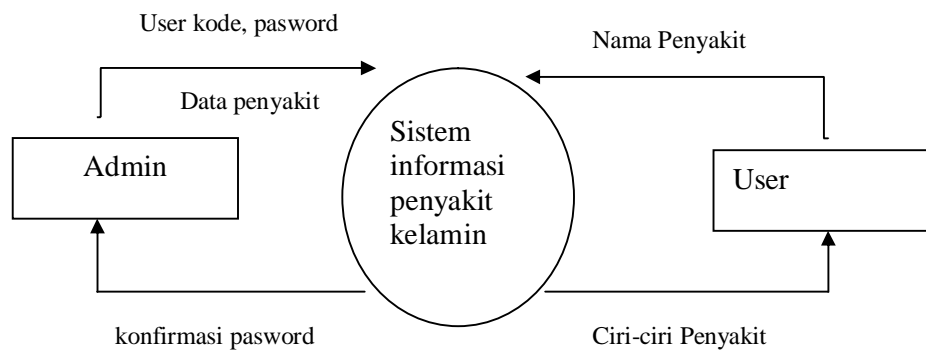
3.6. Desain Sistem

3.6.1 Data Flow Diagram (DFD)

Pada bagian ini akan diberikan gambaran perancangan sistem informasi penyakit kelamin dengan menggunakan diagram alir.

1. Context Diagram

Context Diagram (CD) pada gambar 3.2 merupakan data arus awal , juga disebut DFD level 0. CD ini memiliki satu Externa entity yaitu user dan admin dan sebuah proses yaitu sistem informasi penyakit kelamin.

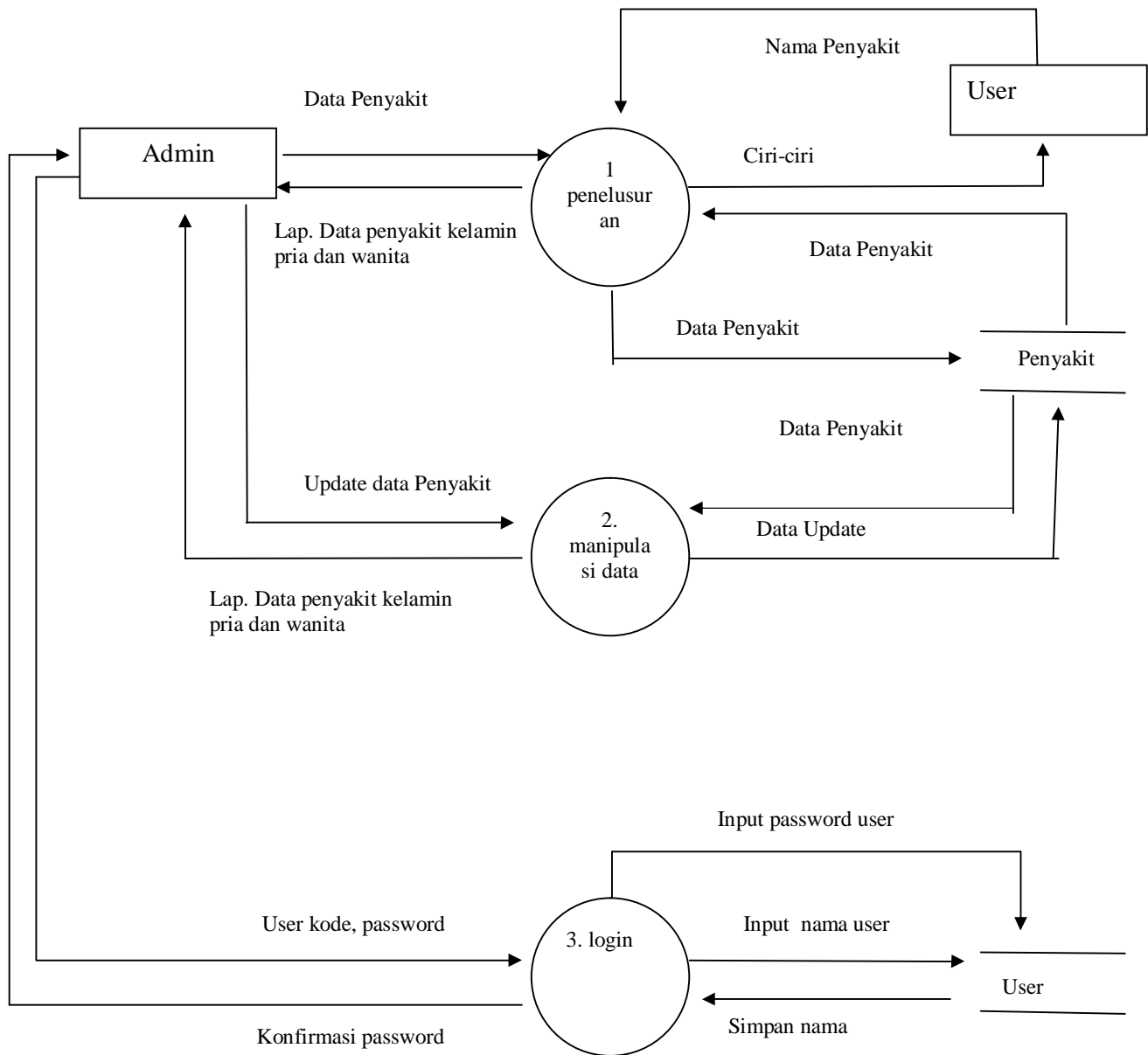


Lap. Data penyakit kelamin pria dan wanita

Gambar 3.2 Data Context Diagram

2.DFD level 1

DFD level 1 paada gambar 3.2 ini merupakan penjabaran darri proses CD . pada DFD level 1 ini mempunyai dua proses yaitu proses penelusuran dan proses manipulasi data.



Gambar 3.3 DFD level 1

3. Kamus data

a. Penyakit = data penyakit = update data penyakit = data update

lap.data penyakit kelamin pria dan wanita = @kd_penyakit

Kd_penyakit= text(5)

Nm_penyakit= text(35)

Cr_penyakit= text(255)

b. User = data user = ciri-ciri= @kd_user

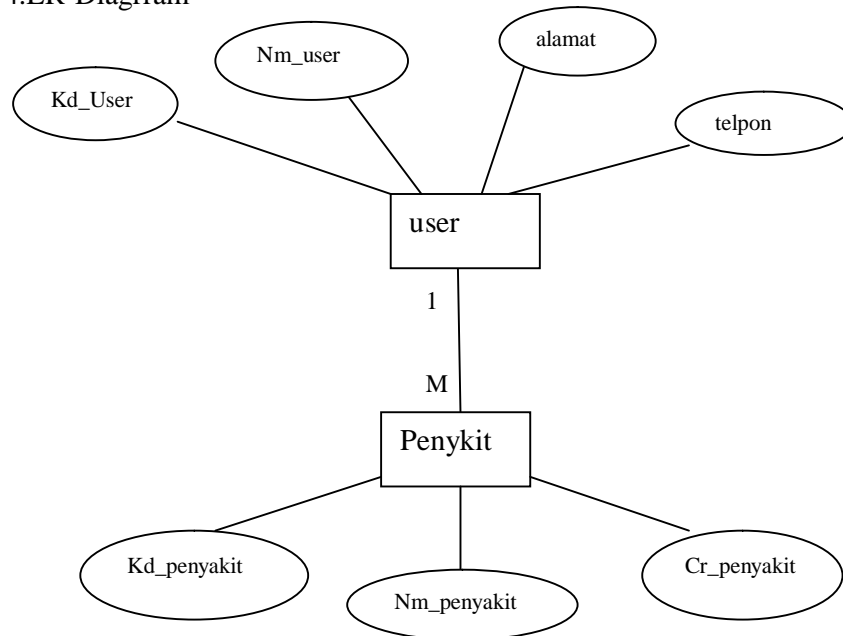
Kd_user=text(5)

Nm_User=text(25)

Alamat=text(80)

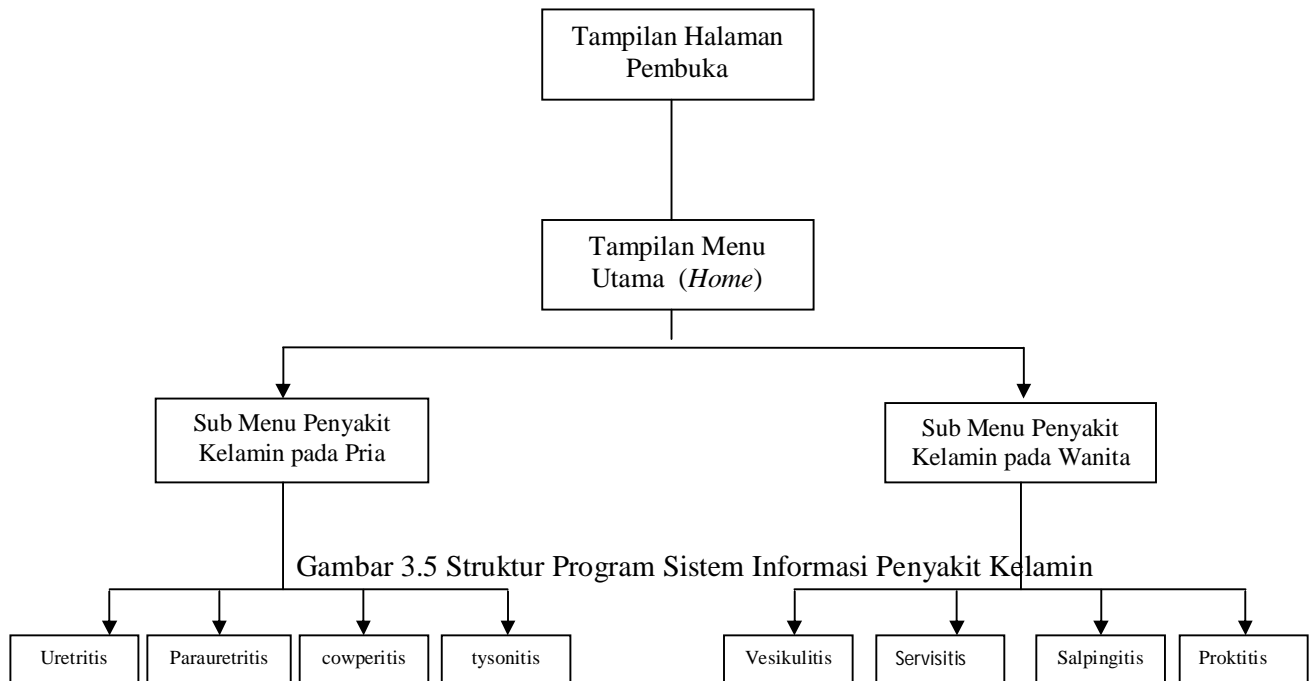
Telpon=text(15)

4.ER-Diagram



Gambar 3.4 E-R Diagram

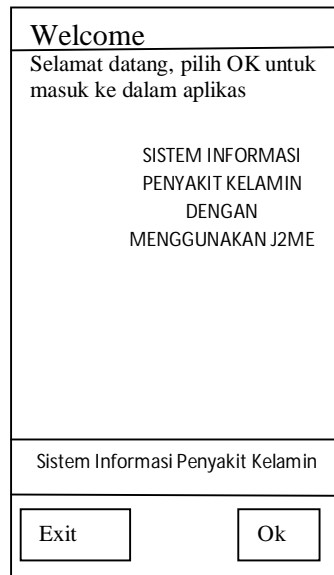
3.7. Struktur Program



3.8. Perancangan Halaman Input dan Output

3.8.1. Halaman Pembuka

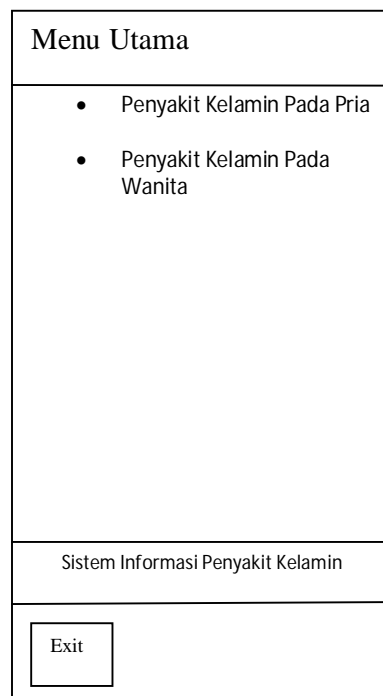
Saat program sistem informasi penyakit kelamin ini pertama kali dijalankan, akan tampil layar pembuka (*splash*) yang berisi gambar dan teks judul program informasi yang tampil selama 3 detik sebelum masuk ke menu utama.



Gambar 3.6 Perancangan Halaman Pembuka

3.8.2. Halaman Menu Utama

Menu utama sistem informasi penyakit kelamin menampilkan dua pilihan fasilitas informasi yaitu menu informasi penyakit kelamin pada pria, dan menu informasi penyakit kelamin pada wanita.



Gambar 3.7 Perancangan Menu Utama

3.8.3. Halaman Menu Penyakit Kelamin Pada Pria

Halaman menu penyakit kelamin pada pria merupakan halaman yang digunakan untuk menampung submenu-submenu yang ditawarkan oleh program informasi ini. Submenu yang ditampilkan yaitu berbagai jenis penyakit kelamin yang bisa terjadi pada pria.

Daftar nama penyakit kelamin pria
<ul style="list-style-type: none">➤ Uretritis➤ Tysonitis➤ Parauretritis➤ Litritis➤ Cowperitis➤ Prostatitis➤ Vesikulitis➤ Vas deferentitis /funikulitis➤ Epididimitis➤ Trigonitis
Daftar Nama Penyakit Kelamin Pria
Back

Gambar 3.8 Perancangan Menu Penyakit Kelamin Pada Pria

3.8.4. Halaman Submenu Penyakit Kelamin Pada Pria

Halaman informasi penyakit kelamin menampilkan informasi nama atau jenis penyakit kelamin yang dapat terjadi pada pria, penyebab masing-masing jenis penyakit kelamin pada pria, gejala yang timbul berdasarkan penyakit jenis penyakit kelamin pada pria, cara penyembuhan berdasarkan jenis penyakit kelamin pada pria.

Ciri-ciri Penyakit Kelamin Pada Pria
<ol style="list-style-type: none">1. Nama/Jenis penyakit kelamin pada pria2. Penyebab penyakit kelamin berdasarkan jenis penyakit kelamin pada pria3. Gejala yang timbul berdasarkan jenis penyakit kelamin pada pria
Gambar Dampak Penyakit
Penyakit Kelamin Pria
Back

Gambar 3.9 Perancangan Submenu Penyakit Kelamin Pada Pria

3.8.5. Halaman Menu Penyakit Kelamin Pada Wanita

Halaman menu penyakit kelamin pada wanita merupakan halaman yang digunakan untuk menampung submenu-submenu yang ditawarkan oleh program informasi ini. Submenu yang ditampilkan yaitu berbagai jenis penyakit kelamin yang bisa terjadi pada wanita.

Daftar Nama Penyakit Kelamin Wanita
<ul style="list-style-type: none">➤ Uretritis➤ Parauretritis/Skenitis➤ Servitis➤ Bartholinitis➤ Salpingitis➤ Proktitis➤ Orofaringitis➤ Konjungtivitis➤ Gonore diseminata
Penyakit Kelamin Wanita
Back

Gambar 3.10 Perancangan Menu Penyakit Kelamin Pada Wanita

3.8.6. Perancangan Submenu Penyakit Kelamin Pada Wanita

Halaman informasi penyakit kelamin menampilkan informasi nama atau jenis penyakit kelamin yang dapat terjadi pada wanita, penyebab masing-masing jenis penyakit kelamin pada wanita, gejala yang timbul berdasarkan penyakit jenis penyakit kelamin pada wanita, cara penyembuhan berdasarkan jenis penyakit kelamin pada wanita.

Informasi Penyakit Kelamin Pada Wanita
<ol style="list-style-type: none">1. Nama/Jenis penyakit kelamin pada wanita2. Penyebab penyakit kelamin berdasarkan jenis penyakit kelamin pada wanita3. Gejala yang timbul berdasarkan jenis penyakit kelamin pada wanita
Gambar Dampak Penyakit
Penyakit Kelamin Wanita
Back

Gambar 3.11 Perancangan Submenu Penyakit Kelamin Pada Wanita

BAB IV

IMPLEMENTASI SISTEM

Implementasi merupakan satu tahap dalam perancangan suatu perangkat lunak. Tahap implementasi dilakukan setelah proses analisa dan perancangan selesai dilakukan. Pada tahap implementasi akan di bahas hal-hal yang berkaitan secara langsung dengan perangkat lunak.

4.1 Implementasi Sistem

4.1.1 Implementasi *Hardware*

Hardware merupakan perangkat keras atau peralatan fisik dari sistem komputer yang berupa mesin-mesin serta rangkaian elektronik. Spesifikasi minimal *hardware* yang digunakan dalam rancang bangun sistem informasi penyakit kelamin dengan menggunakan J2ME yaitu berupa :

Tabel 4.1

Spesifikasi Minimal *Hardware*

Processor	Intel Pentium IV 3.0 GHz
Harddisk	80 GB
Memory	256 MB
Monitor	CRT 17 Inchi
Keyboard	Standart
Mouse	Standart

4.1.2 Implementasi *Software*

Software merupakan perangkat lunak dari sistem komputer yang berupa kumpulan bahasa pemrograman. *Software* yang digunakan dalam implementasi program ini yaitu :

1. Microsoft Windows XP

Alasan pemilihan Windows XP karena :

- a. Tampilan yang *user friendly* atau lebih menarik dibanding dengan Windows sebelumnya.
- b. Kompatibel dengan *software-software* lain.
- c. Lebih mudah mengoperasikannya.

2. NetBeans IDE 6.8

Alasan pemilihan NetBeans karena :

- a. NetBeans merupakan salah satu IDE yang paling tangguh saat ini dalam melakukan pemrograman Java.
- b. NetBeans menyediakan paket yang lengkap dalam pemrograman dari pemrograman standar (aplikasi desktop), pemrograman enterprise, dan pemrograman perangkat mobile.
- c. NetBeans mempunyai beberapa bagian yang memudahkan pengembang untuk melakukan pemrograman.

3. J2ME

Alasan pemilihan J2ME

- a. Teknologi yang fokus kepada perangkat konsumen (mobile).
- b. Perangkat ini memiliki jumlah memori yang terbatas.
- c. Menghabiskan sedikit daya dari baterai.

d. Layar yang kecil dan bandwidth jaringan yang rendah.

4.1.3 Pemilihan *Brainware*

Brainware sering juga disebut pemakai komputer atau orang yang menjalankan peralatan komputer. Ada beberapa istilah yang berhubungan dengan *brainware*, diantaranya adalah operator, *brainware* harus memiliki keahlian dalam melaksanakan instruksi-instruksi program. Agar sistem berjalan dengan baik, *brainware* diharapkan memenuhi syarat sebagai berikut :

1. User

Merupakan pengguna yang terlibat langsung dengan komputer atau *handphone* dalam menjalankan program aplikasi. User yang diperlukan yaitu semua orang yang bisa mengoperasikan *handphone*.

2. Programmer

Merupakan orang yang menguasai bahasa pemrograman dan mampu mengembangkan sistem aplikasi. Dalam pembuatan program ini dibutuhkan seorang programmer yang menguasai bahasa pemrograman J2ME, dan NetBeans.

4.2 Pengujian Sistem

4.2.1. Implementasi Tampilan Pembuka



Gambar 4.1 Tampilan Layar Pembuka

Dari gambar 4.1. dapat dijelaskan bahwa saat program informasi ini pertama kali dijalankan, akan tampil layar pembuka (*splash*) yang berisi gambar dan teks judul program informasi yang tampil selama 3 detik sebelum masuk ke menu utama.

4.2.2. Implementasi Tampilan Menu Utama



Gambar 4.2 Tampilan Menu Utama

Dari gambar 4.2 dapat dijelaskan bahwa halaman menu utama merupakan halaman yang digunakan untuk menampung submenu-submenu yang ditawarkan oleh program informasi ini. Dalam halaman menu utama yang sifatnya sebagai pembuka dan penghubung antar halaman yang ada pada aplikasi ini memiliki dua pilihan, yaitu penyakit kelamin pada pria, dan penyakit kelamin pada wanita.

4.2.3. Implementasi Menu Penyakit Kelamin Pada Pria



Gambar 4.3 Tampilan Menu Penyakit Kelamin Pada Pria

Dari gambar 4.3 dapat dijelaskan bahwa *form* menu penyakit kelamin pada pria merupakan *form* yang digunakan untuk menampung submenu-submenu yang ditawarkan oleh program informasi ini. Submenu yang ditampilkan yaitu berbagai jenis penyakit kelamin yang bisa terjadi pada pria.

4.2.4. Implementasi Submenu Penyakit Kelamin pada Pria



Gambar 4.4 Tampilan Submenu Penyakit Kelamin Pada Pria

Dari gambar 4.4 dapat dijelaskan bahwa *form* submenu penyakit kelamin pada pria ini merupakan *form* yang digunakan untuk membahas materi tentang ciri-ciri atau gejala penyakit kelamin, tertentu pada pria beserta contoh gambar dampak penyakit.

4.2.5. Implementasi Menu Penyakit Kelamin Pada Wanita



Gambar 4.5 Tampilan Menu Penyakit Kelamin Pada Wanita

Dari gambar 4.5 dapat dijelaskan bahwa *form* menu penyakit kelamin pada wanita merupakan *form* yang digunakan untuk menampung submenu-submenu yang ditawarkan oleh program informasi ini. Submenu yang ditampilkan yaitu berbagai jenis penyakit kelamin yang bisa terjadi pada wanita.

4.2.6. Implementasi Submenu Penyakit Kelamin pada Wanita



Gambar 4.6 Tampilan Submenu Penyakit Kelamin Pada Wanita

Dari gambar 4.6 dapat dijelaskan bahwa *form* submenu penyakit kelamin pada wanita ini merupakan *form* yang digunakan untuk membahas materi tentang ciri-ciri atau gejala penyakit kelamin, tertentu pada wanita beserta contoh gambar dampak penyakit.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pembahasan pada bab-bab sebelumnya mengenai laporan tugas akhir yang berjudul “Rancang Bangun Sistem Informasi Penyakit Kelamin Dengan Menggunakan Java 2 Micro Edition (J2ME)” dapat diambil beberapa kesimpulan, yaitu :

1. User interface Informasi Penyakit Kelamin ini dapat menampilkan informasi tentang jenis-jenis penyakit kelamin pada pria dan wanita beserta ciri-ciri atau gejala yang timbul.
2. Program ini dapat diaplikasikan pada *handphon* dengan spesifikasi handphone yang memiliki aplikasi java .
3. User interface ini memiliki ukuran file yang relatif kecil, sehingga tidak diperlukan ukuran memori yang besar dan dapat dijalankan pada *handphone* yang didukung Java.
4. Dapat dipergunakan sebagai panduan dalam memperoleh informasi jenis-jenis penyakit kelamin pada pria dan wanita.

5.2 Saran

Berikut adalah saran-saran yang sekiranya dapat dipergunakan sebagai masukan dalam pengembangan program selanjutnya :

1. Sebaiknya menggunakan *handphone* dengan fasilitas Java, karena jika *handphone* tidak ada fasilitas Java maka aplikasi ini tidak dapat dijalankan.

2. Aplikasi sebagai media informasi yang penulis buat mungkin masih sederhana, sehingga dapat dikembangkan dan disempurnakan lagi. Dari segi tampilan dapat dikembangkan lebih bagus dan lebih interaktif lagi, sehingga pengguna akan semakin tertarik untuk menggunakan aplikasi program informasi ini.
3. Cakupan materi dapat diperluas, dengan menambahkan solusi atau cara penyembuhan/pengobatan dan cara mencegah timbulnya penyakit kelamin pada pria dan wanita.

DAFTAR PUSTAKA

- Benyamin L Sinaga, . (2005), *Dasar –dasar Pemrograman Java 2*, Elex Media
Kompuntindo, Jakarta..
- Djuanda, Adhi,prof.Dr.dr, (2007), *Ilmu Penyakit Kelamin Dan Kulit, Fakultas*
Kedokteran Universitas Indonesia, Jakarta
- Isak Rickyanto.S.T, (2003), *Java 2 Micro Edition Mobile Interface Device*
Programming. Elex Media Komputindo, Jakarta
- Jogiyanto, H.M. (1998). *Analisis & Desain Sistem Informasi*, Andi Offset,
Yogyakarta.
- Komputer, wahana.(2009). Tutorial 5 Hari Membangun GUI dengan Java Netbeans
6.8, Andi, Yogyakarta
- Sigit, W Aloysius. (2008), *7 Proyek Aplikasi Dengan Java*, Elex Media, Jakarta
- Yourdon, Edward (1989), *Modern structure Analysis*, Prentice-Hall International
Editions, United States of America.

LISTING PROGRAM

Program Penyakit Kelamin

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package hello;

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import org.netbeans.microedition.lcdui.SimpleTableModel;
import org.netbeans.microedition.lcdui.SplashScreen;
import org.netbeans.microedition.lcdui.TableItem;

/**
 * @author Administrator
 */
public class HelloMIDlet extends MIDlet implements CommandListener {

    private boolean midletPaused = false;

    //<editor-fold defaultstate="collapsed" desc=" Generated Fields ">
    private Command exitCommand;
    private Command okCommand;
    private Command itemCommand;
    private Command okCommand1;
    private Command Kembali1;
    private Command Kembali2;
    private Command Kembali3;
    private Command Kembali4;
    private Form WelcomeScreen;
    private StringItem stringItem;
    private ImageItem imageItem8;
    private List MenuList;
    private List PyPriaList;
    private List PyWanitaList;
    private Form FUretritisPria;
    private StringItem stringItem1;
    private ImageItem imageItem2;
    private Form FTysonitisPria;
    private StringItem stringItem2;
    private ImageItem imageItem3;
    private Form FParauretritisPria;
    private StringItem stringItem3;
    private ImageItem imageItem4;
    private Form FLitritisPria;
    private StringItem stringItem4;
    private ImageItem imageItem5;
    private Form FProstatitisPria;
    private StringItem stringItem6;
    private ImageItem imageItem7;
    private Form FCowperitisPria;
    private StringItem stringItem5;
    private ImageItem imageItem6;
```

```

private Form FEpididimitisPria;
private StringItem stringItem9;
private Form FVesikulitisPria;
private StringItem stringItem7;
private Form FVasdeferentitisPria;
private StringItem stringItem8;
private Form FParauretritisWt;
private StringItem stringItem12;
private ImageItem imageItem1;
private Form FTrigonitisPria;
private StringItem stringItem10;
private Form FUretritisWt;
private StringItem stringItem11;
private ImageItem imageItem;
private Form FSalpingitisWt;
private StringItem stringItem15;
private Form FServitisWt;
private StringItem stringItem13;
private Form FBartholinitisWt;
private StringItem stringItem14;
private Form FKonjungtivitisWt;
private StringItem stringItem18;
private Form FGonoreWt;
private StringItem stringItem19;
private Form FProktitisWt;
private StringItem stringItem16;
private Form FOrofaringitisWt;
private StringItem stringItem17;
private SimpleTableModel tableModel1;
private Image image2;
private Image image1;
private Image image10;
private Image image9;
private Image image6;
private Image image3;
private Image image8;
private Image image5;
private Image image4;
private Image image7;
private Ticker ticker4;
private Ticker ticker;
private Ticker ticker1;
private Ticker ticker2;
private Ticker ticker3;
private Image image11;
//</editor-fold>

/**
 * The HelloMIDlet constructor.
 */
public HelloMIDlet() {
}

//<editor-fold defaultstate="collapsed" desc=" Generated Methods ">
//</editor-fold>

```

```

//<editor-fold defaultstate="collapsed" desc=" Generated Method: initialize ">
/**
 * Initalizes the application.
 * It is called only once when the MIDlet is started. The method is called before the
<code>startMIDlet</code> method.
 */
private void initialize() {
    // write pre-initialize user code here

    // write post-initialize user code here
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Method: startMIDlet ">
/**
 * Performs an action assigned to the Mobile Device - MIDlet Started point.
 */
public void startMIDlet() {
    // write pre-action user code here
    switchDisplayable(null, getWelcomeScreen());
    // write post-action user code here
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Method: resumeMIDlet ">
/**
 * Performs an action assigned to the Mobile Device - MIDlet Resumed point.
 */
public void resumeMIDlet() {
    // write pre-action user code here

    // write post-action user code here
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Method: switchDisplayable ">
/**
 * Switches a current displayable in a display. The <code>display</code> instance is taken from
<code>getDisplay</code> method. This method is used by all actions in the design for switching
displayable.
 * @param alert the Alert which is temporarily set to the display; if <code>null</code>, then
<code>nextDisplayable</code> is set immediately
 * @param nextDisplayable the Displayable to be set
 */
public void switchDisplayable(Alert alert, Displayable nextDisplayable) {
    // write pre-switch user code here
    Display display = getDisplay();
    if (alert == null) {
        display.setCurrent(nextDisplayable);
    } else {
        display.setCurrent(alert, nextDisplayable);
    }
    // write post-switch user code here
}
}

```

```
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Method: commandAction for  
Displayables ">
```

```
/**  
 * Called by a system to indicated that a command has been invoked on a particular displayable.  
 * @param command the Command that was invoked  
 * @param displayable the Displayable where the command was invoked  
 */  
public void commandAction(Command command, Displayable displayable) {  
    // write pre-action user code here  
    if (displayable == FBartholinitisWt) {  
        if (command == Kembali4) {  
            // write pre-action user code here  
            switchDisplayable(null, getPyWanitaList());  
            // write post-action user code here  
        }  
    } else if (displayable == FCowperitisPria) {  
        if (command == Kembali4) {  
            // write pre-action user code here  
            switchDisplayable(null, getPyPriaList());  
            // write post-action user code here  
        }  
    } else if (displayable == FEpididimitisPria) {  
        if (command == Kembali4) {  
            // write pre-action user code here  
            switchDisplayable(null, getPyPriaList());  
            // write post-action user code here  
        }  
    } else if (displayable == FGonoreWt) {  
        if (command == Kembali4) {  
            // write pre-action user code here  
            switchDisplayable(null, getPyWanitaList());  
            // write post-action user code here  
        }  
    } else if (displayable == FKonjungtivitisWt) {  
        if (command == Kembali4) {  
            // write pre-action user code here  
            switchDisplayable(null, getPyWanitaList());  
            // write post-action user code here  
        }  
    } else if (displayable == FLittritisPria) {  
        if (command == Kembali4) {  
            // write pre-action user code here  
            switchDisplayable(null, getPyPriaList());  
            // write post-action user code here  
        }  
    } else if (displayable == FOfaringitisWt) {  
        if (command == Kembali4) {  
            // write pre-action user code here  
            switchDisplayable(null, getPyWanitaList());  
            // write post-action user code here  
        }  
    } else if (displayable == FParauretritisPria) {  
        if (command == Kembali4) {
```



```

        // write pre-action user code here
        switchDisplayable(null, getPyPriaList());
        // write post-action user code here
    }
} else if (displayable == FParauretritisWt) {
    if (command == Kembali4) {
        // write pre-action user code here
        switchDisplayable(null, getPyWanitaList());
        // write post-action user code here
    }
} else if (displayable == FProktitisWt) {
    if (command == Kembali4) {
        // write pre-action user code here
        switchDisplayable(null, getPyWanitaList());
        // write post-action user code here
    }
} else if (displayable == FProstatitisPria) {
    if (command == Kembali4) {
        // write pre-action user code here
        switchDisplayable(null, getPyPriaList());
        // write post-action user code here
    }
} else if (displayable == FSalpingitisWt) {
    if (command == Kembali4) {
        // write pre-action user code here
        switchDisplayable(null, getPyWanitaList());
        // write post-action user code here
    }
} else if (displayable == FServitisWt) {
    if (command == Kembali4) {
        // write pre-action user code here
        switchDisplayable(null, getPyWanitaList());
        // write post-action user code here
    }
} else if (displayable == FTrigonitisPria) {
    if (command == Kembali4) {
        // write pre-action user code here
        switchDisplayable(null, getPyPriaList());
        // write post-action user code here
    }
} else if (displayable == FTysonitisPria) {
    if (command == Kembali4) {
        // write pre-action user code here
        switchDisplayable(null, getPyPriaList());
        // write post-action user code here
    }
} else if (displayable == FUREtritisPria) {
    if (command == Kembali4) {
        // write pre-action user code here
        switchDisplayable(null, getPyPriaList());
        // write post-action user code here
    }
} else if (displayable == FUREtritisWt) {
    if (command == Kembali4) {
        // write pre-action user code here

```

```

        switchDisplayable(null, getPyWanitaList());
        // write post-action user code here
    }
} else if (displayable == FVasdeferentitisPria) {
    if (command == Kembali4) {
        // write pre-action user code here
        switchDisplayable(null, getPyPriaList());
        // write post-action user code here
    }
} else if (displayable == FVesikulitisPria) {
    if (command == Kembali4) {
        // write pre-action user code here
        switchDisplayable(null, getPyPriaList());
        // write post-action user code here
    }
} else if (displayable == MenuList) {
    if (command == Kembali1) {
        // write pre-action user code here
        switchDisplayable(null, getWelcomeScreen());
        // write post-action user code here
    } else if (command == List.SELECT_COMMAND) {
        // write pre-action user code here
        MenuListAction();
        // write post-action user code here
    }
} else if (displayable == PyPriaList) {
    if (command == Kembali2) {
        // write pre-action user code here
        switchDisplayable(null, getMenuList());
        // write post-action user code here
    } else if (command == List.SELECT_COMMAND) {
        // write pre-action user code here
        PyPriaListAction();
        // write post-action user code here
    }
} else if (displayable == PyWanitaList) {
    if (command == Kembali3) {
        // write pre-action user code here
        switchDisplayable(null, getMenuList());
        // write post-action user code here
    } else if (command == List.SELECT_COMMAND) {
        // write pre-action user code here
        PyWanitaListAction();
        // write post-action user code here
    }
} else if (displayable == WelcomeScreen) {
    if (command == exitCommand) {
        // write pre-action user code here
        exitMIDlet();
        // write post-action user code here
    } else if (command == okCommand1) {
        // write pre-action user code here
        switchDisplayable(null, getMenuList());
        // write post-action user code here
    }
}

```

```
    }  
    // write post-action user code here  
  }  
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: exitCommand ">  
/**  
 * Returns an initialized instance of exitCommand component.  
 * @return the initialized component instance  
 */  
public Command getExitCommand() {  
    if (exitCommand == null) {  
        // write pre-init user code here  
        exitCommand = new Command("Exit", Command.EXIT, 0);  
        // write post-init user code here  
    }  
    return exitCommand;  
}  
//</editor-fold>  
//</editor-fold>  
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: WelcomeScreen ">  
/**  
 * Returns an initialized instance of WelcomeScreen component.  
 * @return the initialized component instance  
 */  
public Form getWelcomeScreen() {  
    if (WelcomeScreen == null) {  
        // write pre-init user code here  
        WelcomeScreen = new Form("Welcome", new Item[] { getStringItem(), getImageItem8() });  
        WelcomeScreen.setTicker(getTicker4());  
        WelcomeScreen.addCommand(getExitCommand());  
        WelcomeScreen.addCommand(getOkCommand1());  
        WelcomeScreen.setCommandListener(this);  
        // write post-init user code here  
    }  
    return WelcomeScreen;  
}  
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem ">  
/**  
 * Returns an initialized instance of stringItem component.  
 * @return the initialized component instance  
 */  
public StringItem getStringItem() {  
    if (stringItem == null) {  
        // write pre-init user code here  
        stringItem = new StringItem("", "Selamat datang, pilih OK untuk masuk ke dalam aplikasi");  
        // write post-init user code here  
    }  
    return stringItem;  
}  
}
```

```
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand ">
```

```
/**
```

```
 * Returns an initialized instance of okCommand component.
```

```
 * @return the initialized component instance
```

```
 */
```

```
public Command getOkCommand() {
```

```
    if (okCommand == null) {
```

```
        // write pre-init user code here
```

```
        okCommand = new Command("Ok", Command.OK, 0);
```

```
        // write post-init user code here
```

```
    }
```

```
    return okCommand;
```

```
}
```

```
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: itemCommand ">
```

```
/**
```

```
 * Returns an initialized instance of itemCommand component.
```

```
 * @return the initialized component instance
```

```
 */
```

```
public Command getItemCommand() {
```

```
    if (itemCommand == null) {
```

```
        // write pre-init user code here
```

```
        itemCommand = new Command("Pria", Command.ITEM, 0);
```

```
        // write post-init user code here
```

```
    }
```

```
    return itemCommand;
```

```
}
```

```
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand1 ">
```

```
/**
```

```
 * Returns an initialized instance of okCommand1 component.
```

```
 * @return the initialized component instance
```

```
 */
```

```
public Command getOkCommand1() {
```

```
    if (okCommand1 == null) {
```

```
        // write pre-init user code here
```

```
        okCommand1 = new Command("Ok", Command.OK, 0);
```

```
        // write post-init user code here
```

```
    }
```

```
    return okCommand1;
```

```
}
```

```
//</editor-fold>
```

```
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: MenuList ">
```

```

/**
 * Returns an initialized instance of MenuList component.
 * @return the initialized component instance
 */
public List getMenuList() {
    if (MenuList == null) {
        // write pre-init user code here
        MenuList = new List("Menu Utama", Choice.IMPLICIT);
        MenuList.append("Penyakit Kelamin Pada Pria", getImage1());
        MenuList.append("Penyakit Kelamin Pada Wanita", getImage1());
        MenuList.setTicker(getTicker4());
        MenuList.addCommand(getKembali1());
        MenuList.setCommandListener(this);
        MenuList.setSelectedFlags(new boolean[] { false, false });
        // write post-init user code here
    }
    return MenuList;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Method: MenuListAction ">
/**
 * Performs an action assigned to the selected list element in the MenuList component.
 */
public void MenuListAction() {
    // enter pre-action user code here
    String __selectedString = getMenuList().getString(getMenuList().getSelectedIndex());
    if (__selectedString != null) {
        if (__selectedString.equals("Penyakit Kelamin Pada Pria")) {
            // write pre-action user code here
            switchDisplayable(null, getPyPriaList());
            // write post-action user code here
        } else if (__selectedString.equals("Penyakit Kelamin Pada Wanita")) {
            // write pre-action user code here
            switchDisplayable(null, getPyWanitaList());
            // write post-action user code here
        }
    }
    // enter post-action user code here
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: Kembali1 ">
/**
 * Returns an initialized instance of Kembali1 component.
 * @return the initialized component instance
 */
public Command getKembali1() {
    if (Kembali1 == null) {
        // write pre-init user code here
        Kembali1 = new Command("Back", Command.BACK, 0);
        // write post-init user code here
    }
    return Kembali1;
}
}

```

```
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: tableModel1 ">
/**
 * Returns an initialized instance of tableModel1 component.
 * @return the initialized component instance
 */
public SimpleTableModel getTableModel1() {
    if (tableModel1 == null) {
        // write pre-init user code here
        tableModel1 = new SimpleTableModel(new java.lang.String[][] {
            new java.lang.String[] { "yang paling sering dijumpai adalah uretritis anterior akuta dan
dapat menjalar ke proksimal,selanjutnya mengakibatkan komplikasi lokal,asendens,dan diseminata.",
"" }}, new java.lang.String[] { "Keterangan", "Ciri-ciri" });
        // write post-init user code here
    }
    return tableModel1;
}
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: Kembali2 ">
/**
 * Returns an initialized instance of Kembali2 component.
 * @return the initialized component instance
 */
public Command getKembali2() {
    if (Kembali2 == null) {
        // write pre-init user code here
        Kembali2 = new Command("Back", Command.BACK, 0);
        // write post-init user code here
    }
    return Kembali2;
}
//</editor-fold>
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: PyPriaList ">
/**
 * Returns an initialized instance of PyPriaList component.
 * @return the initialized component instance
 */
public List getPyPriaList() {
    if (PyPriaList == null) {
        // write pre-init user code here
        PyPriaList = new List("Daftar Nama Penyakit Kelamin Pria", Choice.IMPLICIT);
        PyPriaList.append("Uretritis", getImage2());
        PyPriaList.append("Tysonitis", getImage2());
        PyPriaList.append("Parauretritis ", getImage2());
        PyPriaList.append("Littritis", getImage2());
    }
}
//</editor-fold>
```

```

PyPriaList.append("Cowperitis", getImage2());
PyPriaList.append("Prostatitis", getImage2());
PyPriaList.append("Vesikulitis", getImage2());
PyPriaList.append("Vas deferentitis/funikulitis", getImage2());
PyPriaList.append("Epididimitis", getImage2());
PyPriaList.append("Trigonitis", getImage2());
PyPriaList.setTicker(getTicker1());
PyPriaList.addCommand(getKembali2());
PyPriaList.setCommandListener(this);
PyPriaList.setSelectedFlags(new boolean[] { false, false, false, false, false, false, false, false,
false, false });
    // write post-init user code here
    }
    return PyPriaList;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Method: PyPriaListAction ">
/**
 * Performs an action assigned to the selected list element in the PyPriaList component.
 */
public void PyPriaListAction() {
    // enter pre-action user code here
    String __selectedString = getPyPriaList().getString(getPyPriaList().getSelectedIndex());
    if (__selectedString != null) {
        if (__selectedString.equals("Uretritis")) {
            // write pre-action user code here
            switchDisplayable(null, getFUretritisPria());
            // write post-action user code here
        } else if (__selectedString.equals("Tysonitis")) {
            // write pre-action user code here
            switchDisplayable(null, getFTysonitisPria());
            // write post-action user code here
        } else if (__selectedString.equals("Parauretritis ")) {
            // write pre-action user code here
            switchDisplayable(null, getFParauretritisPria());
            // write post-action user code here
        } else if (__selectedString.equals("Littritis")) {
            // write pre-action user code here
            switchDisplayable(null, getFLittritisPria());
            // write post-action user code here
        } else if (__selectedString.equals("Cowperitis")) {
            // write pre-action user code here
            switchDisplayable(null, getFCowperitisPria());
            // write post-action user code here
        } else if (__selectedString.equals("Prostatitis")) {
            // write pre-action user code here
            switchDisplayable(null, getFProstatitisPria());
            // write post-action user code here
        } else if (__selectedString.equals("Vesikulitis")) {
            // write pre-action user code here
            switchDisplayable(null, getFVesikulitisPria());
            // write post-action user code here
        } else if (__selectedString.equals("Vas deferentitis/funikulitis")) {
            // write pre-action user code here

```

```

        switchDisplayable(null, getFEpididimitisPria());
        // write post-action user code here
    } else if (__selectedString.equals("Epididimitis")) {
        // write pre-action user code here
        switchDisplayable(null, getFEpididimitisPria());
        // write post-action user code here
    } else if (__selectedString.equals("Trigonitis")) {
        // write pre-action user code here
        switchDisplayable(null, getFTrigonitisPria());
        // write post-action user code here
    }
}
// enter post-action user code here
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: Kembali3 ">
/**
 * Returns an initalized instance of Kembali3 component.
 * @return the initialized component instance
 */
public Command getKembali3() {
    if (Kembali3 == null) {
        // write pre-init user code here
        Kembali3 = new Command("Back", Command.BACK, 0);
        // write post-init user code here
    }
    return Kembali3;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: PyWanitaList ">
/**
 * Returns an initalized instance of PyWanitaList component.
 * @return the initialized component instance
 */
public List getPyWanitaList() {
    if (PyWanitaList == null) {
        // write pre-init user code here
        PyWanitaList = new List("Daftar Nama Penyakit Kelamin Wanita", Choice.IMPLICIT);
        PyWanitaList.append("Uretritis", getImage2());
        PyWanitaList.append("Parauretritis/Skenitis", getImage2());
        PyWanitaList.append("Servisititis", getImage2());
        PyWanitaList.append("Bartholinitis", getImage2());
        PyWanitaList.append("Salpingitis", getImage2());
        PyWanitaList.append("Proktitis", getImage2());
        PyWanitaList.append("Orofaringitis", getImage2());
        PyWanitaList.append("Konjungtivitis", getImage2());
        PyWanitaList.append("Gonore diseminata", getImage2());
        PyWanitaList.setTicker(getTicker3());
        PyWanitaList.addCommand(getKembali3());
        PyWanitaList.setCommandListener(this);
        PyWanitaList.setSelectedFlags(new boolean[] { false, false, false, false, false, false,
false, false });
        // write post-init user code here
    }
}

```



```

    }
    return PyWanitaList;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Method: PyWanitaListAction ">
/**
 * Performs an action assigned to the selected list element in the PyWanitaList component.
 */
public void PyWanitaListAction() {
    // enter pre-action user code here
    String __selectedString = getPyWanitaList().getString(getPyWanitaList().getSelectedIndex());
    if (__selectedString != null) {
        if (__selectedString.equals("Uretritis")) {
            // write pre-action user code here
            switchDisplayable(null, getFUretritisWt());
            // write post-action user code here
        } else if (__selectedString.equals("Parauretritis/Skenitis")) {
            // write pre-action user code here
            switchDisplayable(null, getFParauretritisWt());
            // write post-action user code here
        } else if (__selectedString.equals("Servisititis")) {
            // write pre-action user code here
            switchDisplayable(null, getFServisititisWt());
            // write post-action user code here
        } else if (__selectedString.equals("Bartholinitis")) {
            // write pre-action user code here
            switchDisplayable(null, getFBartholinitisWt());
            // write post-action user code here
        } else if (__selectedString.equals("Salpingitis")) {
            // write pre-action user code here
            switchDisplayable(null, getFSalpingitisWt());
            // write post-action user code here
        } else if (__selectedString.equals("Proktitis")) {
            // write pre-action user code here
            switchDisplayable(null, getFProktitisWt());
            // write post-action user code here
        } else if (__selectedString.equals("Orofaringitis")) {
            // write pre-action user code here
            switchDisplayable(null, getFOrofaringitisWt());
            // write post-action user code here
        } else if (__selectedString.equals("Konjungtivitis")) {
            // write pre-action user code here
            switchDisplayable(null, getFKonjungtivitisWt());
            // write post-action user code here
        } else if (__selectedString.equals("Gonore disseminata")) {
            // write pre-action user code here
            switchDisplayable(null, getFGonoreWt());
            // write post-action user code here
        }
    }
    // enter post-action user code here
}
//</editor-fold>

```

```

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: Kembali4 ">
/**
 * Returns an initialized instance of Kembali4 component.
 * @return the initialized component instance
 */
public Command getKembali4() {
    if (Kembali4 == null) {
        // write pre-init user code here
        Kembali4 = new Command("Back", Command.BACK, 0);
        // write post-init user code here
    }
    return Kembali4;
}
//</editor-fold>
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FURETRITISPRIA ">
/**
 * Returns an initialized instance of FURETRITISPRIA component.
 * @return the initialized component instance
 */
public Form getFURETRITISPRIA() {
    if (FURETRITISPRIA == null) {
        // write pre-init user code here
        FURETRITISPRIA = new Form("Ciri-ciri Uretritis", new Item[] { getStringItem1(),
getImageItem2() });
        FURETRITISPRIA.setTicker(getTicker2());
        FURETRITISPRIA.addCommand(getKembali4());
        FURETRITISPRIA.setCommandListener(this);
        // write post-init user code here
    }
    return FURETRITISPRIA;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem1 ">
/**
 * Returns an initialized instance of stringItem1 component.
 * @return the initialized component instance
 */
public StringItem getStringItem1() {
    if (stringItem1 == null) {
        // write pre-init user code here
        stringItem1 = new StringItem("", "Rasa gatal, panas dibagian distal uretra disekitar
\norifisium uretra eksternum, kemudian disusul disuria,\npola kisuria keluar duh tubuh dari ujung
uretra yang \nkadang-kadang disertai darah,dan disertai perasaan \nnyeri pada waktu ereksi. Pada
pemeriksaan tampak \norifisium uretra eksternum eritematosa, edematosa,\ndan ektropion. Tampak
duh tubuh yang mukopurulen,\ndan pada beberapa kasus dapat terjadi pembesaran kelenjar \ngetah
bening inguinal unilateral atau bilateral.");
        // write post-init user code here
    }
    return stringItem1;
}
//</editor-fold>

```

```

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: image1 ">
/**
 * Returns an initialized instance of image1 component.
 * @return the initialized component instance
 */
public Image getImage1() {
    if (image1 == null) {
        // write pre-init user code here
        try {
            image1 = Image.createImage("/cog.png");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
        // write post-init user code here
    }
    return image1;
}
//</editor-fold>

```

```

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: image2 ">
/**
 * Returns an initialized instance of image2 component.
 * @return the initialized component instance
 */
public Image getImage2() {
    if (image2 == null) {
        // write pre-init user code here
        try {
            image2 = Image.createImage("/arrow-right.png");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
        // write post-init user code here
    }
    return image2;
}
//</editor-fold>

```

```

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FTysonitisPria ">
/**
 * Returns an initialized instance of FTysonitisPria component.
 * @return the initialized component instance
 */
public Form getFTysonitisPria() {
    if (FTysonitisPria == null) {
        // write pre-init user code here
        FTysonitisPria = new Form("Ciri-ciri Tysonitis", new Item[] { getStringItem2(),
getImageItem3() });
        FTysonitisPria.setTicker(getTicker2());
        FTysonitisPria.addCommand(getKembali4());
        FTysonitisPria.setCommandListener(this);
        // write post-init user code here
    }
}

```

```

        return FTysonitisPria;
    }
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem2 ">
/**
 * Returns an initialized instance of stringItem2 component.
 * @return the initialized component instance
 */
public StringItem getStringItem2() {
    if (stringItem2 == null) {
        // write pre-init user code here
        stringItem2 = new StringItem("", "Rasa gatal, panas dibagian distral uretra disekitar orifisium uretra \neksternum, kemudian disusul disuria, pola kisuria keluar duh tubuh \ndari ujung uretra yang kadang-kadang disertai darah,dan disertai \nperasaan nyeri pada waktu ereksi. Pada pemeriksaan tampak \norifisium uretra eksternum eritematosa, edematosa,dan ektropion. \nTampak duh tubuh yang mukopurulen,dan pada beberapa kasus \ndapat terjadi pembesaran kelenjar getah bening inguinal unilateral \natau bilateral.");
        // write post-init user code here
    }
    return stringItem2;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FParauretritisPria ">
/**
 * Returns an initialized instance of FParauretritisPria component.
 * @return the initialized component instance
 */
public Form getFParauretritisPria() {
    if (FParauretritisPria == null) {
        // write pre-init user code here
        FParauretritisPria = new Form("Ciri-ciri Parauretritis", new Item[] { getStringItem3(), getImageItem4() });
        FParauretritisPria.setTicker(getTicker2());
        FParauretritisPria.addCommand(getKembali4());
        FParauretritisPria.setCommandListener(this);
        // write post-init user code here
    }
    return FParauretritisPria;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem3 ">
/**
 * Returns an initialized instance of stringItem3 component.
 * @return the initialized component instance
 */
public StringItem getStringItem3() {
    if (stringItem3 == null) {
        // write pre-init user code here
        stringItem3 = new StringItem("", "Rasa gatal, panas dibagian distral uretra disekitar orifisium uretra eksternum, \nkemudian disusul disuria, pola kisuria keluar duh tubuh dari ujung uretra \nyang kadang-kadang disertai darah,dan disertai perasaan nyeri \npada waktu ereksi.");
        // write post-init user code here
    }
}

```

```

    }
    return stringItem3;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FLittritisPria ">
/**
 * Returns an initialized instance of FLittritisPria component.
 * @return the initialized component instance
 */
public Form getFLittritisPria() {
    if (FLittritisPria == null) {
        // write pre-init user code here
        FLittritisPria = new Form("Ciri-ciri Littritis", new Item[] { getStringItem4(), getImageItem5()
});
        FLittritisPria.setTicker(getTicker2());
        FLittritisPria.addCommand(getKembali4());
        FLittritisPria.setCommandListener(this);
        // write post-init user code here
    }
    return FLittritisPria;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem4 ">
/**
 * Returns an initialized instance of stringItem4 component.
 * @return the initialized component instance
 */
public StringItem getStringItem4() {
    if (stringItem4 == null) {
        // write pre-init user code here
        stringItem4 = new StringItem("", "Tidak ada gejala khusus, hanya pada urin ditemukan
\nbenang-benang atau butir-butir. Bila salah satu saluran tersumbat, \ndapat terjadi abses folikular.
Didiagnosis dengan uretroskopi.");
        // write post-init user code here
    }
    return stringItem4;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FCowperitisPria ">
/**
 * Returns an initialized instance of FCowperitisPria component.
 * @return the initialized component instance
 */
public Form getFCowperitisPria() {
    if (FCowperitisPria == null) {
        // write pre-init user code here
        FCowperitisPria = new Form("Ciri-ciri Cowperitis", new Item[] { getStringItem5(),
getImageItem6() });
        FCowperitisPria.setTicker(getTicker2());
        FCowperitisPria.addCommand(getKembali4());
        FCowperitisPria.setCommandListener(this);
        // write post-init user code here
    }
}

```

```

    }
    return FCowperitisPria;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem5 ">
/**
 * Returns an initialized instance of stringItem5 component.
 * @return the initialized component instance
 */
public StringItem getStringItem5() {
    if (stringItem5 == null) {
        // write pre-init user code here
        stringItem5 = new StringItem("", "Keluhan berupa nyeri dan adanya benjolan pada daerah
perineum \ndisertai rasa penuh dan panas, nyeri pada waktu defekasi, dan disuria. \nJika tidak diobati
abses akan pecah melalui kulit perineum, uretra, \natau rektum dan mengakibatkan proktitis. ");
        // write post-init user code here
    }
    return stringItem5;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FProstatitisPria ">
/**
 * Returns an initialized instance of FProstatitisPria component.
 * @return the initialized component instance
 */
public Form getFProstatitisPria() {
    if (FProstatitisPria == null) {
        // write pre-init user code here
        FProstatitisPria = new Form("Ciri-ciri Prostatitis", new Item[] { getStringItem6(),
getImageItem7() });
        FProstatitisPria.setTicker(getTicker2());
        FProstatitisPria.addCommand(getKembali4());
        FProstatitisPria.setCommandListener(this);
        // write post-init user code here
    }
    return FProstatitisPria;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem6 ">
/**
 * Returns an initialized instance of stringItem6 component.
 * @return the initialized component instance
 */
public StringItem getStringItem6() {
    if (stringItem6 == null) {
        // write pre-init user code here
        stringItem6 = new StringItem("", "Ditandai dengan perasaan tidak enak pada daerah perineum
\ndan suprapubis, malese, demam, nyeri kencing sampai hematuri, \nspasme otot uretra sehingga terjadi
retensi urin, \ntenesmus ani, \nsulit buang air besar, dan obstipasi.");
        // write post-init user code here
    }
    return stringItem6;
}

```

```

}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FVesikulitisPria ">
/**
 * Returns an initialized instance of FVesikulitisPria component.
 * @return the initialized component instance
 */
public Form getFVesikulitisPria() {
    if (FVesikulitisPria == null) {
        // write pre-init user code here
        FVesikulitisPria = new Form("Ciri-ciri Vesikulitis", new Item[] { getStringItem7() });
        FVesikulitisPria.setTicker(getTicker2());
        FVesikulitisPria.addCommand(getKembali4());
        FVesikulitisPria.setCommandListener(this);
        // write post-init user code here
    }
    return FVesikulitisPria;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem7 ">
/**
 * Returns an initialized instance of stringItem7 component.
 * @return the initialized component instance
 */
public StringItem getStringItem7() {
    if (stringItem7 == null) {
        // write pre-init user code here
        stringItem7 = new StringItem("", "Demam,polakisuria,hematuria terminal,nyeri pada waktu ereksi atau \nejakulasi, dan spasme mengandung darah. Vesikula/seminalis yang \nmembengkak dan keras seperti sosis,memanjang diatas prostat.\n");
        // write post-init user code here
    }
    return stringItem7;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FVasdeferentitisPria ">
/**
 * Returns an initialized instance of FVasdeferentitisPria component.
 * @return the initialized component instance
 */
public Form getFVasdeferentitisPria() {
    if (FVasdeferentitisPria == null) {
        // write pre-init user code here
        FVasdeferentitisPria = new Form("Ciri-ciri Vasdeferentitis", new Item[] { getStringItem8()
});
        FVasdeferentitisPria.setTicker(getTicker2());
        FVasdeferentitisPria.addCommand(getKembali4());
        FVasdeferentitisPria.setCommandListener(this);
        // write post-init user code here
    }
    return FVasdeferentitisPria;
}
}

```

```

//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem8 ">
/**
 * Returns an initialized instance of stringItem8 component.
 * @return the initialized component instance
 */
public StringItem getStringItem8() {
    if (stringItem8 == null) {
        // write pre-init user code here
        stringItem8 = new StringItem("", "Perasaan nyeri pada daerah abdomen bagian bawah pada sisi yang sama.");
        // write post-init user code here
    }
    return stringItem8;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FEpididimitisPria ">
/**
 * Returns an initialized instance of FEpididimitisPria component.
 * @return the initialized component instance
 */
public Form getFEpididimitisPria() {
    if (FEpididimitisPria == null) {
        // write pre-init user code here
        FEpididimitisPria = new Form("Ciri-ciri Epididimitis", new Item[] { getStringItem9() });
        FEpididimitisPria.setTicker(getTicker2());
        FEpididimitisPria.addCommand(getKembali4());
        FEpididimitisPria.setCommandListener(this);
        // write post-init user code here
    }
    return FEpididimitisPria;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem9 ">
/**
 * Returns an initialized instance of stringItem9 component.
 * @return the initialized component instance
 */
public StringItem getStringItem9() {
    if (stringItem9 == null) {
        // write pre-init user code here
        stringItem9 = new StringItem("", "Cairan irigator terlalu panas atau terlalu pekat, instrumentasi yang kasar, \nepididimitis dan tali spermatika membengkak dan teraba panas, juga testis \nsehingga menyerupai hidrokel sekunder. Pada penekanan terasa nyeri sekali, \nbila mengenai kedua epididimitis dapat mengakibatkan sterilitas. ");
        // write post-init user code here
    }
    return stringItem9;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FTrigonitisPria ">

```



```

/**
 * Returns an initialized instance of FTrigonitisPria component.
 * @return the initialized component instance
 */
public Form getFTrigonitisPria() {
    if (FTrigonitisPria == null) {
        // write pre-init user code here
        FTrigonitisPria = new Form("Ciri-ciri Trigosnitis", new Item[] { getStringItem10() });
        FTrigonitisPria.setTicker(getTicker2());
        FTrigonitisPria.addCommand(getKembali4());
        FTrigonitisPria.setCommandListener(this);
        // write post-init user code here
    }
    return FTrigonitisPria;
}
//</editor-fold>

<<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem10 ">
/**
 * Returns an initialized instance of stringItem10 component.
 * @return the initialized component instance
 */
public StringItem getStringItem10() {
    if (stringItem10 == null) {
        // write pre-init user code here
        stringItem10 = new StringItem("", "Trigonitis menimbulkan gejala poliuria, disuria terminal,
dan hematuria.");
        // write post-init user code here
    }
    return stringItem10;
}
//</editor-fold>

<<editor-fold defaultstate="collapsed" desc=" Generated Getter: FUretritisWt ">
/**
 * Returns an initialized instance of FUretritisWt component.
 * @return the initialized component instance
 */
public Form getFUretritisWt() {
    if (FUretritisWt == null) {
        // write pre-init user code here
        FUretritisWt = new Form("Ciri-ciri Uretritis", new Item[] { getStringItem11(),
getImageItem() });
        FUretritisWt.setTicker(getTicker3());
        FUretritisWt.addCommand(getKembali4());
        FUretritisWt.setCommandListener(this);
        // write post-init user code here
    }
    return FUretritisWt;
}
//</editor-fold>

<<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem11 ">
/**
 * Returns an initialized instance of stringItem11 component.

```

```

    * @return the initialized component instance
    */
    public StringItem getStringItem1() {
        if (stringItem1 == null) {
            // write pre-init user code here
            stringItem1 = new StringItem("", "Gejala utama ialah disuria, kadang-kadang
poliuria, \norifisium uretra ekstemum tampak merah, edematosa \ndan ada sekret mukopurulen.");
            // write post-init user code here
        }
        return stringItem1;
    }
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FParauretritisWt ">
/**
 * Returns an initalized instance of FParauretritisWt component.
 * @return the initialized component instance
 */
public Form getFParauretritisWt() {
    if (FParauretritisWt == null) {
        // write pre-init user code here
        FParauretritisWt = new Form("Ciri-ciri Parauretritis", new Item[] { getStringItem12(),
getImageItem1() });
        FParauretritisWt.setTicker(getTicker3());
        FParauretritisWt.addCommand(getKembali4());
        FParauretritisWt.setCommandListener(this);
        // write post-init user code here
    }
    return FParauretritisWt;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem12 ">
/**
 * Returns an initalized instance of stringItem12 component.
 * @return the initialized component instance
 */
public StringItem getStringItem12() {
    if (stringItem12 == null) {
        // write pre-init user code here
        stringItem12 = new StringItem("", "Kelenjar parauretra dapat terkena,tetapi abses jarang
terjadi.");
        // write post-init user code here
    }
    return stringItem12;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FServisititisWt ">
/**
 * Returns an initalized instance of FServisititisWt component.
 * @return the initialized component instance
 */
public Form getFServisititisWt() {
    if (FServisititisWt == null) {

```

```

        // write pre-init user code here
        FServitisWt = new Form("Ciri-ciri Servitis", new Item[] { getStringItem13() });
        FServitisWt.setTicker(getTicker3());
        FServitisWt.addCommand(getKembali4());
        FServitisWt.setCommandListener(this);
        // write post-init user code here
    }
    return FServitisWt;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem13 ">
/**
 * Returns an initialized instance of stringItem13 component.
 * @return the initialized component instance
 */
public StringItem getStringItem13() {
    if (stringItem13 == null) {
        // write pre-init user code here
        stringItem13 = new StringItem("stringItem13", "Rasa nyeri pada punggung bawah, serviks
tampak merah \ndengan erosi dan sekret mukopurulen. Duh tubuh akan terlihat lebih banyak.");
        // write post-init user code here
    }
    return stringItem13;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FBartholinitisWt ">
/**
 * Returns an initialized instance of FBartholinitisWt component.
 * @return the initialized component instance
 */
public Form getFBartholinitisWt() {
    if (FBartholinitisWt == null) {
        // write pre-init user code here
        FBartholinitisWt = new Form("Ciri-ciri Bartholinitis", new Item[] { getStringItem14() });
        FBartholinitisWt.setTicker(getTicker3());
        FBartholinitisWt.addCommand(getKembali4());
        FBartholinitisWt.setCommandListener(this);
        // write post-init user code here
    }
    return FBartholinitisWt;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem14 ">
/**
 * Returns an initialized instance of stringItem14 component.
 * @return the initialized component instance
 */
public StringItem getStringItem14() {
    if (stringItem14 == null) {
        // write pre-init user code here

```

```
        stringItem14 = new StringItem("", "Labium mayor pada sisi yang terkena  
membengkak,\nmerah dan nyeri tekan. Kelenjar bartholin membengkak,\nnterasa nyeri sekali ketika  
penderita berjalan, \npenderita sukar duduk");
```

```
        // write post-init user code here  
    }  
    return stringItem14;  
}  
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FSalpingitisWt ">  
/**
```

```
 * Returns an initialized instance of FSalpingitisWt component.  
 * @return the initialized component instance  
 */  
public Form getFSalpingitisWt() {  
    if (FSalpingitisWt == null) {  
        // write pre-init user code here  
        FSalpingitisWt = new Form("Ciri-ciri Salpingitis", new Item[] { getStringItem15() });  
        FSalpingitisWt.setTicker(getTicker3());  
        FSalpingitisWt.addCommand(getKembali4());  
        FSalpingitisWt.setCommandListener(this);  
        // write post-init user code here  
    }  
    return FSalpingitisWt;  
}  
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem15 ">  
/**
```

```
 * Returns an initialized instance of stringItem15 component.  
 * @return the initialized component instance  
 */  
public StringItem getStringItem15() {  
    if (stringItem15 == null) {  
        // write pre-init user code here  
        stringItem15 = new StringItem("", "Nyeri pada daerah abdomen bawah, duh tubuh  
vagina,\nndisuria, dan menstruasi yang tidak teratur atau abnormal. ");  
        // write post-init user code here  
    }  
    return stringItem15;  
}  
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FProktitisWt ">  
/**
```

```
 * Returns an initialized instance of FProktitisWt component.  
 * @return the initialized component instance  
 */  
public Form getFProktitisWt() {  
    if (FProktitisWt == null) {  
        // write pre-init user code here  
        FProktitisWt = new Form("Ciri-ciri Proktitis", new Item[] { getStringItem16() });  
        FProktitisWt.setTicker(getTicker3());  
        FProktitisWt.addCommand(getKembali4());  
        FProktitisWt.setCommandListener(this);  
    }  
    return FProktitisWt;  
}  
//</editor-fold>
```

```

        // write post-init user code here
    }
    return FProktitisWt;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem16 ">
/**
 * Returns an initialized instance of stringItem16 component.
 * @return the initialized component instance
 */
public StringItem getStringItem16() {
    if (stringItem16 == null) {
        // write pre-init user code here
        stringItem16 = new StringItem("", "Terasa seperti terbakar pada daerah anus dan pada
\npemeriksaan tampak mukosa eritematosa, edematosa, \ndan tertutup pus mukopurulen.");
        // write post-init user code here
    }
    return stringItem16;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FOrofaringitisWt ">
/**
 * Returns an initialized instance of FOrofaringitisWt component.
 * @return the initialized component instance
 */
public Form getFOrofaringitisWt() {
    if (FOrofaringitisWt == null) {
        // write pre-init user code here
        FOrofaringitisWt = new Form("Ciri-ciri Orofaringitis", new Item[] { getStringItem17() });
        FOrofaringitisWt.setTicker(getTicker3());
        FOrofaringitisWt.addCommand(getKembali4());
        FOrofaringitisWt.setCommandListener(this);
        // write post-init user code here
    }
    return FOrofaringitisWt;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem17 ">
/**
 * Returns an initialized instance of stringItem17 component.
 * @return the initialized component instance
 */
public StringItem getStringItem17() {
    if (stringItem17 == null) {
        // write pre-init user code here
        stringItem17 = new StringItem("", "Keluhan sering bersifat asimtomatik, orofaring \ntampak
\neksudat mukopurulen yang ringan atau sedang.");
        // write post-init user code here
    }
    return stringItem17;
}
//</editor-fold>

```

```

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FKonjungtivitisWt ">
/**
 * Returns an initialized instance of FKonjungtivitisWt component.
 * @return the initialized component instance
 */
public Form getFKonjungtivitisWt() {
    if (FKonjungtivitisWt == null) {
        // write pre-init user code here
        FKonjungtivitisWt = new Form("Ciri-ciri Kunjungtivitis", new Item[] { getStringItem18() });
        FKonjungtivitisWt.setTicker(getTicker3());
        FKonjungtivitisWt.addCommand(getKembali4());
        FKonjungtivitisWt.setCommandListener(this);
        // write post-init user code here
    }
    return FKonjungtivitisWt;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem18 ">
/**
 * Returns an initialized instance of stringItem18 component.
 * @return the initialized component instance
 */
public StringItem getStringItem18() {
    if (stringItem18 == null) {
        // write pre-init user code here
        stringItem18 = new StringItem("", "Keluhannya berupa fotofobi, konjungtiva bengkak dan
merah \ndan keluar eksudat mukopurulen. Bila tidak diobati dapat \nberakibat terjadinya ulkus
kornea, panoftalmitis sampai timbul kebutaan.");
        // write post-init user code here
    }
    return stringItem18;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: FGonoreWt ">
/**
 * Returns an initialized instance of FGonoreWt component.
 * @return the initialized component instance
 */
public Form getFGonoreWt() {
    if (FGonoreWt == null) {
        // write pre-init user code here
        FGonoreWt = new Form("Ciri-ciri Gonore", new Item[] { getStringItem19() });
        FGonoreWt.setTicker(getTicker3());
        FGonoreWt.addCommand(getKembali4());
        FGonoreWt.setCommandListener(this);
        // write post-init user code here
    }
    return FGonoreWt;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: stringItem19 ">

```

```

/**
 * Returns an initialized instance of stringItem19 component.
 * @return the initialized component instance
 */
public StringItem getStringItem19() {
    if (stringItem19 == null) {
        // write pre-init user code here
        stringItem19 = new StringItem("", "Gejala yang timbul dapat berupa artritis (terutama mono
arthritis) \nmiokarditis,endokarditis, perikarditis, meningitis, dan dermatitis.");
        // write post-init user code here
    }
    return stringItem19;
}
//</editor-fold>

```

```

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: imageItem ">
/**
 * Returns an initialized instance of imageItem component.
 * @return the initialized component instance
 */
public ImageItem getImageItem() {
    if (imageItem == null) {
        // write pre-init user code here
        imageItem = new ImageItem("imageItem", getImage9(), ImageItem.LAYOUT_DEFAULT,
"<Missing Image>");
        // write post-init user code here
    }
    return imageItem;
}
//</editor-fold>

```

```

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: image7 ">
/**
 * Returns an initialized instance of image7 component.
 * @return the initialized component instance
 */
public Image getImage7() {
    if (image7 == null) {
        // write pre-init user code here
        try {
            image7 = Image.createImage("/5.jpg");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
        // write post-init user code here
    }
    return image7;
}
//</editor-fold>

```

```

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: image4 ">
/**
 * Returns an initialized instance of image4 component.

```

```

* @return the initialized component instance
*/
public Image getImage4() {
    if (image4 == null) {
        // write pre-init user code here
        try {
            image4 = Image.createImage("/2.jpg");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
        // write post-init user code here
    }
    return image4;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: image3 ">
/**
 * Returns an initialized instance of image3 component.
 * @return the initialized component instance
 */
public Image getImage3() {
    if (image3 == null) {
        // write pre-init user code here
        try {
            image3 = Image.createImage("/1.jpg");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
        // write post-init user code here
    }
    return image3;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: image6 ">
/**
 * Returns an initialized instance of image6 component.
 * @return the initialized component instance
 */
public Image getImage6() {
    if (image6 == null) {
        // write pre-init user code here
        try {
            image6 = Image.createImage("/4.jpg");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
        // write post-init user code here
    }
    return image6;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: image5 ">

```



```

/**
 * Returns an initialized instance of image5 component.
 * @return the initialized component instance
 */
public Image getImage5() {
    if (image5 == null) {
        // write pre-init user code here
        try {
            image5 = Image.createImage("/3.jpg");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
        // write post-init user code here
    }
    return image5;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: image8 ">
/**
 * Returns an initialized instance of image8 component.
 * @return the initialized component instance
 */
public Image getImage8() {
    if (image8 == null) {
        // write pre-init user code here
        try {
            image8 = Image.createImage("/6.jpg");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
        // write post-init user code here
    }
    return image8;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: imageItem2 ">
/**
 * Returns an initialized instance of imageItem2 component.
 * @return the initialized component instance
 */
public ImageItem getImageItem2() {
    if (imageItem2 == null) {
        // write pre-init user code here
        imageItem2 = new ImageItem("Gambar", getImage10(), ImageItem.LAYOUT_DEFAULT,
"<Missing Image>");
        // write post-init user code here
    }
    return imageItem2;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: imageItem3 ">
/**

```

```

    * Returns an initialized instance of imageItem3 component.
    * @return the initialized component instance
    */
    public ImageItem getImageItem3() {
        if (imageItem3 == null) {
            // write pre-init user code here
            imageItem3 = new ImageItem("Gambar", getImage8(), ImageItem.LAYOUT_DEFAULT,
"<Missing Image>");
            // write post-init user code here
        }
        return imageItem3;
    }
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: imageItem4 ">
/**
 * Returns an initialized instance of imageItem4 component.
 * @return the initialized component instance
 */
public ImageItem getImageItem4() {
    if (imageItem4 == null) {
        // write pre-init user code here
        imageItem4 = new ImageItem("Gambar", getImage7(), ImageItem.LAYOUT_DEFAULT,
"<Missing Image>");
        // write post-init user code here
    }
    return imageItem4;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: imageItem5 ">
/**
 * Returns an initialized instance of imageItem5 component.
 * @return the initialized component instance
 */
public ImageItem getImageItem5() {
    if (imageItem5 == null) {
        // write pre-init user code here
        imageItem5 = new ImageItem("Gambar", getImage6(), ImageItem.LAYOUT_DEFAULT,
"<Missing Image>");
        // write post-init user code here
    }
    return imageItem5;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: imageItem6 ">
/**
 * Returns an initialized instance of imageItem6 component.
 * @return the initialized component instance
 */
public ImageItem getImageItem6() {
    if (imageItem6 == null) {
        // write pre-init user code here

```

```

        imageItem6 = new ImageItem("Gambar", getImage5(), ImageItem.LAYOUT_DEFAULT,
"<Missing Image>");
        // write post-init user code here
    }
    return imageItem6;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: imageItem1 ">
/**
 * Returns an initialized instance of imageItem1 component.
 * @return the initialized component instance
 */
public ImageItem getImageItem1() {
    if (imageItem1 == null) {
        // write pre-init user code here
        imageItem1 = new ImageItem("imageItem1", null, ImageItem.LAYOUT_DEFAULT,
"<Missing Image>");
        // write post-init user code here
    }
    return imageItem1;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: image9 ">
/**
 * Returns an initialized instance of image9 component.
 * @return the initialized component instance
 */
public Image getImage9() {
    if (image9 == null) {
        // write pre-init user code here
        try {
            image9 = Image.createImage("/7.jpg");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
        // write post-init user code here
    }
    return image9;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: image10 ">
/**
 * Returns an initialized instance of image10 component.
 * @return the initialized component instance
 */
public Image getImage10() {
    if (image10 == null) {
        // write pre-init user code here
        try {
            image10 = Image.createImage("/8.jpg");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
    }
}

```

```

        }
        // write post-init user code here
    }
    return image10;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: imageItem7 ">
/**
 * Returns an initialized instance of imageItem7 component.
 * @return the initialized component instance
 */
public ImageItem getImageItem7() {
    if (imageItem7 == null) {
        // write pre-init user code here
        imageItem7 = new ImageItem("imageItem7", getImage3(),
ImageItem.LAYOUT_DEFAULT, "<Missing Image>");
        // write post-init user code here
    }
    return imageItem7;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: imageItem8 ">
/**
 * Returns an initialized instance of imageItem8 component.
 * @return the initialized component instance
 */
public ImageItem getImageItem8() {
    if (imageItem8 == null) {
        // write pre-init user code here
        imageItem8 = new ImageItem("", getImage11(), ImageItem.LAYOUT_DEFAULT,
"<Missing Image>", Item.PLAIN);
        // write post-init user code here
    }
    return imageItem8;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: image11 ">
/**
 * Returns an initialized instance of image11 component.
 * @return the initialized component instance
 */
public Image getImage11() {
    if (image11 == null) {
        // write pre-init user code here
        try {
            image11 = Image.createImage("/background.jpg");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
        // write post-init user code here
    }
}

```

```

    }
    return image11;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: ticker ">
/**
 * Returns an initialized instance of ticker component.
 * @return the initialized component instance
 */
public Ticker getTicker() {
    if (ticker == null) {
        // write pre-init user code here
        ticker = new Ticker("Daftar Nama Penyakit Kelamin Wanita");
        // write post-init user code here
    }
    return ticker;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: ticker1 ">
/**
 * Returns an initialized instance of ticker1 component.
 * @return the initialized component instance
 */
public Ticker getTicker1() {
    if (ticker1 == null) {
        // write pre-init user code here
        ticker1 = new Ticker("Daftar Nama Penyakit Kelamin Pria");
        // write post-init user code here
    }
    return ticker1;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: ticker2 ">
/**
 * Returns an initialized instance of ticker2 component.
 * @return the initialized component instance
 */
public Ticker getTicker2() {
    if (ticker2 == null) {
        // write pre-init user code here
        ticker2 = new Ticker("Penyakit Kelamin Pria");
        // write post-init user code here
    }
    return ticker2;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: ticker3 ">
/**
 * Returns an initialized instance of ticker3 component.
 * @return the initialized component instance
 */

```

```

public Ticker getTicker3() {
    if (ticker3 == null) {
        // write pre-init user code here
        ticker3 = new Ticker("Penyakit Kelamin Wanita");
        // write post-init user code here
    }
    return ticker3;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: ticker4 ">
/**
 * Returns an initialized instance of ticker4 component.
 * @return the initialized component instance
 */
public Ticker getTicker4() {
    if (ticker4 == null) {
        // write pre-init user code here
        ticker4 = new Ticker("Sistem Informasi Penyakit Kelamin");
        // write post-init user code here
    }
    return ticker4;
}
//</editor-fold>
/**
 * Returns a display instance.
 * @return the display instance.
 */
public Display getDisplay () {
    return Display.getDisplay(this);
}

/**
 * Exits MIDlet.
 */
public void exitMIDlet() {
    switchDisplayable (null, null);
    destroyApp(true);
    notifyDestroyed();
}

/**
 * Called when MIDlet is started.
 * Checks whether the MIDlet have been already started and initialize/starts or resumes the MIDlet.
 */
public void startApp() {
    if (midletPaused) {
        resumeMIDlet ();
    } else {
        initialize ();
        startMIDlet ();
    }
    midletPaused = false;
}
}

```

```
/**
 * Called when MIDlet is paused.
 */
public void pauseApp() {
    midletPaused = true;
}

/**
 * Called to signal the MIDlet to terminate.
 * @param unconditional if true, then the MIDlet has to be unconditionally terminated and all
resources has to be released.
 */
public void destroyApp(boolean unconditional) {
}
}
```

