

**PEMBANGUNAN SISTEM INFORMASI REGISTER KOHORT
BIDAN DESA DAN PEMETAAN KESEHATAN IBU DAN ANAK
MENGUNAKAN POLA PERANCANGAN *FACTORY METHOD***

(Studi Kasus Badan Kesehatan Ibu dan Anak (BKIA)
Puskesmas Kecamatan Tumpang Kabupaten Malang)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Faisol Andi Sefihara

NIM: 125150201111073



INFORMATIKA / ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

PEMBANGUNAN SISTEM INFORMASI REGISTER KOHORT BIDAN DESA
DAN PEMETAAN KESEHATAN IBU DAN ANAK
MENGUNAKAN POLA PERANCANGAN FACTORY METHOD

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Faisal Andi Sefihara
NIM: 125150201111073

Skripsi ini telah diuji dan dinyatakan lulus pada
18 Mei 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Fajar Pradana, S.ST, M.Eng
NIP: 19871121 201504 1 004

Indriati, S.T, M.Kom
NIP: 19831013 201504 2 002

Mengetahui
Ketua Program Studi Informatika / Ilmu Komputer

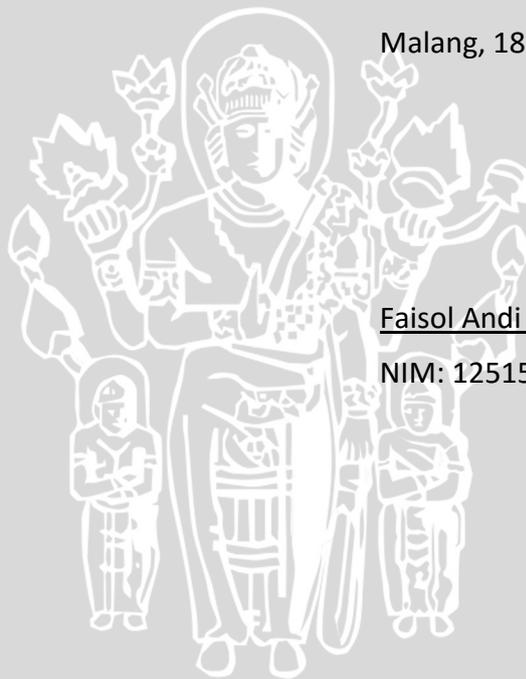
Issa Arwani, S.Kom, M.Sc
NIP: 198330922 201212 1 003

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 18 Mei 2016



Faisol Andi Sefihara

NIM: 125150201111073

ABSTRAK

Kesehatan ibu dan anak merupakan parameter utama kesehatan nasional. Proses pelayanan kesehatan ibu dan anak dimulai dari tingkat bawah yaitu posyandu oleh bidan desa setempat. Sistem administrasi ibu dan anak pada bidan desa masih menggunakan catatan manual yaitu buku register kohort. Register kohort merupakan catatan administrasi rekam medis dari ibu dan anak dalam suatu desa pada kurun waktu tertentu. Buku register kohort terdiri dari register ibu, register bayi, register balita dan register anak prasekolah. Proses administrasi pada empat buku ini dilakukan secara manual dengan ukuran buku register yang sangat besar karena memiliki banyak kolom atau atribut. Selain itu perekapan yang dilakukan bidan desa setiap bulannya juga dilakukan secara manual. Perekapan tersebut terdiri dari lima belas indikator sesuai dengan program nasional PWS-KIA yang nantinya akan dianalisis untuk pemetaan status cakupan sebuah desa. Dari masalah tersebut pada penelitian ini akan dibangun sistem informasi digital yang akan mengelola data register kohort ibu dan anak. Sistem yang bernama SIPKIA (Sistem Informasi dan Pemetaan Kesehatan Ibu dan Anak) ini akan memberikan fitur-fitur otomatis dalam pengelolaan data seperti proses perekapan pada masing-masing indikator cakupan dan pemetaan status cakupan desa tiap indikator pemantauan. SIPKIA dibangun menggunakan pola *perancangan factory method* untuk mewadahi adanya pengembangan sistem, mengingat masih banyak administrasi yang dilakukan oleh bidan desa diluar register kohort seperti administrasi keluarga berencana, imunisasi dan gizi. Pemilihan pola *factory method* sendiri karena komponen *pattern* pada *factory method* memiliki kesamaan dengan objek sistem yang akan dibangun. Komponen dalam *factory method* secara umum terdiri dari *creator* (pabrik) yang akan diimplementasikan pada bidan desa dan komponen *product* yang akan diimplementasikan pada register kohort. Dengan adanya pola dalam SIPKIA diharapkan pengembang dapat lebih mudah untuk melakukan pengembangan sistem yaitu dengan cara mengikuti pola yang ada dan menggunakan ulang bagian *source code* yang telah ada.

Kata kunci: register kohort, indikator cakupan, status cakupan desa, *factory method*.

ABSTRACT

Mothers and children's health are major national health parameters. The service processes of mothers and children's health started from the bottom level, which is Posyandu, by local village midwives. Administrative system of mothers and children by village midwives still using a manual records called cohort registers book. Cohort registers are administrative notes from the medical records of mothers and children in a village at a certain time. Cohort registers book consisted of mother registers, infant registers, toddler registers and preschoolers registers. Administrative process of these books are done manually by the large-sized registry book because it has a lot of columns or attributes. Beside that, the recording conducted monthly by midwives also performed manually. The recording consists of fifteen indicators that matches with the national program PWS-KIA which will be analyzed for village's coverage status mapping. Based on problems above, in this research a digital information system will be built that would manage cohort registers' data of mothers and children. This system, which is called SIPKIA (Sistem Informasi dan Pemetaan Kesehatan Ibu dan Anak), will provide automated features for data management process such as recording process on each coverage indicators and coverage status mapping of villages for every monitoring indicator. SIPKIA was built using design patterns of factory method to facilitate the development of the system, since there are many administrative done by midwives aside the administrative registers cohorts such as family planning, immunization and nutrition administrations. Selection of factory method pattern itself chosen because the factory method pattern components has a similarity with the system object which will be built. Factory method's components generally consisted of creator which will be implemented to midwives and product component which will be implemented to cohort register. With available patterns in SIPKIA, it is expected that the developers can develop system more easily which is by following the existing patterns and reuse parts of existing source code.

Keyword : cohort register, coverage indicators, village's coverage status, factory method.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena atas berkah, rahmat serta hidayah-Nya sehingga penulis dapat menyelesaikan laporan Skripsi dengan judul “Pembangunan Sistem Informasi dan Pemetaan Kesehatan Ibu dan Bayi Menggunakan Pola Perancangan *Factory Method*” dengan baik dan tepat waktu. Keberhasilan tersebut tak lepas dari bantuan dan dukungan dari berbagai pihak baik secara moril maupun materil. Oleh karena itu dalam kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Bapak Fajar Pradana, S.ST, M.Eng selaku dosen pembimbing 1.
2. Ibu Indriati, S.T, M.Kom selaku dosen pembimbing 2.
3. Issa Arwani, S.Kom, M.Sc selaku ketua program studi Informatika.
4. Ibu Khusfiatul, Amd.Keb selaku narasumber utama yaitu Bidan Desa Kidal Kecamatan Tumpang yang telah memberikan informasi pencatatan informasi register kohort dan meminjami buku register kohort.
5. Ibu Sri Mulyani Amd.Keb selaku bidan koordiantor pengelola KIA Kecamatan Tumpang yang telah memberikan informasi sekaligus data mengenai PWS-KIA dan perekapannya.
6. dr.Sri Juliati selaku kepala Puskesmas Kecamatan Tumpang yang telah memberikan izin untuk penelitian di Puskesmas Tumpang.
7. Bapak Achmad Taufik dan Ibu Umi Nurul Hikmah, orang tua penulis dan seluruh keluarga atas dukungan dan do’a yang telah diberikan.
8. Fahmi Alfiani, Amd.Keb. selaku pemberi objek masalah judul skripsi yaitu mengenai register kohort pada bidan desa.
9. Ovan Jaya Wardana yang telah membantu hal non teknis proses skripsi saya, menjadi operator saat presentasi di puskesmas Tumpang, hingga bantuan ketika sidang berlangsung.
10. Rayan Suryadikara yang telah membantu menerjemahkan (*translate*) abstrak kedalam bahasa inggris.
11. Abdul Khoir yang telah membantu proses pelaksanaan seminar hasil dan sidang ujian skripsi.
12. Seluruh pihak yang telah membantu kelancaran skripsi saya yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan laporan ini masih banyak kekurangan baik format laporan maupun isinya. Oleh karena itu, kami mengharapkan saran dan kritik yang membangun dari para pembaca guna perbaikan laporan selanjutnya. Semoga laporan ini dapat memberikan manfaat bagi semua pihak, Aamiin.

Malang, 18 Mei 2016

Faisol Andi Sefihara
sfaisolandi@gmail.com

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	4
1.3 Tujuan	5
1.4 Manfaat	5
1.5 Batasan masalah	5
1.6 Sistematika pembahasan	6
BAB 2 LANDASAN KEPUSTAKAAN	7
2.1 Kajian Pustaka	7
2.2 Register Kohort Bidan Desa	8
2.3 Pemetaan Kesehatan Ibu dan Anak	9
2.3.1 Indikator Pemantauan	10
2.3.2 Status Cakupan Desa	18
2.4 Konsep dasar Sistem Informasi	19
2.5 Model Pengembangan Waterfall	21
2.5.1 Rekayasa Kebutuhan	23
2.5.2 Pengujian Perangkat Lunak	24
2.6 Pola Perancangan (<i>Design Pattern</i>)	27
2.7 Perancangan <i>Factory Pattern</i>	30
2.7.1 <i>Abstract Factory</i>	30
2.7.2 <i>Factory Method</i>	31

2.8 Unified Modelling Language	32
2.8.1 Diagram <i>Use Case</i>	33
2.8.2 Diagram Sequence	35
2.8.3 Diagram Kelas	37
BAB 3 METODOLOGI	40
3.1 Studi Literatur	41
3.2 Pengumpulan Data	42
3.3 Hipotesis	42
3.4 Rekayasa Kebutuhan	42
3.5 Perancangan Sistem	43
3.6 Implementasi Sistem	43
3.7 Pengujian dan Analisa	43
3.8 Pengambilan Kesimpulan	43
BAB 4 REKAYASA KEBUTUHAN	44
4.1 Proses Rekayasa Kebutuhan	45
4.1.1 Elisitasi Kebutuhan	45
4.1.2 Spesifikasi Kebutuhan	48
4.1.3 Validasi dan Verifikasi	51
4.1.4 Manajemen Kebutuhan	53
4.2 Model Kebutuhan	58
4.2.1 Pemodelan Usecase Diagram	58
4.2.2 Pemodelan <i>Sequence Diagram</i>	70
4.2.3 Pemodelan <i>Class Diagram</i>	77
BAB 5 PERANCANGAN DAN IMPLEMENTASI	79
5.1 Perancangan Sistem	79
5.1.1 Perancangan Pola <i>Factory Method</i>	79
5.1.2 Perancangan Arsitektur Sistem	81
5.1.3 Pemodelan Kelas	82
5.1.4 Pemodelan Data	84
5.1.5 Perancangan Komponen	89
5.1.6 Perancangan Antarmuka Pengguna	92
5.2 Implementasi	96

5.2.1 Spesifikasi Sistem	96
5.2.2 Batasan Implementasi.....	97
5.2.3 Implementasi Program.....	97
5.2.4 Implementasi Antarmuka	101
BAB 6 PENGUJIAN DAN ANALISIS.....	109
6.1 Pengujian	109
6.1.1 Pengujian Unit.....	110
6.1.2 Pengujian Integrasi.....	118
6.1.3 Pengujian Validasi	126
6.1.4 Pengujian Pattern.....	134
6.2 Analisis	136
6.2.1 Analisis Hasil Uji Pengujian Unit.....	136
6.2.2 Analisis Hasil Uji Pengujian Integrasi	136
6.2.3 Analisis Hasil Uji Pengujian Validasi.....	136
6.2.4 Analisis Hasil Uji Pengujian Pattern	138
BAB 7 PENUTUP	140
7.1 Kesimpulan.....	140
7.2 Saran	140
DAFTAR PUSTAKA.....	142
LAMPIRAN A FORM VALIDASI DAN VERIFIKASI	144



DAFTAR TABEL

Tabel 2.1 Cakupan-cakupan PWS-KIA dan presentase target	10
Tabel 2.3 Deskripsi macam pola perancangan	27
Tabel 2.4 Pengelompokan pola perancangan.....	29
Tabel 2.5 Komponen <i>use case diagram</i>	34
Tabel 2.6 Komponen <i>sequence diagram</i>	36
Tabel 2.7 Komponen <i>class diagram</i>	38
Tabel 4.1 Identifikasi <i>stakeholder</i>	45
Tabel 4.2 <i>Requirement discovery</i>	46
Tabel 4.3 Daftar kebutuhan fungsional.....	48
Tabel 4.5 Validasi dan verifikasi kebutuhan fungsional	51
Tabel 4.7 Identifikasi user	54
Tabel 4.8 Identifikasi kebutuhan.....	54
Tabel 4.9 Pemetaan kebutuhan fungsional dengan <i>use case</i>	58
Tabel 4.10 Skenario <i>use case</i> lihat kohort ibu	63
Tabel 4.11 Skenario <i>use case</i> tambah kohort ibu	64
Tabel 4.12 Skenario <i>use case</i> ubah kohort ibu	65
Tabel 4.13 Skenario <i>use case</i> hapus kohort ibu	66
Tabel 4.14 Skenario <i>use case</i> lihat rekapan desa.....	67
Tabel 4.15 Skenario <i>use case</i> cetak rekapan desa	68
Tabel 4.16 Skenario <i>use case</i> pemetaan desa	69
Tabel 4.17 Skenario <i>use case</i> cetak pemetaan desa.....	70
Tabel 4.18 Pemetaan <i>use case</i> dengan <i>sequence</i>	71
Tabel 5.1 Komponen perancangan pola <i>factory method</i>	80
Tabel 5.2 Atribut data register ibu	88
Tabel 5.3 Atribut data register bayi	89
Tabel 5.4 Atribut data user	89
Tabel 5.5 Spesifikasi perangkat keras	97
Tabel 5.6 Spesifikasi perangkat lunak	97
Tabel 6.1 Kasus uji algoritma lihat register	114
Tabel 6.2 Kasus uji algoritma lihat register	118

Tabel 6.3 Pengujian integrasi tiap modul lihat register	120
Tabel 6.4 Pengujian integrasi modul utama lihat register	120
Tabel 6.5 Pengujian integrasi tiap modul tambah register.....	123
Tabel 6.6 Pengujian integrasi modul utama tambah register	123
Tabel 6.7 Pengujian lihat register	126
Tabel 6.8 Pengujian tambah register	126
Tabel 6.9 Pengujian ubah register	127
Tabel 6.10 Pengujian hapus register	127
Tabel 6.11 Pengujian lihat pelayanan	127
Tabel 6.12 Pengujian tambah pelayanan.....	128
Tabel 6.13 Pengujian ubah pelayanan	128
Tabel 6.14 Pengujian hapus pelayanan.....	128
Tabel 6.15 Pengujian lihat rekapan desa	129
Tabel 6.16 Pengujian cetak rekapan desa.....	129
Tabel 6.17 Pengujian lihat rekapan kecamatan	129
Tabel 6.18 Pengujian cetak rekapan kecamatan	130
Tabel 6.19 Pengujian pemetaan desa	130
Tabel 6.20 Pengujian cetak rekapan desa.....	130
Tabel 6.21 Pengujian lihat pemetaan kecamatan.....	131
Tabel 6.22 Pengujian cetak pemetaan kecamatan	131
Tabel 6.23 Pengujian perbarui rekapan	131
Tabel 6.24 Pengujian lihat user	132
Tabel 6.25 Pengujian tambah user	132
Tabel 6.26 Pengujian ubah user	132
Tabel 6.27 Pengujian hapus user	133
Tabel 6.28 Pengujian login	133
Tabel 6.29 Pengujian logout.....	133
Tabel 6.31 Ilustrasi penambahan komponen pada sistem	135
Tabel 6.32 Hasil pengujian fungsional	136



DAFTAR GAMBAR

Gambar 2.1 Contoh analisis sederhana dari seluruh cakupan	18
Gambar 2.2 Model pengembangan sistem waterfall	21
Gambar 2.3 Contoh notasi flow graph	25
Gambar 2.4 Struktur pola <i>abstract factory</i>	31
Gambar 2.5 Struktur pola <i>factory method</i>	32
Gambar 2.6 Contoh <i>use case diagram</i>	35
Gambar 2.7 Contoh <i>sequence diagram</i>	37
Gambar 2.8 contoh <i>class diagram</i>	39
Gambar 3.1 Diagram metodologi penelitian	40
Gambar 4.1 Diagram rekayasa kebutuhan	44
Gambar 4.2 Proses bisnis sistem informasi dan pemetaan	47
Gambar 4.3 Identifikasi aktor	54
Gambar 4.4 Diagram <i>use case</i> sistem	62
Gambar 4.5 Diagram <i>sequence</i> lihat kohort ibu	73
Gambar 4.6 Diagram <i>sequence</i> tambah kohort ibu	74
Gambar 4.7 Diagram <i>sequence</i> ubah kohort ibu	75
Gambar 4.8 Diagram <i>sequence</i> hapus kohort ibu	76
Gambar 4.9 Analisis <i>class diagram</i>	77
Gambar 5.1 Diagram perancangan dan implementasi	79
Gambar 5.2 <i>Class diagram</i> perancangan pola <i>factory method</i>	81
Gambar 5.3 Arsitektur sistem	82
Gambar 5.4 <i>Class diagram</i> SIPKIA	83
Gambar 5.5 ERD pengelolaan data register kohort	85
Gambar 5.6 ERD pengelolaan data user sistem	85
Gambar 5.7 ERD pengelolaan data register kohort	86
Gambar 5.8 ERD pengelolaan data user dan pemetaan sipkia	87
Gambar 5.9 Perancangan komponen client	90
Gambar 5.10 Perancangan komponen <i>concrete creator</i>	91
Gambar 5.11 Perancangan komponen <i>concrete product</i>	92
Gambar 5.12 Perancangan komponen rekapan register	92

Gambar 5.13 Antarmuka halaman lihat register	93
Gambar 5.14 Antarmuka tambah register kohort	93
Gambar 5.15 Antarmuka ubah register	94
Gambar 5.16 Antarmuka hapus register.....	95
Gambar 5.17 Halaman lihat pelayanan.....	95
Gambar 5.18 Antarmuka tambah pelayanan.....	96
Gambar 5.19 Kode program komponen client	99
Gambar 5.20 Kode program komponen <i>concrete creator</i>	100
Gambar 5.21 Kode program komponen <i>concrete product</i>	101
Gambar 5.22 Antarmuka halaman lihat register	102
Gambar 5.23 Antarmuka halaman tambah register	103
Gambar 5.24 Antarmuka halaman ubah register	103
Gambar 5.25 Antarmuka halaman hapus register.....	104
Gambar 5.26 Antarmuka halaman lihat pelayanan	104
Gambar 5.27 Antarmuka halaman tambah pelayanan.....	105
Gambar 5.28 Antarmuka halaman ubah pelayanan	105
Gambar 5.29 Antarmuka halaman hapus pelayanan	106
Gambar 5.30 Antarmuka lihat rekapan desa	106
Gambar 5.31 Antarmuka lihat pemetaan desa.....	107
Gambar 5.32 Antarmuka lihat rekapan kecamatan.....	108
Gambar 5.33 Antarmuka lihat pemetaan desa.....	108
Gambar 6.1 Tahap-tahap pengujian dan analisis	109
Gambar 6.2 Pembentukan node algoritma lihat register	111
Gambar 6.3 Flowgraph algoritma lihat register.....	112
Gambar 6.4 Node algoritma tambah register.....	116
Gambar 6.5 Flowgraph algoritma lihat register.....	117
Gambar 6.6 Hirarki bottom-up integration lihat register	119
Gambar 6.7 Tampilan hasil pengujian integrasi modul lihat register.....	122
Gambar 6.8 Hirarki bottom-up integration tambah register.....	122
Gambar 6.9 Tampilan hasil pengujian integrasi modul tambah register	125



DAFTAR LAMPIRAN

LAMPIRAN A FORM VALIDASI DAN VERIFIKASI 144



BAB 1 PENDAHULUAN

1.1 Latar belakang

Kesehatan merupakan hal yang sangat penting dalam kehidupan manusia. Tingkat kesehatan masyarakat berhubungan erat dengan pelayanan kesehatan oleh badan kesehatan nasional. Pelayanan kesehatan yang kurang maksimal mempunyai pengaruh terhadap kesejahteraan masyarakat. Pelayanan kesehatan termasuk administrasi informasi kesehatan berhubungan erat dengan sistem kesehatan nasional (Hovenga dan Grain, 2013). Salah satu upaya yang dapat dilakukan yaitu meningkatkan pelayanan kesehatan ibu dan anak yang merupakan sebuah parameter utama untuk menentukan tingkat kesehatan sebuah negara. Di negara Indonesia pelayanan kesehatan ibu dan anak dimulai dari tingkat bawah atau desa yaitu pada bidan desa atau Pos Pelayanan Terpadu (Posyandu). Setingkat diatas bidan desa yaitu Badan Kesehatan Ibu dan Anak (BKIA) pada puskesmas kecamatan yang bertanggung jawab terhadap data rekapan seluruh desa. Masalah utama sistem pelayanan pada bidan desa masih menggunakan sistem administrasi konvensional. Padahal proses pengarsipan dengan metode manual dalam mencatat data yang berkelanjutan dari riwayat penyakit pasien sangat tidak efektif (Surya ; Wibowo ; Amelia , 2012).

Administrasi pada bidan desa menggunakan sebuah buku register kohort yaitu berisi data riwayat ibu hamil , bayi (*neonatal*), balita, dan anak pra sekolah. Kohort berasal dari kata *cohort* yang memiliki makna sebuah proses pengamatan prospektif terhadap suatu objek penelitian. Sedangkan pada sudut pandang pelayanan kebidanan register kohort merupakan sumber data administrasi pelayanan ibu hamil, ibu nifas, neonatal, bayi dan balita. Tujuan kohort tersebut adalah untuk mengidentifikasi masalah kesehatan ibu dan anak yang terdeteksi di rumah tangga yang dapat dilihat dari data bidan tersebut. Masing-masing buku kohort memiliki atribut berbeda dalam jumlah yang sangat banyak, sehingga buku kohort memiliki ukuran yang besar. Padahal proses pelayanan kesehatan Ibu dan anak dalam suatu desa tidak berlangsung pada pustu (puskesmas pembantu) desa namun juga pada Pos pelayanan Terpadu (posyandu) terkait. Hal ini tentu tidak efektif karena administrasi yang harus menggunakan sebuah buku besar yang harus dibawa kemana-mana.

Proses pengisian registrasi kohort dilakukan oleh bidan desa ketika ada kegiatan posyandu berlangsung. Buku kohort memiliki ukuran yang besar dan proses memasukkan datanya cukup rumit karena harus mencari baris ibu atau baris bayi yang akan dilayani, setelah itu memasukkan datanya menggunakan kode-kode tertentu pada kolom yang sangat kecil. Sedangkan buku kohort sendiri ada empat macam buku berukuran A3 yang harus dibuka pada meja posyandu. Padahal setiap kegiatan posyandu selalu terjadi antrian panjang ibu-ibu yang akan dilakukan pelayanan. Karena dituntut proses pelayanan dan registrasi yang cepat serta benar, seorang bidan desa melakukan registrasi pelayanan tidak dibuku

kohort melainkan pada buku bantu yang dibuatnya sendiri. Buku register kohort harus berisi data yang benar karena data tersebut merupakan acuan bidan desa untuk membuat sebuah laporan, sehingga penulisan pada buku kohort harus hati-hati dan tidak bisa dilakukan dalam keadaan mendesak seperti ketika pelayanan dengan antrian panjang. Setelah proses pelayanan posyandu selesai baru bidan desa melakukan registrasi ulang dari buku bantu tersebut ke semua buku register kohort. Dari sini saja proses registrasi bidan dilakukan dua kali pada empat buku berukuran besar yang tentu sangat tidak efektif.

Proses administrasi pada buku register kohort tidak berhenti pada tingkat bidan desa saja. Pada akhir bulan buku register kohort harus direkap untuk menjadi laporan bulanan pada BKIA puskesmas kecamatan. Perekapan laporan tersebut diambil dari empat buku register kohort. Perekapan tersebut ada lima belas aspek yang dinilai dengan rumus ketentuan tersendiri. Sekali lagi seorang bidan harus melakukan kegiatan yang cukup menyita waktu yang lama karena harus merekap data dari empat buku yang berukuran besar. Karena banyaknya laporan yang harus disiapkan oleh seorang bidan desa untuk puskesmas kecamatan akhirnya sering terjadi keterlambatan laporan kepada BKIA setempat.

Laporan tiap bulan tersebut akan dikalkulasi dalam setahun oleh BKIA kecamatan sebagai acuan pemetaan kesuksesan program pemerintah yaitu Pemantauan Wilayah Setempat Kesehatan Ibu dan Anak (PWS-KIA) (DepKes, 2010). Ada lima belas aspek yang akan dinilai untuk memetakan sebuah desa kedalam status cakupan yaitu termasuk dalam status cakupan baik, kurang, cukup, atau jelek. Melihat hal tersebut perlu adanya sistem pemetaan kesehatan ibu dan bayi berdasarkan status cakupan sebuah desa. Sistem pemetaan ini dapat digunakan oleh BKIA untuk mengetahui desa atau kelurahan mana yang paling memerlukan perhatian dan tindak lanjut yang harus dilakukan, seperti perlu dibuat program penyuluhan atau sosialisasi terhadap desa atau kelurahan yang memiliki status cakupan yang jelek atau yang lainnya.

Solusi yang dapat diambil dari permasalahan tersebut adalah menciptakan sebuah sistem berbasis komputer berupa sistem informasi dan pemetaan kesehatan ibu. Untuk membangun sebuah sistem informasi membutuhkan sebuah model pengembangan perangkat lunak atau yang sering disebut dengan SDLC (System Development Life Cycle). SDLC adalah proses membangun atau mengembangkan perangkat lunak menggunakan metodologi-metodologi yang digunakan oleh orang-orang sebelumnya untuk membangun sebuah perangkat lunak (Rosa & Shalahuddin, 2013). SDLC memiliki beberapa model dalam penerapan tahapan prosesnya. Setiap model SDLC memiliki kelebihan dan kelemahan masing-masing. Beberapa model dasar SDLC adalah model waterfall, model prototype, model iteratif, model spiral, incremental development, dan reuse-oriented development (Rosa & Shalahuddin, 2013). Dalam penelitian ini menggunakan model pengembangan waterfall (air terjun). Model waterfall memiliki kegiatan proses spesifikasi dasar, pengembangan, validasi, dan evolusi. Beberapa kegiatan tersebut terdiri fase spesifikasi persyaratan, perancangan

perangkat lunak, implementasi, pengujian, dan sebagainya (Sommerville, 2011). Model waterfall lebih cocok digunakan ketika kebutuhan pelanggan telah dipahami dengan jelas dan kemungkinan perubahan kebutuhan selama pembangunan perangkat lunak sangat kecil (Rosa & Shalahuddin, 2013).

Dari beberapa fase dalam model waterfall fase perancangan merupakan hal yang penting dalam SDLC karena fase tersebut merupakan jembatan fase analisis kebutuhan dan fase implementasi. Fase desain berhubungan dengan desain arsitektur perangkat lunak. Fase perancangan merupakan bagian tersulit dalam membangun suatu aplikasi (Kusumaningrum; Kusumo & Widowati, 2008). Beberapa hambatan dalam perancangan adalah mendesain spesifikasi yang ada secara efektif yang mampu mengatasi perubahan dan spesifikasi dimasa yang akan datang, sedangkan spesifikasi dimasa datang kurang bisa dipahami dan hal ini merupakan hal sering terjadi. Selain itu usaha menggunakan kembali (*reuse*) desain yang ada sering melibatkan komponen-komponen yang sebenarnya tidak bisa digunakan ulang (*not-reuse*) dan yang paling penting pengembang sering lebih melihat pada sisi *coding*, tidak melihat pada sisi kekuatan desain. Kesimpulan yang dapat diambil adalah kesulitan yang sering dihadapi oleh *programmer* atau *designer* adalah bagaimana membuat desain yang dapat bersifat reusable dan flexible (Kusumaningrum; Kusumo & Widowati, 2008). Dari masalah mendesain tersebut dapat diselesaikan dengan konsep pola pola perancangan (*design pattern*) yang merupakan desain perangkat lunak yang mengacu solusi umum yang dapat digunakan kembali atau berulang-ulang untuk menyelesaikan masalah umum atau khusus.

Dalam penelitian ini register kohort dapat didesain dengan konsep pola perancangan untuk mengantisipasi adanya pengembangan dalam spesifikasi. Beberapa metode pola perancangan yaitu pola observer, factory, facade, adapter, dan decorator (Freeman dkk, 2004). Komponen baik subjek dan objek dalam penelitian ini adalah bidan desa sekecamatan dan register kohort ibu, bayi, balita dan apras. Komponen tersebut cocok dengan komponen pada pola *factory* yaitu terdiri *creator* yang dapat diimplementasikan pada bidan desa dan komponen *product* yang dapat diimplementasikan pada register kohort. Maka dari itu pada penelitian ini menggunakan perancangan *factory pattern* karena komponen dalam pola tersebut memiliki kecocokan dengan objek sistem yang dibangun.

Penelitian mengenai sistem register kohort masih sedikit dilakukan, namun ada penelitian sebelumnya yang dilakukan oleh Yendi Irawan (2010) dengan judul "Sistem Informasi Pengolahan Data Pasien Kohort Ibu Pada Bagian Kebidanan di Puskesmas Garuda Bandung". Penelitian ini fokus pada administrasi register ibu pada cakupan K4 yaitu rekam medis dari awal ibu hamil hingga proses kelahiran. Sistem ini dibangun dengan aplikasi Microsoft Visual Basic dan menggunakan database Sql Server 2000. Penelitian lain mengenai objek kebidanan telah dilakukan oleh Khoiri (2008) dengan judul "Pengembangan Sistem Informasi Posyandu Guna Mendukung Surveilans Kesehatan Ibu dan Anak Berbasis Masyarakat Pada Desa Siaga (Studi Kasus Di Kelurahan Manisrejo Kecamatan

Taman Kota Madiun Provinsi Jawa Timur)”. Sistem tersebut membangun sistem informasi posyandu secara digital. Sistem ini dibangun dengan bahasa PHP dan database MySQL. Untuk penelitian mengenai metode factory pattern pernah dilakukan oleh Kusumaningrum (2008) yaitu menggunakan abstract factory pattern yang telah dibuktikan dapat menjamin bagusnya tingkat reusability perangkat lunak. Dari berbagai penelitian tersebut akan dilakukan kajian untuk perbandingan dan mengadopsi bagian-bagian tertentu untuk mencapai sistem yang lebih sempurna.

Dari semua paparan diatas maka sebuah catatan medis berupa sistem informasi berbasis web yang dapat memenuhi kebutuhan administrasi bidan desa sebagai rekap medis dan pelaporan kepada BKIA di kecamatan perlu dibangun. Sistem tersebut bernama Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method* atau SIPKIA (Sistem Informasi dan Pemetaan Kesehatan Ibu dan Anak). Sistem informasi data pasien dan rekam medis tersebut bertujuan untuk memberikan informasi yang diperlukan sehingga keputusan yang tepat dapat diambil untuk membantu kinerja bidan desa dalam mendukung program pemerintah agar dapat dilaksanakan secara efektif dan efisien (Imbar dan Kurniawan, 2012).

1.2 Rumusan masalah

Berdasarkan paparan latar belakang tersebut, secara umum masalah yang akan dibahas dalam skripsi ini adalah bagaimana hasil pembangunan Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method* di Kecamatan Tumpang Kabupaten Malang. Secara khusus, permasalahan yang akan dikaji dalam skripsi ini, antara lain :

1. Bagaimana hasil rekayasa kebutuhan sistem dalam pembangunan Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method*?
2. Bagaimana hasil perancangan factory pattern dalam pembangunan Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method*?
3. Bagaimana hasil implementasi Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method* dalam rangka meningkatkan mutu pelayanan kesehatan?
4. Bagaimana hasil pengujian Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method*?

1.3 Tujuan

Tujuan yang ingin dicapai dari penulisan skripsi ini, antara lain :

1. Mengetahui dan melakukan proses administrasi informasi kesehatan anak dan ibu serta pemetaan kesehatan ibu terhadap kesehatan bayi secara digital.
2. Merancang dan membangun Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method*
3. Menguji Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method*

1.4 Manfaat

Pembuatan Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method* ini diharapkan dapat bermanfaat bagi :

1. Membantu seluruh bidan desa di Kecamatan Tumpang dalam melakukan proses administrasi informasi kesehatan ibu dan anak dengan lebih mudah, efektif dan efisien.
2. Membantu seluruh bidan desa di Kecamatan Tumpang Kabupaten Malang dalam merekap laporan informasi kesehatan bidan desa
3. Membantu Badan Kesehatan Ibu dan Anak (BKIA) Puskesmas Kecamatan Tumpang Kabupaten Malang dalam merekap laporan informasi kesehatan bidan desa
4. Membantu Badan Kesehatan Ibu dan Anak (BKIA) Puskesmas Kecamatan Tumpang Kabupaten Malang dalam melihat pemetaan kesehatan ibu terhadap kesehatan bayi pada sebuah desa sebagai acuan perancangan program BKIA
5. Penelitian ini dapat digunakan untuk pengembangan penelitian selanjutnya terutama dalam hal sistem informasi kesehatan ibu dan anak pada bidan desa.

1.5 Batasan masalah

Agar permasalahan yang dirumuskan lebih terfokus, penelitian dalam skripsi ini dibatasi dalam hal :

1. Penelitian dilakukan di Kecamatan Tumpang Kabupaten Malang.
2. Objek penelitian merupakan bidan desa dan Badan Kesehatan Ibu dan Anak (BKIA) puskesmas Kecamatan Tumpang Kabupaten Malang
3. Parameter pemetaan adalah program nasional PWS-KIA (Pemantauan Wilayah Setempat- Kesehatan Ibu dan Anak)
4. Pembangunan Sistem Informasi menggunakan pola perancangan *factory method*

1.6 Sistematika pembahasan

Penulisan skripsi ini meliputi beberapa bab sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini berisi tentang latar belakang rumusan masalah, batasan masalah, tujuan , manfaat, sistematika penulisan dalam membangun Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory method* dan rencana serta jadwal penelitian.

BAB II : LANDASAN KEPUSTAKAAN

Bab ini berisi tentang penjelasan teori – teori sebagai referensi dalam pembuatan sistem seperti template, platform dan metode yang digunakan yaitu pola perancangan *factory pattern*.

BAB III : METODOLOGI PENELITIAN

Berisi metodologi penelitian pengerjaan skripsi yang mencakup studi literatur, pengumpulan data, rekayasa kebutuhan dan perancangan, implementasi, pengujian dan analisis, serta pengambilan kesimpulan.

BAB IV : REKAYASA KEBUTUHAN

Membahas tentang bagaimana menganalisis kebutuhan Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory method* yang mencakup elisitasi, spesifikasi, validasi dan verifikasi dan manajemen kebutuhan serta pemodelan hasil rekayasa kebutuhan.

BAB V : PERANCANGAN DAN IMPLEMENTASI

Berisi perancangan dan implementasi yang dilakukan berdasarkan kebutuhan sistem. Bab ini berisi perancangan sistem yang terdiri dari perancangan pola, perancangan arsitektur, pemodelan kelas, pemodelan data, perancangan komponen, perancangan antarmuka dan implementasi yang terdiri dari spesifikasi sistem, batasan implementasi, implementasi program dan implementasi antarmuka.

BAB VI : PENGUJIAN DAN ANALISIS

Menjelaskan proses pengujian sistem serta analisis yang dibuat berdasarkan pengujian.

BAB VII: PENUTUP

Berisi kesimpulan dari pembahasan bab – bab yang telah disampaikan dan saran terhadap desain aplikasi yang telah ditawarkan.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada penelitian pertama dengan yang dilakukan oleh Yendi Irawan (2010) dengan judul “Sistem Informasi Pengolahan Data Pasien Kohort Ibu Pada Bagian Kebidanan di Puskesmas Garuda Bandung”. Penelitian ini fokus pada administrasi register ibu pada cakupan K4 yaitu rekam medis dari awal ibu hamil hingga proses kelahiran. sistem informasi ini bertujuan untuk mengelola data-data pasien yang mendaftar untuk berobat pada periode waktu tertentu. Dalam kegiatan ini sistem informasi yang dirancang sangatlah penting karena untuk mencegah terjadinya kesalahan prosedur dalam pelaksanaan pendaftaran pasien pengolahan data dapat dilakukan dengan sebaik-baiknya. Sistem informasi pendaftaran pasien yang digunakan pada puskesmas babatan bandung dalam pengolahan dan penyimpanan datanya masih bersifat manual belum menggunakan software aplikasi-aplikasi khusus yang menangani proses pendaftaran pasien dan rekam medik, oleh karena itu diperlukan adanya suatu sistem informasi berbasis komputersasi khususnya dibagian tata usaha dengan mengembangkan sistem informasi rakam medik pasien guna mempermudah dalam mengolahan data mulai rekam medik pasien sampai pada tahap pembuatan laporan data pasien dan laporan rekam medik. Sistem ini dibangun dengan aplikasi Microsoft Visual Basic dan menggunakan database Sql Server 2000.

Pada penelitian yang kedua dengan judul “Pengembangan Sistem Informasi Posyandu Guna Mendukung Surveilans Kesehatan Ibu dan Anak Berbasis Masyarakat Pada Desa Siaga (Studi Kasus Di Kelurahan Manisrejo Kecamatan Taman Kota Madiun Provinsi Jawa Timur)” telah dilakukan oleh Abu Khoiri. Sistem ini secara garis besar membangun sistem informasi digital dari sistem informasi posyandu yang telah ada. Sistem diawali dari kader posyandu melakukan registrasi kesehatan ibu dan anak menggunakan instrumen Sistem Informasi Posyandu (SIP). Data yang disimpan dalam sistem antara lain hasil timbang bayi, imunisasi, tingkat risiko ibu hamil, status gizi anak ,imunisasi TT, daftar ibu dan anak yang hidup dan mati, serta daftar hadir petugas pada saat layanan Posyandu. Subyek penelitian terdiri dari aktor end user (kader Posyandu, Pokja IV TP PKK, petugas gizi Puskesmas dan bidan wilayah) dan supporting end user (Ketua Desa Siaga dan Kepala Kelurahan). Sistem ini dikembangkan dengan bahasa pemrograman PHP dan data base MySQL. (Khoiri, 2008).

Penelitian ketiga memiliki kesamaan metode yang digunakan yaitu menggunakan perancangan factory pattern. Penelitian ini mengimplementasikan refactoring untuk mendapatkan suatu desain yang terstruktur, sehingga akan didapatkan sebuah kesamaan pola atau struktur dengan salah satu pattern yang terdapat pada GoF (Gang of Four). Pattern yang akan dipakai adalah Abstract Factory design pattern, yang menyediakan sebuah interface untuk membangun keluarga dari objek yang berhubungan tanpa menspesifikasikan konkret kelasnya.

Dalam penentuan ada tidaknya kesamaan pola desain yang diperoleh tersebut dengan pola desain pada Abstract Factory pattern adalah ditinjau dari adanya kemiripan ataupun kesamaan pada struktur notasi interaksi antara kelas dan objek yang terjadi. Meskipun nantinya akan diperoleh sebuah 'implementation note' yang berbeda dengan Abstract Factory pattern (Kusumaningrum; Kusumo & Widowati, 2008).

Penelitian tentang merancang dan membangun sistem informasi yang ditujukan kepada bidan desa masih sedikit dilakukan. Beberapa penelitian sebelumnya perlu dilakukan sebuah pengembangan sistem untuk proses optimasi pelayanan, terutama belum ada yang membahas tentang pemetaan kesehatan ibu terhadap kesehatan bayi. Berdasarkan penelitian yang telah dilakukan penulis dapat menyimpulkan bahwa memang dibutuhkan sebuah sistem informasi pelayanan kesehatan ibu dan anak pada bidan desa. Mengingat kesehatan anak dan ibu sangat menentukan tingkat kesehatan sebuah desa dan negara. Pada penelitian ini penulis mencoba menggabungkan bagian-bagian beberapa penelitian sebelumnya. Pada penelitian pertama yang dilakukan oleh Yendi Irawan (2010) penulis bermaksud mengembangkan objek penelitian yang pada awalnya hanya terfokus pada cakupan ibu hamil yaitu atau istilah medisnya cakupan K4 kini penulis berusaha mengembangkan menjadi lima belas cakupan yang berasal dari register ibu, bayi, balita dan apras. Sedangkan pada penelitian kedua yang dilakukan oleh Khoiri (2008) penulis mengadopsi sistem yang berbasis web dengan menggunakan php dan data base mySql. Selain itu end user bidan desan dan kader posyandu juga diterapkan pada sistem informasi yang akan dibangun. Selanjutnya metode yang digunakan mengadopsi penelitian yang dilakukan oleh kusumaningrum (2008) yaitu menggunakan perancangan factory pattern. Dalam penelitian tersebut abstract factory pattern telah dibuktikan dapat menjamin bagusnya tingkat reusability perangkat lunak dan terdapat kecenderungan untuk menghasilkan nilai coupling dan cohesion suatu kelas yang mendukung atau meningkatkan reusability perangkat lunak. Oleh karena itu pada penelitian ini diharapkan juga akan mencapai reusability perangkat lunak yang baik.

2.2 Register Kohort Bidan Desa

Kesehatan sangat penting untuk kehidupan dinegara Indonesia, namun sayangnya kurangnya pelayanan kesehatan mempunyai pengaruh terhadap kesejahteraan masyarakat dalam bidang kesehatan. Informasi kesehatan sangat berpengaruh terhadap pemberian pelayanan kesehatan, sehingga sistem informasi kesehatan berhubungan erat dari sistem kesehatan nasional. Sistem Informasi digunakan untuk mengumpulkan, memproses, dokumentasi, menyimpan dan melaporkan informasi dari kesehatan masyarakat. Pada tahun 1991 Institute of Medicine Amerika Serikat mencatat bahwa catatan pasien berbasis komputer adalah teknologi penting untuk kesehatan. Selama ini pemerintah, organisasi kesehatan dan pengembang perangkat lunak berusaha mengadopsi strategi mereka sendiri-sendiri untuk akuisisi dan pengembangan sistem informasi kesehatan. Akibatnya banyak informasi kesehatan disimpan dan

dikelola dalam sistem yang tidak dapat terhubung satu sama lain (Hovenga dan Grain, 2013).

Di Indonesia proses administrasi kesehatan ibu dan anak menggunakan buku register kohort. Kohort berasal dari kata cohort yang memiliki arti suatu proses pengamatan prospektif, survey prospektif terhadap suatu objek maupun subjek. Sedangkan dalam sudut pelayanan kebidanan register kohort adalah sumber data pelayanan ibu hamil (antenatal), bayi (neonatal), balita dan apras. Dari beberapa fungsi kohort tersebut dapat dibagi menjadi buku kohort ibu, kohort bayi, kohort balita dan kohort apras.

Register kohort ibu merupakan sumber data pelayanan ibu hamil dan bersalin, serta segala keadaan atau resiko yang dimiliki ibu yang diorganisir sedemikian rupa yang pengkoleksiannya melibatkan kader posyandu dan dukun bayi di wilayahnya setiap bulan yang mana informasi pada saat ini lebih difokuskan pada kesehatan ibu dan bayi baru lahir tanpa adanya duplikasi informasi. Register kohort bayi merupakan sumber data pelayanan kesehatan bayi, termasuk masa neonatal. Neonatal adalah interval waktu yang dimulai dari saat lahir sampai 28 hari setelah lahir. Sedangkan register kohort balita merupakan sumber data pelayanan kesehatan balita, umur 12 bulan sampai dengan 5 tahun. Data dalam register balita juga meliputi data imunisasi dan gizi pada balita. Sedangkan Register kohort apras adalah sumber data pelayanan anak pra sekolah yaitu usia 5 tahun sampai 7 tahun. Data yang terkoleksi dalam register ini adalah data pelayanan kesehatan imunisasi dan gizi dari apras. Proses pencatatannya sendiri biasanya dilakukan pada play group atau taman kanak-kanak desa setempat.

Alur proses pengisian register kohort dilakukan dari tingkat bawah yaitu posyandu atau tingkat Rukun Warga (RW). Proses pencatatan pada tahap ini menggunakan buku kohort RW atau biasanya menggunakan buku bantu yang terdiri dari register kohort ibu, bayi, balita, apras. Setiap desa memiliki rata-rata empat sampai enam posyandu. Setelah dilakukan proses administrasi pada posyandu data tersebut harus disalin pada buku kohort desa yang sangat tebal. Proses ini membutuhkan waktu yang lama dan sangat tidak efektif. Bisa dihitung jika setiap desa terdiri dari lima posyandu dan masing-masing posyandu melayani 50 ibu beserta bayinya maka data yang disalin adalah 500 data riwayat kesehatan. Proses tersebut tidak berhenti pada tingkat desa saja namun masih perlu dilakukan rekapan data pada tingkat puskesmas kecamatan. Tugas tersebut dilakukan oleh petugas BKIA kecamatan. Melihat proses yang sangat rumit dan panjang tersebut tentu sangat dibutuhkan sebuah sistem yang dapat menyelesaikan masalah tersebut.

2.3 Pemetaan Kesehatan Ibu dan Anak

Bidan desa sebagai wakil BKIA kecamatan yang bertugas di desa masing-masing wajib melaporkan kegiatan pelayanan kesehatan ibu dan anak berdasarkan program pemerintah PWS-KIA. Dalam buku pedoman PWS-KIA ada beberapa indikator pemantauan yang harus dilaporkan oleh bidan desa pada tiap bulannya.

Dari laporan tersebut maka dalam setahun akan muncul pemetaan mengenai status cakupan dalam suatu desa (Depkes,2010).

2.3.1 Indikator Pemantauan

Indikator pemantauan program kesehatan ibu dan anak yang dipakai untuk PWS KIA terdiri dari indikator yang dapat menggambarkan keadaan kegiatan pokok dalam program KIA. Sasaran yang digunakan dalam PWS KIA berdasarkan kurun waktu satu tahun dengan prinsip konsep wilayah (misalnya: Untuk provinsi memakai sasaran provinsi, untuk kabupaten memakai sasaran kabupaten). Indikator pemantauan terdiri dari lima belas cakupan beserta target yang harus dicapai (Depkes,2010). Berdasarkan data di Puskesmas Kecamatan Tumpang cakupan dan target dapat dilihat pada Tabel 2.1 berikut :

Tabel 2.1 Cakupan-cakupan PWS-KIA dan presentase target

No.	Kode Cakupan	Definisi	Target (%)
1.	K1	Cakupan Akses Pelayanan Antenatal	100%
2.	K4	Cakupan pelayanan ibu hamil	96%
3.	Pn	Cakupan Persalinan oleh tenaga kesehatan	100%
4.	Fn	Cakupan Persalinan di fasilitas kesehatan	100%
5.	KF3	Cakupan pelayanan nifas oleh tenaga kesehatan	96%
6.	KN 1	Cakupan Pelayanan Neonatus Pertama	97%
7.	KN L	Cakupan Pelayanan Kesehatan Neonatus 0-28 hari	95%
8.	DKn	Cakupan deteksi faktor risiko dan komplikasi oleh Masyarakat	87%
9.	HR	Ibu Hamil Resiko Tinggi Komplikasi yang ditangani	87%
10.	PK	Cakupan Penanganan Komplikasi Obstetri	80%
11.	PKn	Cakupan Penanganan Komplikasi Neonatus	79%
12.	KBy	Cakupan Pelayanan Kesehatan Bayi 29 hari - 12 bulan	90%
13.	KBal	Cakupan Pelayanan Anak Balita (12- 59 bulan)	100%
14.	BS	Cakupan Pelayanan Kesehatan Anak Balita Sakit yang dilayani dengan MTBS	90%
15.	Dap	Cakupan DDTK APRAS	90%

Sumber : Buku Panduan PWS-KIA DepKes 2010

Dari beberapa cakupan-cakupan tersebut memiliki perhitungan yang berbeda-beda untuk pencapaian target.

2.3.1.1 Cakupan Akses Pelayanan Antenatal (K1)

Cakupan Akses Pelayanan Antenatal (K1) adalah cakupan ibu hamil yang pertama kali mendapat pelayanan antenatal (kunjungan pertama) oleh tenaga kesehatan atau bidan desa di suatu wilayah kerja pada kurun waktu tertentu. Indikator akses ini digunakan untuk mengetahui jangkauan pelayanan antenatal serta kemampuan program dalam menggerakkan masyarakat.

Rumus yang dipakai untuk perhitungannya adalah :

$$K1 = \frac{\Sigma x}{\Sigma y} \times 100\% \quad (2.1)$$

Keterangan :

Σx = Jumlah ibu hamil yang pertama kali mendapat pelayanan antenatal (kunjungan pertama) oleh tenaga kesehatan disuatu wilayah kerja dan kurun waktu tertentu

Σy = Jumlah sasaran ibu hamil disuatu wilayah kerja dalam 1 tahun

Jumlah sasaran ibu hamil dalam 1 tahun dapat diperoleh melalui proyeksi, dihitung berdasarkan perkiraan jumlah ibu hamil dengan menggunakan rumus :

$$\Sigma y = 1,10 \times \text{angka kelahiran kasar (CBR)} \times \text{jumlah penduduk}$$

Angka kelahiran kasar (CBR) yang digunakan adalah angka terakhir CBR kabupaten/kota yang diperoleh dari kantor perwakilan Badan Pusat Statistik (BPS) di kabupaten/kota.

Contoh :

Untuk menghitung perkiraan jumlah ibu hamil di desa/kelurahan X di kabupaten Y yang mempunyai penduduk sebanyak 2.000 jiwa dan angka CBR terakhir kabupaten Y 27,0/1.000 penduduk, maka :

$$\Sigma y = 1,10 \times 0,027 \times 2.000 = 59,4$$

Jadi sasaran ibu hamil di desa/kelurahan X adalah 59 orang.

2.3.1.2 Cakupan Pelayanan ibu hamil (K4)

Cakupan K4 adalah cakupan ibu hamil yang telah memperoleh pelayanan antenatal sesuai dengan standar, paling sedikit empat kali dengan distribusi waktu 1 kali pada trimester ke-1, 1 kali pada trimester ke-2 dan 2 kali pada trimester ke-3 disuatu wilayah kerja pada kurun waktu tertentu. Dengan indikator ini dapat diketahui cakupan pelayanan antenatal secara lengkap (memenuhi standar pelayanan dan menepati waktu yang ditetapkan), yang menggambarkan tingkat perlindungan ibu hamil di suatu wilayah, di samping menggambarkan kemampuan manajemen ataupun kelangsungan program KIA. Rumus yang dipergunakan adalah :

$$K4 = \frac{\Sigma x}{\Sigma y} \times 100\% \quad (2.2)$$

Keterangan :

Σx = Jumlah ibu hamil yang mendapatkan pelayanan antenatal minimal 4 kali sesuai standar oleh tenaga kesehatan disuatu wilayah kerja pada kurun waktu tertentu

Σy = Jumlah sasaran ibu hamil disuatu wilayah kerja dalam 1 tahun

2.3.1.3 Cakupan Persalinan oleh tenaga kesehatan (Pn)

Cakupan Pn adalah cakupan ibu bersalin yang mendapat layanan pertolongan persalinan oleh tenaga kesehatan atau bidan desa yang memiliki kompetensi kebidanan, di suatu wilayah kerja dalam kurun waktu tertentu. Dengan indikator ini dapat diperkirakan ukuran persalinan yang ditangani oleh tenaga kesehatan dan ini menggambarkan kemampuan manajemen program KIA dalam pertolongan persalinan sesuai standar.

Rumus yang dipergunakan adalah :

$$Pn = \frac{\Sigma x}{\Sigma y} \times 100\% \quad (2.3)$$

Keterangan :

Σx = Jumlah persalinan yang ditolong oleh tenaga kesehatan kompeten atau bidan desa disuatu wilayah kerja pada kurun waktu tertentu

Σy = Jumlah sasaran ibu bersalin disuatu wilayah kerja dalam 1 tahun

Jumlah sasaran ibu bersalin dalam 1 tahun dihitung dengan menggunakan rumus:

$$\Sigma y = 1,05 \times \text{angka kelahiran kasar (CBR)} \times \text{jumlah penduduk}$$

Contoh :

Untuk menghitung perkiraan jumlah ibu bersalin di desa/kelurahan X di kabupaten Y yang mempunyai penduduk sebanyak 2.000 penduduk dan angka CBR terakhir kabupaten Y 27,0/1.000 penduduk maka :

$$\Sigma y = 1,05 \times 0,027 \times 2.000 = 56,7$$

Jadi sasaran ibu bersalin di desa/kelurahan X adalah 56 orang.

2.3.1.4 Cakupan Persalinan di fasilitas kesehatan (Fn)

Cakupan Fn adalah cakupan ibu bersalin yang mendapat layanan pertolongan persalinan di fasilitas kesehatan seperti bidan desa, pustu, bidan praktek swasta, puskesmas rumah sakit (bukan dirumah ibu hamil dalam kurun waktu tertentu).

Rumus yang dipergunakan adalah :

$$Fn = \frac{\Sigma x}{\Sigma y} \times 100\% \quad (2.3)$$

Keterangan :

Σx = Jumlah persalinan di fasilitas kesehatan pada kurun waktu tertentu

Σy = Jumlah sasaran ibu bersalin disuatu wilayah kerja dalam 1 tahun

2.3.1.5 Cakupan pelayanan nifas oleh tenaga kesehatan (KF3)

Cakupan Kf3 adalah cakupan pelayanan kepada ibu pada masa 6 jam sampai dengan 42 hari pasca bersalin sesuai standar paling sedikit tiga kali dengan pembagian waktu 6 jam s/d hari ke-3 (KF1), hari ke-4 s/d hari ke-28 (KF2) dan hari ke-29 s/d hari ke-42 (KF3) setelah bersalin di suatu wilayah kerja pada kurun waktu tertentu. Dengan indikator ini maka bisa diketahui cakupan pelayanan nifas secara lengkap (memenuhi standar pelayanan dan menepati waktu yang ditetapkan serta untuk menjaring KB Pasca Persalinan), yang menggambarkan jangkauan dan kualitas pelayanan kesehatan ibu nifas, Keluarga Berencana di samping menggambarkan kemampuan manajemen ataupun kelangsungan program KIA.

Rumus yang dipergunakan adalah :

$$KF3 = \frac{\Sigma x}{\Sigma y} \times 100\% \quad (2.4)$$

Keterangan :

Σx = Jumlah ibu nifas yang telah memperoleh tiga kali pelayanan nifas sesuai standar oleh tenaga kesehatan disuatu wilayah kerja pada kurun waktu tertentu

Σy = Jumlah sasaran ibu nifas di suatu wilayah kerja dalam 1 tahun

Jumlah sasaran ibu nifas sama dengan jumlah sasaran ibu bersalin

2.3.1.6 Cakupan Pelayanan Neonatus Pertama (KN 1)

Cakupan KN-1 adalah cakupan neonatus yang mendapatkan pelayanan sesuai standar pada 6 - 48 jam setelah lahir di suatu wilayah kerja pada kurun waktu tertentu. Dengan indikator ini dapat diketahui jangkauan pelayanan kesehatan neonatal.

Rumus yang dipergunakan adalah sebagai berikut :

$$KN 1 = \frac{\Sigma x}{\Sigma y} \times 100\% \quad (2.5)$$

Keterangan :

Σx = Jumlah neonatus yang mendapatkan pelayanan sesuai standar pada

6-48 jam setelah lahir di suatu wilayah kerja pada kurun waktu tertentu

Σy = Jumlah seluruh sasaran bayi di suatu wilayah kerja dalam 1 tahun

Jumlah sasaran bayi bisa didapatkan dari perhitungan berdasarkan jumlah perkiraan (angka proyeksi) bayi dalam satu wilayah tertentu dengan menggunakan rumus sebagai berikut :

$$\Sigma y = \text{Crude Birth Rate} \times \text{jumlah penduduk}$$

Contoh :

Untuk menghitung jumlah perkiraan bayi di suatu desa Z di Kota Y Propinsi X yang mempunyai penduduk sebanyak 1.500 jiwa dan angka CBR terakhir Kota Y 24,8/1.000 penduduk, maka :

$$\Sigma y = 0,0248 \times 1500 = 37,2$$

Jadi sasaran bayi di desa Z adalah 37 bayi.

2.3.1.7 Cakupan Pelayanan Kesehatan Neonatus 0-28 hari (KN Lengkap)

Cakupan KN Lengkap adalah cakupan neonatus yang mendapatkan pelayanan sesuai standar paling sedikit tiga kali dengan distribusi waktu 1 kali pada 6-48 jam, 1 kali pada hari ke 3-hari ke 7 dan 1 kali pada hari ke 8-hari ke 28 setelah lahir disuatu wilayah kerja pada kurun waktu tertentu. Dengan indikator ini dapat diketahui efektifitas dan kualitas pelayanan kesehatan neonatal.

Rumus yang dipergunakan adalah sebagai berikut :

$$KN \text{ Lengkap} = \frac{\Sigma x}{\Sigma y} \times 100\% \quad (2.6)$$

Keterangan :

Σx = Jumlah neonatus yang telah memperoleh 3 kali pelayanan kunjungan neonatal sesuai standar di suatu wilayah kerja pada kurun waktu tertentu

Σy = Jumlah seluruh sasaran bayi di suatu wilayah kerja dalam 1 tahun

2.3.1.8 Cakupan Deteksi Faktor Risiko dan Komplikasi oleh Masyarakat (DKn)

Cakupan DKn adalah cakupan ibu hamil dengan faktor risiko atau komplikasi yang ditemukan oleh kader atau dukun bayi atau masyarakat serta dirujuk ke tenaga kesehatan di suatu wilayah kerja pada kurun waktu tertentu. Masyarakat disini, bisa keluarga ataupun ibu hamil, bersalin, nifas itu sendiri. Indikator ini menggambarkan peran serta dan keterlibatan masyarakat dalam mendukung upaya peningkatan kesehatan ibu hamil, bersalin dan nifas.

Rumus yang dipergunakan adalah sebagai berikut :

$$DKn = \frac{\Sigma x}{20\% \times \Sigma y} \times 100\% \quad (2.7)$$

Keterangan :

Σx = Jumlah ibu hamil yang berisiko yang ditemukan kader atau dukun bayi atau masyarakat di suatu wilayah kerja pada kurun waktu tertentu

Σy = jumlah sasaran ibu hamil di suatu wilayah kerja dalam 1 tahun

2.3.1.9 Ibu Hamil Resiko Tinggi Komplikasi yang ditangani (HR)

Cakupan HR adalah jumlah ibu hamil dengan faktor risiko tinggi yang ditemukan oleh kader atau dukun bayi atau masyarakat serta ditangani oleh tenaga kesehatan di suatu wilayah kerja pada kurun waktu tertentu.

Rumus yang dipergunakan adalah sebagai berikut :

$$HR = \frac{\Sigma x}{20\% \times \Sigma y} \times 100\% \quad (2.7)$$

Keterangan :

Σx = Jumlah ibu hamil yang berisiko tinggi di suatu wilayah kerja pada kurun waktu tertentu

Σy = jumlah sasaran ibu hamil di suatu wilayah kerja dalam 1 tahun

2.3.1.10 Cakupan Penanganan Komplikasi Obstetri (PK)

Cakupan PK adalah cakupan Ibu dengan komplikasi kebidanan di suatu wilayah kerja pada kurun waktu tertentu yang ditangani secara definitif sesuai dengan standar oleh tenaga kesehatan kompeten pada tingkat pelayanan dasar dan rujukan. Penanganan definitif adalah penanganan/pemberian tindakan terakhir untuk menyelesaikan permasalahan setiap kasus komplikasi kebidanan. Indikator ini mengukur kemampuan manajemen program KIA dalam menyelenggarakan pelayanan kesehatan secara professional kepada ibu hamil bersalin dan nifas dengan komplikasi.

Rumus yang dipergunakan adalah sebagai berikut :

$$PK = \frac{\Sigma x}{20\% \times \Sigma y} \times 100\% \quad (2.8)$$

Keterangan :

Σx = Jumlah komplikasi kebidanan yang mendapatkan penanganan definitif di suatu wilayah kerja pada kurun waktu tertentu

Σy = jumlah sasaran ibu hamil di suatu wilayah kerja dalam 1 tahun

2.3.1.11 Cakupan Penanganan Komplikasi Neonatus (PKn)

Cakupan PKn adalah cakupan neonatus dengan komplikasi yang ditangani secara definitif oleh tenaga kesehatan kompeten pada tingkat pelayanan dasar dan rujukan di suatu wilayah kerja pada kurun waktu tertentu. Penanganan definitif merupakan pemberian tindakan akhir pada setiap kasus komplikasi

neonatus yang pelaporannya dihitung 1 kali pada masa neonatal. Kasus komplikasi yang ditangani adalah seluruh kasus yang ditangani tanpa melihat hasilnya hidup atau mati. Indikator ini menunjukkan kemampuan sarana pelayanan kesehatan dalam menangani kasus-kasus kegawatdaruratan neonatal, yang kemudian ditindaklanjuti sesuai dengan kewenangannya, atau dapat dirujuk ke tingkat pelayanan yang lebih tinggi.

Rumus yang dipergunakan adalah sebagai berikut :

$$PKn = \frac{\Sigma x}{15\% \times \Sigma y} \times 100\% \quad (2.9)$$

Keterangan :

Σx = Jumlah neonatus dengan komplikasi yang mendapat penanganan definitif di suatu wilayah kerja pada kurun waktu tertentu

Σy = jumlah sasaran bayi di suatu wilayah kerja dalam 1 tahun

2.3.1.12 Cakupan Pelayanan Kesehatan Bayi 29 hari - 12 bulan (KBy)

Cakupan KBy adalah cakupan bayi yang mendapatkan pelayanan paripurna minimal 4 kali yaitu 1 kali pada umur 29 hari - 2 bulan, 1 kali pada umur 3-5 bulan, dan satu kali pada umur 6-8 bulan dan 1 kali pada umur 9-11 bulan sesuai standar di suatu wilayah kerja pada kurun waktu tertentu. Dengan indikator ini dapat diketahui efektifitas, continuum of care dan kualitas pelayanan kesehatan bayi.

Rumus yang dipergunakan adalah sebagai berikut :

$$KBy = \frac{\Sigma x}{\Sigma y} \times 100\% \quad (2.10)$$

Keterangan :

Σx = Jumlah bayi yang telah memperoleh 4 kali pelayanan kesehatan sesuai standar di suatu wilayah kerja pada kurun waktu tertentu

Σy = Jumlah seluruh sasaran bayi di suatu wilayah kerja dalam 1 tahun

2.3.1.13 Cakupan Pelayanan Anak Balita 12- 59 bulan (KBal)

Cakupan KBal adalah cakupan anak balita (12-59 bulan) yang memperoleh pelayanan sesuai standar, meliputi pemantauan pertumbuhan minimal 8x setahun, pemantauan perkembangan minimal 2 x setahun, pemberian vitamin A 2 x setahun.

Rumus yang dipergunakan adalah sebagai berikut :

$$KBal = \frac{\Sigma x}{\Sigma y} \times 100\% \quad (2.11)$$

Keterangan :

Σx = Jumlah anak balita yg memperoleh pelayanan sesuai standar disuatu wilayah kerja pada kurun waktu tertentu

Σy = Jumlah seluruh anak balita disuatu wilayah kerja dalam 1 tahun

2.3.1.14 Cakupan Pelayanan Kesehatan Anak Balita Sakit yang dilayani dengan MTBS (BS)

Cakupan BS adalah cakupan anak balita (umur 12-59 bulan) yang berobat ke Puskesmas dan mendapatkan pelayanan kesehatan sesuai standar (MTBS) di suatu wilayah kerja pada kurun waktu tertentu.

Rumus yang dipergunakan adalah sebagai berikut :

$$BS = \frac{\Sigma x}{\Sigma y} \times 100\% \quad (2.12)$$

Keterangan :

Σx = Jumlah anak balita sakit yg memperoleh pelayanan sesuai tatalaksana MTBS di Puskesmas di suatu wilayah kerja pada kurun waktu tertentu

Σy = Jumlah seluruh anak balita sakit yang berkunjung ke Puskesmas disuatu wilayah kerja dalam 1 tahun

Jumlah anak balita sakit diperoleh dari kunjungan balita sakit yang datang ke puskesmas (register rawat jalan di Puskesmas). Jumlah anak balita sakit yang mendapat pelayanan standar diperoleh dari format pencatatan dan pelaporan MTBS.

2.3.1.15 Cakupan DDTK APRAS (Dap)

Cakupan Dap adalah cakupan anak prasekolah (5-6 tahun) yang memperoleh pelayanan sesuai standar, meliputi pemantauan pertumbuhan minimal 4x setahun, pemantauan perkembangan minimal 2 x setahun, pemberian vitamin A 2 x setahun.

Rumus yang dipergunakan adalah sebagai berikut :

$$Dap = \frac{\Sigma x}{\Sigma y} \times 100\% \quad (2.13)$$

Keterangan :

Σx = Jumlah anak prasekolah yg memperoleh pelayanan sesuai standar disuatu wilayah kerja pada kurun waktu tertentu

Σy = Jumlah seluruh anak prasekolah disuatu wilayah kerja dalam 1 tahun

2.3.2 Status Cakupan Desa

Status cakupan desa merupakan hasil analisis dari rekapan dari seluruh indikator cakupan beserta target yang dicapai. Analisis merupakan suatu pengolahan data dan evaluasi dari suatu informasi yang sesuai dan relevant dalam menentukan suatu tindakan yang terbaik dari berbagai macam alternatif variasi yang ada. Analisis yang dapat dilakukan mulai dari yang sederhana hingga analisis lanjut sesuai dengan tingkatan penggunaannya. Data yang di analisis adalah data register kohort ibu, bayi, balita dan anak pra sekolah serta cakupan.

Pemetaan berdasar analisis data membandingkan cakupan hasil kegiatan antar wilayah terhadap target dan kecenderungan dari waktu ke waktu. Analisis ini bermanfaat untuk mengetahui desa/kelurahan mana yang paling memerlukan perhatian dan tindak lanjut yang harus dilakukan. Selain di Puskesmas, analisis ini dapat juga dilakukan oleh Bidan di Desa dimana Bidan di Desa dapat menilai cakupan indikator PWS KIA di desanya untuk menilai kemajuan desanya. Di Poskesdes seorang Bidan di Desa dapat membuat grafik cakupan indikator PWS KIA sehingga dia bisa mengikuti perkembangan dan menindak lanjutinya (Depkes,2010).

Untuk contoh analisis sederhana yang menggambarkan cakupan pada sebuah desa dapat dilihat pada Gambar 2.1 berikut,

Desa/ Kelurahan	Cakupan Terhadap Target		Terhadap Cakupan Bulan Lalu			Status Desa/ Kelurahan
	Diatas	Dibawah	Naik	Turun	Tetap	
A	+		+			Baik
B	+				+	Baik
C	+			+		Kurang
D		+	+			Cukup
E		+		+		Jelek

Gambar 2.1 Contoh analisis sederhana dari seluruh cakupan

Dari Gambar 2.1 maka dapat dikategorikan ada empat status cakupan dalam sebuah desa. Berikut penjelasan masing-masing status cakupannya,

2.3.2.1 Status Cakupan Baik

Status cakupan baik adalah desa atau kelurahan dengan cakupan diatas target yang ditetapkan untuk bulan ini, dan mempunyai kecenderungan cakupan bulanan yang meningkat atau tetap jika dibandingkan dengan cakupan bulan lalu. Desa/kelurahan-desa/kelurahan ini adalah desa/kelurahan A dan desa/kelurahan B. Jika keadaan tersebut berlanjuti, maka desa/kelurahan-desa/kelurahan tersebut akan mencapai atau melebihi target tahunan yang ditentukan.

2.3.2.2 Status Cakupan Kurang

Status cakupan kurang adalah desa atau kelurahan dengan cakupan diatas target bulan ini, namun mempunyai kecenderungan cakupan bulanan yang menurun jika dibandingkan dengan cakupan bulan lalu. Desa/kelurahan dalam kategori ini adalah desa/kelurahan C, yang perlu mendapatkan perhatian karena cakupan bulan lalu ini hanya 5% (lebih kecil dari cakupan bulan minimal 7,5%). Jika cakupan terus menurun, maka desa/kelurahan tersebut tidak akan mencapai target tahunan yang ditentukan.

2.3.2.3 Status Cakupan Cukup

Status cakupan cukup adalah desa atau kelurahan dengan cakupan dibawah target bulan ini, namun mempunyai kecenderungan cakupan bulanan yang meningkat jika dibandingkan dengan cakupan bulan lalu. Desa/kelurahan dalam kategori ini adalah desa/kelurahan D, yang perlu didorong agar cakupan bulanan selanjutnya tidak lebih daripada cakupan bulanan minimal 7,5%. Jika keadaan tersebut dapat terlaksana, maka desa/kelurahan ini kemungkinan besar akan mencapai target tahunan yang ditentukan.

2.3.2.4 Status Cakupan Jelek

Status cakupan jelek adalah desa atau kelurahan dengan cakupan dibawah target bulan ini, dan mempunyai kecenderungan cakupan bulanan yang menurun dibandingkan dengan bulan lalu. Desa/kelurahan dalam kategori ini adalah desa/kelurahan E, yang perlu diprioritaskan untuk pembinaan agar cakupan bulanan selanjutnya dapat ditingkatkan diatas cakupan bulanan minimal agar dapat mengejar kekurangan target sampai bulan Juni, sehingga dapat pula mencapai target tahunan yang ditentukan.

2.4 Konsep dasar Sistem Informasi

Gondodiyoto (2007) dalam Surya; Wibowo; Amelia (2013) menyatakan sistem adalah merupakan kumpulan dari elemen-elemen atau subsistem yang orientasinya terhadap tujuan tertentu. Berdasarkan batasan pengertian tersebut, sistem mempunyai beberapa karakteristik. Sistem adalah kumpulan komponen-komponen atau resource yang saling berkaitan secara terpadu, terintegrasi dalam suatu hubungan hirarkis. Selain itu sistem memiliki sasaran yang akan dicapai. Setiap sistem berusaha mencapai satu atau lebih sasaran yang merupakan arah, yang merupakan kekuatan yang memberikan arah suatu sistem. Konstruksi sistem terdiri dari: masukan proses keluaran. Masukan merupakan semua arus berwujud atau tidak berwujud yang masuk ke sistem. Keluaran merupakan semua arus keluar atau akibat yang dihasilkan. Proses terdiri dari metode yang digunakan untuk mengubah masukan menjadi keluaran. Sistem juga harus memiliki pengguna. Setiap sistem harus mengarahkan sub sistemnya agar dapat mencapai sasaran. Sasaran sistem sebagai ukuran penentu keberhasilan suatu sistem. Sistem memiliki keterbatasan. Sistem memiliki sub sistem yang membentuk suatu

jaringan terpadu dan karakteristik terakhir adalah sistem memerlukan pengendalian atau mainten.

Informasi merupakan kumpulan data yang diambil dari suatu kegiatan. Dalam hal ini administrasi juga disebut sebagai informasi. Informasi adalah yang sudah dapat digunakan atau memiliki arti yang dapat dimanfaatkan untuk penerimanya, menggambarkan suatu kejadian dan kesatuan nyata yang dapat dimengerti dan dapat digunakan untuk pengambilan keputusan, sekarang maupun masa depan (Gondodiyoto, 2007 dalam Surya; Wibowo; Amelia, 2013). Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal datum atau data item. Data sebagai input perlu diolah oleh suatu sistem pengolahan data agar dapat menjadi output, yaitu informasi yang lebih berguna bagi pemakainya.

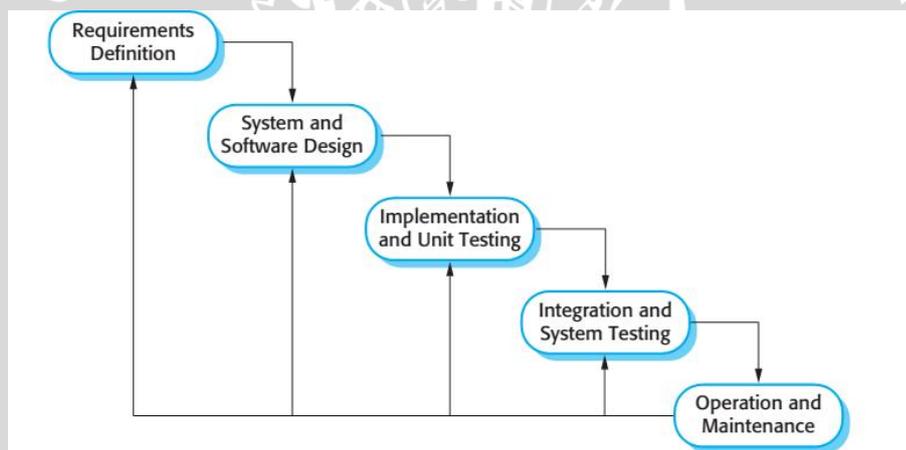
Dari uraian tersebut dapat dikatakan bahwa informasi adalah data yang diolah menjadi bentuk yang lebih berguna, lebih bermanfaat dan lebih berarti bagi penggunanya. Data menggambarkan suatu kejadiankejadian, data dinyatakan sebagai symbolsymbol, gambar-gambar, kata-kata, angkaangka, atau huruf-huruf yang menunjukkan suatu ide, obyek, kondisi atau situasi tertentu. Informasi digunakan untuk pengambilan keputusan. Bagi manajemen suatu organisasi, informasi berguna untuk membantu dalam pengambilan keputusan yang menentukan keberhasilan atau kesuksesan organisasi pada masa yang akan datang.

Jadi menurut penjelasan mengenai sistem dan informasi diatas, dapat disimpulkan bahwa sistem informasi (sistem administrasi) adalah sistem yang akan menghasilkan sebuah informasi. Lebih lanjut, sistem informasi diartikan sebagai sistem berbasis komputer yang terdiri dari perangkat keras, perangkat lunak, pengguna, data, proses, dan prosedur yang bekerja sama untuk menghasilkan informasi yang berguna yang berhubungan dengan operasi dari sebuah organisasi bisnis. Sebuah sistem informasti dapat menerima sumber daya data sebagai input, dan kemudian menerjemahkan sumber daya ini menjadi informasi sebagai output. Sebuah sistem informasi dapat mendukung kegiatan baik jangka pendek dan jangka panjang bagi pengguna dalam sebuah organisasi (Pizam, 2005).

Sistem informasi (administrasi) modern memecahkan berbagai data, informasi, dan masalah berbasis pengetahuan. Di masa lalu, sistem informasi sebagian besar secara eksklusif berorientasi pada data. Tujuan utamanya adalah untuk menyimpan, mengambil, memanipulasi, dan menampilkan data (Hill, 2005). Kebutuhan sistem informasi menciptakan berbagai klasifikasi. Klasifikasi sistem informasi menurut Kaliski (2006) dalam bukunya yang berjudul Encyclopedia of Business and Finance cetakan Macmillan Reference USA tahun terdiri dari klasifikasi tingkat organisasi, klasifikasi area fungsional, klasifikasi area dukungan dan klasifikasi arsitektur

2.5 Model Pengembangan Waterfall

Pengembangan model perangkat lunak dalam hal ini adalah sistem informasi terdiri dari banyak pendekatan. Umumnya model proses menggambarkan proses organisasi dari sistem. Beberapa model antara lain model waterfall (air terjun), incremental development, and reuse-oriented development. Dalam penelitian ini menggunakan model pengembangan waterfall (air terjun). Model waterfall adalah paradigma tertua untuk rekayasa perangkat lunak. Akan Tetapi akhir-akhir ini akibat kritik model proses ini telah menyebabkan bahkan pendukungnya bersemangat mempertanyakan efektivitasnya. (Pressman, 2010) Model waterfall memiliki kegiatan proses spesifikasi dasar, pengembangan, validasi, dan evolusi. Beberapa kegiatan tersebut terdiri fase spesifikasi persyaratan, perancangan perangkat lunak, implementasi, pengujian, dan sebagainya. Model waterfall merupakan model yang diterbitkan pertama dari proses pengembangan perangkat lunak yang berasal dari sistem yang lebih umum dari proses rekayasa Model ini diilustrasikan pada Gambar 2.2. Karena kaskade dari satu fase ke yang lain, model ini dikenal sebagai model waterfall (air terjun) atau siklus hidup perangkat lunak. Model air terjun pada prinsipnya anda harus merencanakan dan menjadwalkan semua proses kegiatan sebelum mulai bekerja pada mereka.(Sommerville, 2011).



Gambar 2.2 Model pengembangan sistem waterfall

Menurut Sommerville (2011) tahapan-tahapan dari model waterfall adalah sebagai berikut :

1. Analisis Kebutuhan

Sistem pelayanan, kendala, dan tujuan ditetapkan melalui konsultasi dengan pengguna sistem. Mereka kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem. Tahapan analisis kebutuhan terdiri dari tahap elisitasi, spesifikasi, validasi dan verifikasi serta manajemen kebutuhan.

2. Sistem dan perancangan perangkat lunak

Proses desain sistem mengalokasikan persyaratan baik perangkat keras atau perangkat lunak sistem dengan membentuk sistem secara keseluruhan

arsitektur. Software desain melibatkan identifikasi dan menggambarkan abstraksi sistem perangkat lunak yang mendasar dan hubungan mereka. Perancangan terdiri dari perancangan arsitektur, perancangan data, perancangan komponen dan perancangan antarmuka.

3. Implementasi dan unit testing

Selama tahap ini, desain perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Unit pengujian melibatkan memverifikasi bahwa setiap unit memenuhi spesifikasinya.

4. Integrasi dan pengujian sistem

Unit program individu atau program diintegrasikan dan diuji sebagai sistem yang lengkap untuk memastikan bahwa perangkat lunak persyaratan telah dipenuhi. Setelah pengujian, sistem perangkat lunak yang dikirimkan ke pelanggan.

5. Operasi dan pemeliharaan

Biasanya (meskipun tidak selalu), ini adalah fase terpanjang dalam siklus hidup. Sistem ini dipasang dan dimasukkan ke dalam penggunaan praktis. Pemeliharaan melibatkan kesalahan mengoreksi yang tidak ditemukan di awal tahap siklus hidup, meningkatkan pelaksanaan unit sistem dan meningkatkan pelayanan sistem sebagai requirements are baru yang ditemukan.

Pada prinsipnya, hasil dari setiap tahap adalah satu atau lebih dokumen yang disetujui. Tahap berikutnya tidak harus mulai sampai tahap sebelumnya selesai. Dalam prakteknya, tahap ini tumpang tindih dan memberikan informasi satu sama lain. Dalam Waktu desain, masalah dengan persyaratan diidentifikasi. Selama coding, masalah desain ditemukan dan sebagainya. Proses perangkat lunak bukan model linear sederhana namun melibatkan umpan balik dari satu fase yang lain. Dokumen yang dihasilkan dalam setiap fase mungkin kemudian telah diubah untuk mencerminkan perubahan yang dibuat.

Karena biaya produksi dan menyetujui dokumen, iterasi dapat mahal dan melibatkan ulang yang signifikan. Oleh karena itu, setelah sejumlah kecil iterasi, itu adalah normal untuk membekukan bagian dari pengembangan, seperti spesifikasi, dan untuk melanjutkan tahap perkembangan selanjutnya. Masalah yang tersisa untuk resolusi kemudian, diabaikan, atau diprogram sekitar. Ini pembekuan dini persyaratan dapat berarti bahwa sistem tidak akan melakukan apa yang diinginkan pengguna. Hal ini juga dapat menyebabkan buruk terstruktur sistem sebagai masalah desain dielakkan oleh trik implementasi. Selama fase siklus hidup akhir (operasi dan pemeliharaan) perangkat lunak diletakkan mulai digunakan. Kesalahan dan kelalaian dalam persyaratan perangkat lunak yang asli ditemukan. Program dan desain kesalahan muncul dan kebutuhan untuk fungsi baru diidentifikasi. Oleh karena itu sistem harus berevolusi untuk tetap berguna. Membuat perubahan ini (software maintenance) mungkin melibatkan mengulangi tahap proses sebelumnya.

Model air terjun ini konsisten dengan model proses rekayasa lainnya dan dokumentasi diproduksi pada setiap fase. Hal ini membuat proses terlihat

sehingga manajer dapat memantau kemajuan terhadap rencana pembangunan. Masalah utama adalah partisi fleksibel proyek dalam tahap yang berbeda. Komitmen harus dilakukan pada tahap awal dalam proses, yang membuatnya sulit untuk merespon perubahan kebutuhan pelanggan.

Pada prinsipnya, model air terjun harus digunakan hanya ketika persyaratan yang dipahami dengan baik dan tidak mungkin berubah secara radikal selama pengembangan sistem. Namun, model air terjun mencerminkan jenis proses yang digunakan dalam teknik lainnya proyek. Seperti mudah untuk menggunakan model manajemen umum untuk keseluruhan proyek, Proses perangkat lunak berbasis pada model air terjun arestill umum digunakan.

Sebuah varian penting dari model air terjun adalah pengembangan sistem formal, di mana model matematika dari spesifikasi sistem yang dibuat. Model ini kemudian halus, menggunakan transformasi matematika yang melestarikan konsistensi, ke dalam kode executable. Berdasarkan asumsi bahwa transformasi matematika Anda benar, karena itu Anda dapat membuat argumen yang kuat bahwa program yang dihasilkan dalam cara konsisten dengan spesifikasinya.

Proses formal pembangunan, seperti yang berbasis onthe metode B (Schneider, 2001; Wordsworth, 1996 dalam Sommerville, 2011) sangat cocok untuk pengembangan sistem yang memiliki keamanan yang ketat, keandalan, atau persyaratan keamanan. Pendekatan formal menyederhanakan produksi kasus keselamatan atau keamanan. Hal ini menunjukkan kepada pelanggan atau regulator bahwa sistem benar-benar memenuhi persyaratan keselamatan atau keamanan. Proses berdasarkan transformasi formal umumnya hanya digunakan dalam pengembangan sistem keselamatan-kritis atau penting keamanan. Mereka memerlukan khusus keahlian. Untuk sebagian besar sistem ini prosesdoes tidak menawarkan costbenefits signifikan terhadap pendekatan lain untuk pengembangan sistem (Sommerville, 2011).

2.5.1 Rekayasa Kebutuhan

Rekayasa kebutuhan adalah deskripsi dari apa yang harus sistem lakukan dengan menyediakan layanan atau kendala pada operasi (Sommerville, 2011). Kebutuhan tersebut mencerminkan kebutuhan pelanggan untuk sebuah sistem yang memiliki tujuan tertentu seperti mengontrol perangkat, menempatkan perintah, atau mencari informasi. Proses menemukan keluaran, menganalisis, mendokumentasikan dan memeriksa layanan serta kendala sistem disebut dengan rekayasa kebutuhan.

Rekayasa kebutuhan terdiri dari tahap elisitasi, spesifikasi kebutuhan, dan manajemen kebutuhan. Menurut Sommerville (2011) berikut pengertian masing-masing tahapannya,

- Elisitasi kebutuhan
Tahap elisitasi adalah kegiatan dimana developer perangkat lunak bekerja dengan *customers* dan pengguna akhir dari sistem untuk mengetahui tentang

domain aplikasi, layanan yang harus disediakan sistem, kinerja yang diperlukan sistem, kendala hardware, dan sebagainya.

- **Spesifikasi kebutuhan**
Spesifikasi kebutuhan adalah proses menuliskan pengguna dan kebutuhan sistem dalam dokumen kebutuhan. Idealnya, pengguna dan persyaratan sistem harus jelas, tidak ambigu, mudah dimengerti, lengkap, dan konsisten.
- **Validasi dan verifikasi**
Dari spesifikasi kebutuhan yang telah didefinisikan perlu adanya validasi dan verifikasi. Validasi kebutuhan adalah proses memeriksa bahwa kebutuhan-kebutuhan sistem benar-benar mendefinisikan sistem yang benar-benar diinginkan oleh client.
- **Manajemen Kebutuhan**
Kebutuhan sistem perangkat lunak besar selalu berubah. Salah satu alasan untuk ini adalah bahwa sistem biasanya dikembangkan untuk mengatasi masalah-masalah yang tidak bisa sepenuhnya didefinisikan. Karena masalah tidak dapat didefinisikan secara penuh, persyaratan software tidak lengkap.

2.5.2 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan hal penting untuk menjamin kualitas dari sistem yang dibangun serta merepresentasikan spesifikasi, desain dan pengkodean perangkat lunak (Gunadarma, 2014 dalam Aulia, 2014). Pengujian ini bertujuan untuk mencari kesalahan-kesalahan yang mungkin terjadi pada perangkat lunak dan dilakukan untuk mengatasi kegagalan yang akan mengakibatkan adanya pekerjaan tambahan dan biaya tambahan yang tidak perlu. Pengujian perangkat lunak dilakukan berdasarkan prinsip-prinsip pengujian diantaranya (Wisnu, 2011 dalam Aulia, 2014) :

- Semua pengujian harus dapat diruntut sampai kepada spesifikasi kebutuhan perangkat lunak
- Pengujian harus dimulai dari lingkup yang besar
- Pengujian yang mendalam tidak mungkin dilakukan karena tidak mungkin mengeksekusi semua jalur permutasi
- Supaya efektif (memiliki probabilitas yang tinggi dalam menemukan kesalahan), pengujian dilakukan oleh pihak lain yang independen
- Pengujian harus direncanakan jauh sebelum dilakukan

2.5.2.1 Pengujian *White Box*

White-box atau pengujian struktural adalah pendekatan sistematis untuk menguji kode program yang digunakan untuk merancang tes kesalahan. Tujuannya adalah untuk merancang tes yang menyediakan beberapa tingkat cakupan program. Artinya, serangkaian tes harus memastikan bahwa setiap jalan yang logis melalui program ini pernah dijalankan, dengan konsekuensi bahwa setiap

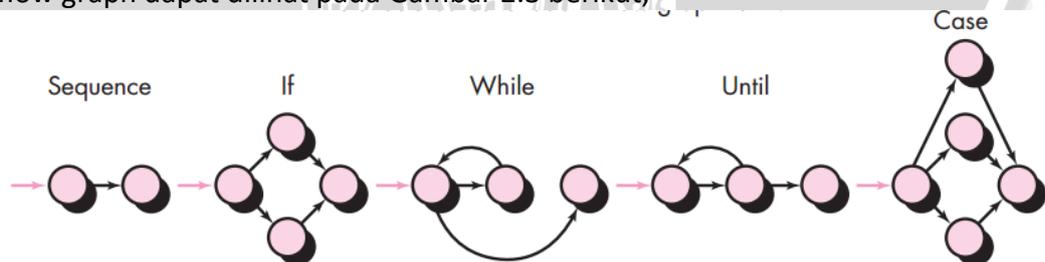
pernyataan program dijalankan minimal sekali. Analisis pelaksanaan program dapat digunakan untuk menunjukkan bahwa tingkat cakupan tes telah dicapai (Sommerville, 2011).

Pengujian whitebox merupakan salah satu metode desain uji kasus yang menggunakan struktur kontrol dari perancangan prosedural untuk menghasilkan kasus-kasus uji dalam algoritma. Menurut Pressman (2010) dengan menggunakan metode ujicoba whitebox, para pengembang software dapat menghasilkan kasus-kasus uji yang antara lain :

1. Menjamin bahwa seluruh jalur independen dalam modul telah dilakukan minimal satu kali
2. Melakukan semua keputusan logika baik dari sisi benar maupun salah
3. Melakukan semua perulangan sesuai batasan yang ada dan dalam batasan operasionalnya
4. Menguji struktur data internal untuk memastikan validitas data

Pengujian white box juga dapat dikatakan sebagai pengujian unit. Pengujian unit berfokus pada upaya verifikasi pada unit terkecil dari desain perangkat lunak, komponen perangkat lunak atau modul (Pressman, 2010). Uji unit berfokus pada logika proses internal dan struktur data dalam batas-batas komponen. Jenis pengujian dapat dilakukan secara paralel untuk beberapa komponen. Salah satu metode ujicoba unit adalah ujicoba berbasis alur (Basis Path Testing). Ujicoba berbasis alur merupakan teknik ujicoba whitebox pertama yang diusulkan oleh Tom McCabe (Pressman, 2010). Langkah-langkah pengujian basis path menurut Pressman (2010) adalah sebagai berikut.

1. Mendefinisikan node pada kode program
Hal ini dilakukan dengan menentukan kode program yang akan berdiri sebagai node-node yang akan digambarkan pada flow graph.
2. Membuat notasi flow graph
Notasi flow graph mendefinisikan control alur dari program. Contoh notasi flow graph dapat dilihat pada Gambar 2.3 berikut,



Gambar 2.3 Contoh notasi flow graph

3. Mendefinisikan independent program path
Jalur independen adalah setiap jalan melalui program yang memperkenalkan setidaknya satu set baru pernyataan. Contoh dari independent path seperti, Path 1 : 1-2-4-6-7 ; Path 2 : 1-2-3-4-5-6-7.

4. Menentukan cyclomatic complexity $V(G)$
Cyclomatic complexity dapat dihitung menggunakan tiga cara yaitu, (1) jumlah dari region dari flow graph, (2) $V(G) = E - N + 2$ dimana E adalah edge atau garis dan N adalah node atau simpul, (3) $V(G) = P + 1$ dimana P adalah predicate node pada flowgraph
5. Membuat test case
Test case adalah merupakan tindakan yang dilakukan dalam kasus uji sesuai dengan jalur yang didefinisikan

2.5.2.2 Pengujian Black Box

Pengujian black-box, juga disebut pengujian validasi, berfokus pada persyaratan fungsional perangkat lunak. Artinya, teknik pengujian black-box memungkinkan anda untuk mendapatkan set kondisi input yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program. Pengujian black-box bukan merupakan alternatif untuk teknik white-box. Sebaliknya pendekatan komplementer yang kemungkinan akan mengungkap kelas yang berbeda dari kesalahan daripada metode whitebox.

Menurut Wisnu dalam Aulia (2014) pengujian black-box berusaha menemukan kesalahan dalam kategori berikut: (1) Kesalahan fungsi atau fungsi yang hilang, (2) kesalahan antarmuka, (3) kesalahan dalam struktur data atau akses database eksternal, (4) perilaku atau kinerja kesalahan, dan (5) inisialisasi dan kesalahan terminasi. Tidak seperti pengujian white-box, yang dilakukan pada awal proses pengujian, pengujian blackbox cenderung diterapkan selama tahap akhir pengujian. Karena pengujian black-box sengaja mengabaikan struktur kontrol, perhatian difokuskan pada informasi domain. Tes ini dirancang untuk menjawab pertanyaan-pertanyaan berikut:

- Bagaimana validitas fungsional diuji?
- Bagaimana perilaku dan kinerja sistem diuji?
- Apa kelas input akan membuat kasus uji yang baik?
- Apakah sistem sangat sensitif terhadap nilai input tertentu?
- Bagaimana batas-batas kelas data diisolasi?
- Apa kecepatan data dan volume data yang dapat mentolerir sistem?
- Apa efek akan kombinasi spesifik data terhadap operasi sistem?

Dengan menerapkan teknik black-box, Anda memperoleh satu set kasus uji yang memenuhi kriteria berikut : (1) uji kasus yang mengurangi, dengan jumlah yang lebih besar dari satu, jumlah kasus uji tambahan yang harus dirancang untuk mencapai wajar pengujian, dan (2) uji kasus yang memberitahu Anda sesuatu tentang ada atau tidak adanya kelas kesalahan, bukan kesalahan terkait hanya dengan tes khusus di tangan (Pressman, 2010).

2.5.2.3 Pengujian Integrasi

Pengujian integrasi adalah teknik yang sistematis untuk membangun arsitektur perangkat lunak dengan melakukan tes untuk mengungkap kesalahan yang terkait dengan tampilan program. Tujuannya adalah untuk mengambil

komponen unit diuji dan membangun program terstruktur yang telah ditentukan pada tahap desain (Pressman,2010). Pengujian ini akan memastikan apakah komponen-komponen dalam sistem atau perangkat lunak berinteraksi dengan cara yang sesuai diharapkan. Pengujian integrasi terdiri dari beberapa strategi yaitu strategi top-down dan strategi bottom-up.

Pengujian integrasi top-down adalah pendekatan inkremental untuk pembangunan arsitektur perangkat lunak (Pressman ,2010). Modul terintegrasi dengan memindahkan ke bawah melalui control hirarki, yaitu dimulai dengan modul utama (program utama). Sedangkan strategi yang lain adalah strategi bottom-up. Startegi bottom-up dimulai dari konstruksi dan pengujian dengan modul atom (yaitu., komponen di tingkat terendah dalam struktur program). Karena komponen yang terintegrasi dari bawah ke atas (Pressman,2010).

2.6 Pola Perancangan (*Design Pattern*)

Dilingkungan sekitar kita telah banyak menemukan banyak pola seperti pola rumah, pola gedung, pola pintu dan sebagainya. Hal itu juga terjadi dalam mendesain sebuah perangkat lunak. Perlu adanya pola dalam merancang sebuah perangkat lunak untuk efisiensi sumber daya. Selain itu proses desain juga lebih mudah karena pengembang hanya mengikuti pola desain yang sudah ada. Menurut Gamma dkk. (1995) secara umum, pola memiliki empat elemen penting yaitu, nama pola, masalah (*problem*), solusi dan konsekuensi. Macam-macam pola terdiri dari 23 pola yaitu *abstract factory*, *adapter*, *bridge*, *builder*, *chain of responbility*, *command*, *composite*, *decorator*, *facade*, *factory method*, *flyweight*, *interpreter*, *iterator*, *mediator*, *memento*, *observer*, *prototype*, *proxy*, *singleton*, *state*, *strategy*, *template method*, dan *visitor*. Deskripsi dari 23 pola tersebut dapat dilihat pada Tabel 2.2 berikut,

Tabel 2.2 Deskripsi macam pola perancangan

No	Nama Pola	Deskripsi
1	<i>Abstract Factory</i>	Menyediakan sebuah antarmuka untuk menciptakan keluarga yang terkait atau tergantung objek tanpa menentukan kelas asli mereka.
2	<i>Adapter</i>	Mengkonversi antarmuka dari sebuah kelas menjadi antarmuka lain yang klien harapkan.Kelas adapter tidak bekerja sama karena interface tidak kompatibel
3	<i>Bridge</i>	Memisahkan suatu abstraksi dari pelaksanaannya sehingga dua objek bervariasi secara independen.
4	<i>Builder</i>	Memisahkan pembangunan obyek yang kompleks dari perwakilannya sehingga proses konstruksi yang sama dapat membuat representasi yang berbeda.
5	<i>Chain of Responsibility</i>	Hindari kopling pengirim dari permintaan untuk penerima dengan memberikan lebih dari satu objek kesempatan untuk menangani permintaan.

Tabel 2.2 Deskripsi macam pola perancangan (lanjutan)

No	Nama Pola	Deskripsi
6	<i>Command</i>	Merangkul permintaan sebagai objek, sehingga membiarkan parameter klien dengan permintaan yang berbeda, antrian atau permintaan log.
7	<i>Composite</i>	Menulis objek ke dalam struktur pohon untuk mewakili bagian-keseluruhan hirarki. Komposit memungkinkan klien memperlakukan objek individu dan komposisi benda seragam.
8	<i>Decorator</i>	Melampirkan tanggung jawab tambahan untuk sebuah objek dinamis. Dekorator memberikan alternatif fleksibel untuk subclassing untuk memperluas fungsionalitas.
9	<i>Facade</i>	Menyediakan antarmuka yang seragam untuk satu set interface dalam subsistem. Penglihatan mendefinisikan antarmuka tingkat tinggi yang membuat subsistem lebih mudah digunakan.
10	<i>Factory Method</i>	Mendefinisikan interface untuk membuat sebuah objek, tapi membiarkan subclass memutuskan kelas mana yang akan dibuat. Metode pabrik memungkinkan kelas defer menginstansiasi untuk subclass
11	<i>Flyweight</i>	Penggunaan memori/data secara bersama dengan objek sejenis.
12	<i>Interpreter</i>	Tafsir bahasa, mendefinisikan representasi untuk tata bahasa sama dengan interpreter yang menggunakan representasi untuk menafsirkan kalimat dalam bahasa.
13	<i>Iterator</i>	Menyediakan cara untuk mengakses elemen objek agregat berurutan tanpa memaparkan representasi yang mendasarinya.
14	<i>Mediator</i>	Mendefinisikan suatu objek yang merangkul bagaimana satu set objek berinteraksi.
15	<i>Memento</i>	Tanpa melanggar enkapsulasi, menangkap dan eksternalisasi obyek keadaan internal sehingga objek dapat dikembalikan ke keadaan ini di kemudian hari.
16	<i>Observer</i>	Mendefinisikan ketergantungan satu ke banyak antara objek sehingga ketika satu objek berubah, semua perubahan akan diberitahu dan diperbarui secara otomatis (broadcast).
17	<i>Prototype</i>	Tentukan jenis objek untuk membuat menggunakan instance prototipikal, dan membuat objek baru dengan menyalin prototipe ini.
18	<i>Proxy</i>	Menyediakan pengganti atau tempat untuk objek lain untuk mengontrol akses kesana.

Tabel 2.2 Deskripsi macam pola perancangan (lanjutan)

No	Nama Pola	Deskripsi
19	<i>Singleton</i>	Memastikan kelas hanya memiliki satu contoh, dan memberikan titik global akses ke sana.
20	<i>State</i>	Memungkinkan sebuah objek untuk mengubah perilakunya ketika ada perubahan <i>state</i> internal. Objek akan muncul untuk mengubah kelasnya.
21	<i>Strategy</i>	Mendefinisikan keluarga algoritma, merangkum masing-masing, dan membuat mereka dipertukarkan. Strategi memungkinkan algoritma bervariasi secara independen dari klien yang menggunakannya.
22	<i>Template Method</i>	Mendefinisikan kerangka algoritma dalam sebuah operasi, menunda beberapa langkah-langkah untuk subclass. Metode Template memungkinkan subclass mendefinisikan kembali langkah-langkah tertentu algoritma terbuka tanpa mengubah struktur algoritma.
23	<i>Visitor</i>	Merupakan operasi yang akan dilakukan pada unsur-unsur dari sebuah objek struktur. Pengunjung memungkinkan Anda menentukan operasi baru tanpa mengubah kelas elemen yang beroperasi.

Sumber : Gamma dkk. (1995)

Pola perancangan sangat bervariasi berdasarkan rincian masing-masing. Karena banyaknya pola perancangan maka dibutuhkan pengelolaan yaitu dengan mengklasifikasikan sehingga muncul kategori. Selain membantu pembelajaran pengkategorian ini juga membantu upaya untuk menemukan pola baru. Gamma dkk. (1995) mengelompokkan pola perancangan menjadi tiga yaitu pola *reational*, *structural* dan *behaviour*. Pola *creational* menyangkut proses pembuatan obyek. Pola *structural* berhubungan dengan komposisi kelas atau objek. Pola *behaviour* mencirikan cara dimana kelas atau benda berinteraksi dan mendistribusikan *responsibility*. Pengelompokan pola-pola dapat dilihat pada Tabel 2.3 tersebut,

Tabel 2.3 Pengelompokan pola perancangan

Cakupan	Tujuan		
	Creational	Structural	Behaviour
Kelas	<i>Factory Method</i> Adapter	Adapter	Interpreter Template Method
Objek	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

Cakupan atau ruang lingkup menentukan apakah pola lebih berperilaku untuk kelas atau objek. Pola kelas berurusan dengan hubungan antara kelas dan subclass. Hubungan ini dibentuk melalui warisan, sehingga pola lebih statis-tetap pada saat kompilasi. Sedangkan pola objek menangani hubungan objek, yang dapat diubah saat run-time dan lebih dinamis. Hampir semua pola menggunakan warisan sampai batas tertentu. Jadi satu-satunya pola berlabel "kelas pola" adalah pola yang fokus pada hubungan kelas (Gamma dkk, 1995).

2.7 Perancangan *Factory Pattern*

Fase perancangan adalah tahap dimana mendesain sistem yang akan dibangun sesuai dengan analisis kebutuhan yang telah dilakukan. Beberapa pola perancangan adalah decorator, facade, factory, adapter, singleton, observer dan adapter dan lain-lain (Freeman dkk, 2004). Dari berbagai pola tersebut dikelompokkan menjadi tiga kategori yaitu *creational*, *structure*, dan *behaviour* (Freeman dkk, 2004). Pola perancangan factory merupakan pola perancangan yang memiliki komponen atau sistem seperti layaknya sebuah pabrik dan termasuk dalam kategori pola *creational*. Komponen-komponen pada *factory pattern* terdiri dari *creator*, *concrete creator*, *product*, dan *concrete product*.

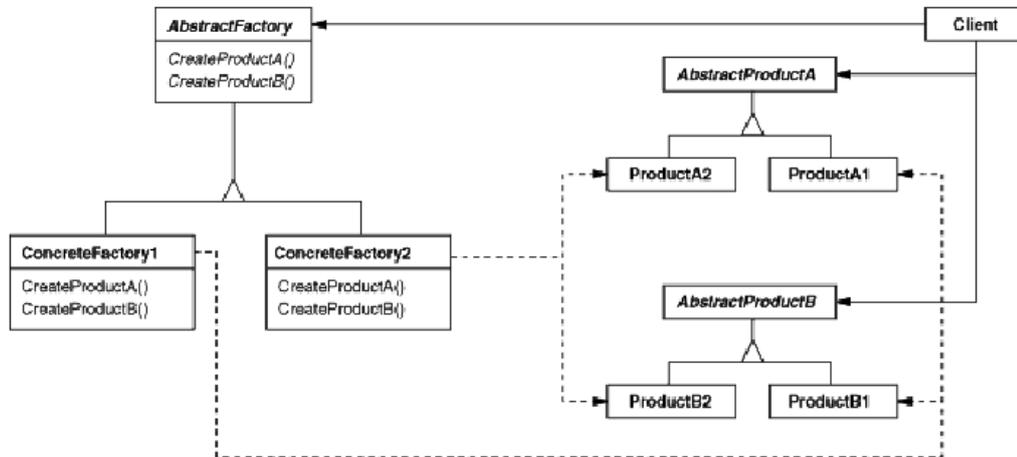
Factory Pattern merupakan pola yang digunakan untuk memisahkan (*decouple*) proses pembuatan/instansiasi sebuah produk atau objek dari produk lain (klien) yang menggunakannya. Tujuannya adalah agar perubahan pada *product* class tidak menyebabkan kita harus mengubah kode pada client. Paling tidak akibat dari perubahan itu bisa diminimalisir. Dan juga agar komponen factory dapat digunakan oleh banyak class. Factory merupakan objek yang berfungsi membuat objek lain (produk). Kelas ini menyembunyikan proses pembuatan produk dari klien sehingga klien tidak perlu tahu proses pembuatannya, bahkan klien juga tidak perlu tahu nama kelas dari produk yang dia minta. Factory pattern memiliki dua jenis yaitu *abstract factory* dan *factory method*.

2.7.1 *Abstract Factory*

Pola *abstract factory* menyediakan sebuah antarmuka agar klien dapat memperoleh contoh kelas (objek) sesuai dengan antarmuka atau protokol tertentu tanpa harus mengetahui secara jelas kelas apa yang mereka dapatkan. Metode ini memiliki kelebihan untuk enkapsulasi dan penggunaan ulang kode program, karena implementasi dapat dimodifikasi tanpa mengharuskan setiap perubahan kode pada sisi klien. Pabrik juga dapat digunakan untuk mengelola alokasi dan inisialisasi proses, karena pabrik tidak perlu selalu mengalokasikan objek baru setiap kali ada permintaan klien (Ellis dkk., 2007).

Menurut Gamma dkk. (1995) penerapan pola *abstract factory* adalah ketika (1) sistem harus independen bagaimana produk dibuat, disusun, dan digambarkan. (2) Sistem harus dikonfigurasi dengan salah satu *family* produk. (3) Keluarga dari objek produk terkait adalah didesain untuk digunakan bersama, dan dibutuhkan untuk memberlakukan batasan. (4) Pengembang ingin menyediakan

perpustakaan kelas produk dan hanya ingin menggunakan antarmuka, bukan implementasinya. Untuk lebih jelasnya struktur dari pola *abstract factory* dapat dilihat pada Gambar 2.4 berikut,



Gambar 2.4 Struktur pola *abstract factory*

Sumber : (Gamma dkk, 1995)

Komponen-komponen dalam pola *abstract factory*,

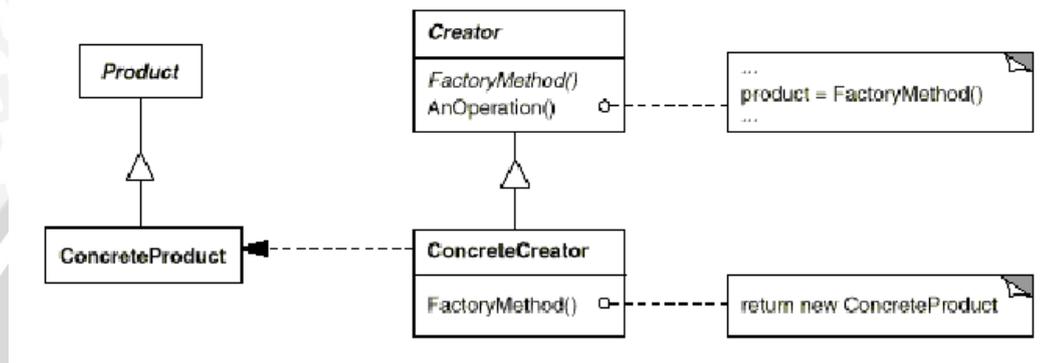
- *Abstract Factory*
Menyatakan sebuah antarmuka untuk operasi menciptakan produk abstrak benda.
- *Concrete Factory*
Mengimplementasikan operasi untuk membuat objek produk asli atau nyata.
- *Abstract Product*
Menyatakan jenis forum antarmuka objek produk.
- *Concrete Product*
Mendefinisikan objek produk yang akan dibuat oleh kelas asli yang sesuai pabrik. Mengimplementasikan interface Produk Abstrak.
- Client
Hanya menggunakan interface yang dinyatakan oleh Pabrik Abstrak dan Kelas abstrak produk.

2.7.2 *Factory Method*

Pola *factory method* memiliki hubungan dengan pola *abstract factory* namun pola ini lebih sederhana, pola *factory method* memungkinkan klien untuk mendapatkan objek yang tidak diketahui kelas yang mengimplementasikan antarmuka tertentu. Daripada mengandalkan kelas pabrik terpisah untuk membuat turunan dari kelas produk, kelas produk itu sendiri memiliki metode pabrik yang mengembalikan sebuah objek yang sesuai dengan antarmuka yang didefinisikan oleh kelas itu. Biasanya, kelas menerapkan pola metode pabrik akan

menjadi kelas abstrak dengan beberapa subclass konkret dan akan menyajikan sebuah metode statis (Ellis dkk, 2007).

Menurut Gamma dkk. (1995) penerapan pola *factory method* adalah ketika (1) Kelas tidak bisa mengantisipasi kelas dari objek yang akan diciptakan. (2) Kelas menginginkan subclass untuk menentukan objek yang dibuat. (3) Kelas memberikan tanggung jawab untuk salah satu dari beberapa helper subclass dan pengembang ingin melokalisasikan pengetahuan yang telah diberikan kepada helper subclass. Untuk lebih jelasnya struktur dari pola *factory method* dapat dilihat pada Gambar 2.5 berikut,



Gambar 2.5 Struktur pola *factory method*

Sumber : (Gamma dkk, 1995)

Komponen-komponen pada pola pada Gambar 2.5 tersebut adalah sebagai berikut,

- *Product*
Mendefinisikan interface dari objek *factory method* yang diciptakan
- *Concrete Product*
Menimplementasikan interface *product*.
- *Creator*
Mendeklarasi *factory method*, yang mengembalikan sebuah objek dari tipe produk. *Creator* juga dapat menentukan implementasi standart metode pabrik untuk membuat objek produk.
- *Concrete Creator*
Menimpa (*overrides*) metode pabrik untuk mengembalikan sebuah instance dari *concrete product*.

2.8 Unified Modelling Language

Menurut Melor, Balcer & Foreword (2002) Unified Modeling Language adalah bahasa untuk spesifikasi, membangun, visualisasi, danmendokumentasikan artefak dari sistem perangkat lunak secara intensif . Sedangkan menurut Pressman (2010) Unified Modelling Language (UML) adalah bahasa standar untuk

penulisan blue print perangkat lunak. UML dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. Dilain kata, seperti arsitek bangunan membuat blue print yang akan digunakan oleh konstruksi perusahaan, arsitek software membuat diagram UML untuk membantu pengembang perangkat lunak membangun perangkat lunak. Jika anda memahami kosakata UML (diagram ,elemen bergambar dan artinya), anda dapat lebih mudah memahami dan menentukan sistem dan menjelaskan desain sistem kepada orang lain

Beberapa jenis diagram yang berfungsi dalam memetakan bahasa pemrograman tertentu, adalah sebagai berikut :

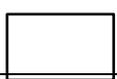
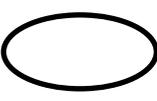
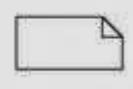
2.8.1 Diagram *Use Case*

Diagram *use case* merupakan sebuah diagram yang menunjukkan hubungan antara aktor dan kasus penggunaan dalam sistem (Booch dkk, 1999). Sebuah use case menggambarkan serangkaian kegiatan yang dilakukan untuk menghasilkan beberapa output. Setiap use case menggambarkan bagaimana pengguna eksternal memicu suatu event yang harus segera direspon oleh sistem. Sebagai contoh, dalam sistem peminjaman pada rental video, pelanggan bisa menyewa video atau mengembalikan video yang telah disewa, tetapi ada potensi bahwa video yang dikembalikan disewa mungkin mengalami keterlambatan pada saat pengembalian. Ketika terjadi event yaitu terjadi keterlambatan pada pengembalian video yang dipinjam maka respon dari sistem adalah menghitung banyaknya denda yang harus dibayar oleh penyewa. Tindakan menyewa atau mengembalikan video serta semua peristiwa yang memicu serangkaian kegiatan yang harus dilaksanakan oleh sistem. Event dalam sistem dapat dianggap sebagai respon terhadap suatu peristiwa yang menjadi pemicu dari suatu kegiatan. Bila tidak ada event / peristiwa pada sistem maka sistem berada pada kondisi istirahat, pada kondisi ini sistem dengan sabar menunggu adanya trigger untuk memicu suatu kegiatan. Ketika trigger terjadi, sistem (dan orang-orang menggunakannya) merespon, dengan melakukan tindakan yang didefinisikan dalam *use case*, dan kemudian kembali ke status menunggu (wait).

Use case dibuat untuk membantu dalam memahami situasi dan membantu dalam menyederhanakan langkah-langkah pemodelan selanjutnya. Harus dilakukan pembuatan use case setiap kali terjadi proses rekayasa ulang atau terjadi perubahan pada proses bisnis yang secara signifikan akan mengubah cara orang bekerja. Ingat bahwa *use case* menggambarkan sebuah sistem serta fungsinya dari perspektif pengguna. Oleh karena itu, penting untuk melibatkan pengguna dalam penciptaan *use case* serta mengkomunikasikannya ke seluruh pengguna sehingga pengguna memahami kegiatan yang direncanakan untuk sistem baru. Keterlibatan pengguna juga membantu untuk memastikan bahwa tidak ada langkah penting atau tugas yang dihilangkan dari perancangan use case (Suprpto, Aditya Rachmadi, Akbar dalam Aulia , 2014).

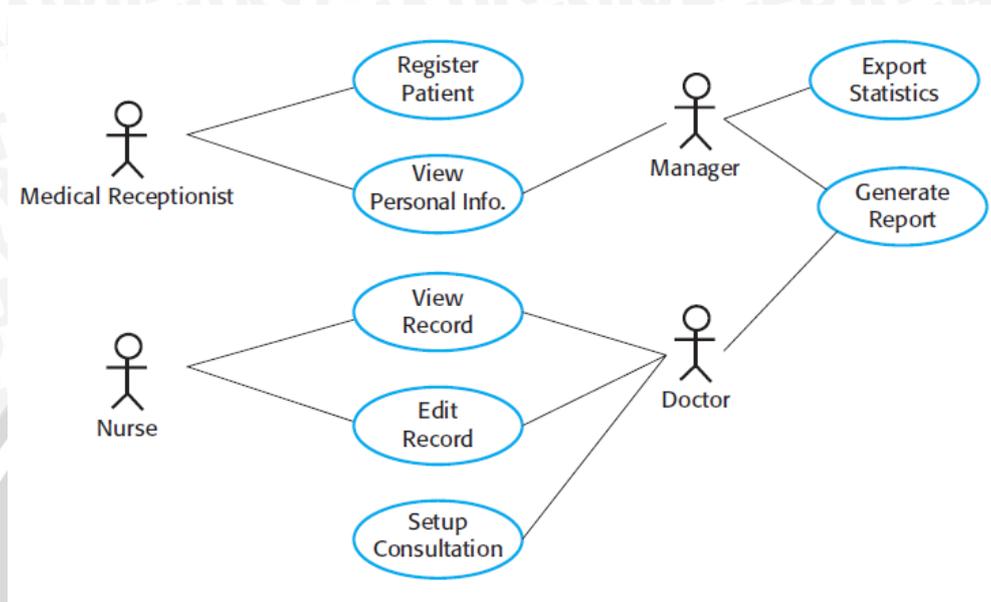
Menurut Aulia (2014) komponen-komponen dari diagram *use case* adalah aktor, *use case* dan asosiasi atau relasi. Komponen *use case* untuk lebih jelasnya dapat dilihat pada Tabel 2.4 berikut,

Tabel 2.4 Komponen *use case* diagram

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber : (Rosa dan Shalahuddin, 2014).

Selain menggambarkan proses dalam bentuk diagram, *use case* diagram juga disertai dengan tabel yang menjelaskan rangkaian isi tiap *use case*. Dalam tabel tersebut dijelaskan beberapa hal, diantaranya adalah aktor mana yang menggunakan *use case* apa dan *use case* mana yang memasukkan *use case* lain. Contoh *use case* diagram dapat dilihat pada Gambar 2.6 berikut :



Gambar 2.6 Contoh *use case* diagram

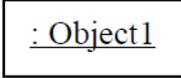
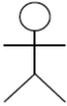
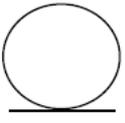
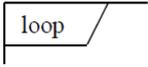
Sumber : (Sommerville, 2011)

Dalam diagram *use case* tersebut ada empat aktor yaitu Medical Receptionist, Nurse, Export Statistic dan Doctor. Setiap aktor mempunyai fungsi sendiri-sendiri atau memiliki beberapa fungsi yang sama dengan aktor yang lain seperti Doctor dan Nurse sama-sama memiliki fungsi View Record dan Edit Record.

2.8.2 Diagram Sequence

Diagram sequence digunakan untuk menelusuri eksekusi skenario dalam konteks yang sama sebagai diagram komunikasi (Booch, 1991). Diagram sequence adalah jenis diagram interaksi yang sering digunakan dimana didalamnya menunjukkan bagaimana aliran data saling berinteraksi dalam beberapa behavior. Keuntungan dari menggunakan dari diagram sewuence adalah lebih mudah untuk membaca pesan yang lewat. *Sequence diagram* lebih baik daripada diagram objek dalam hal menangkap semantik skenario awal siklus pengembangan, sebelum protokol kelas individu telah diidentifikasi (Booch, 1991). Diagram sequence akan menunjukkan sejumlah objek contoh dan pesan yang melewati objek-objek ini dalam *use case*. Komponen-komponen *sequence diagram* dapat dilihat pada Tabel 2.5 berikut (Aulia, 2014),

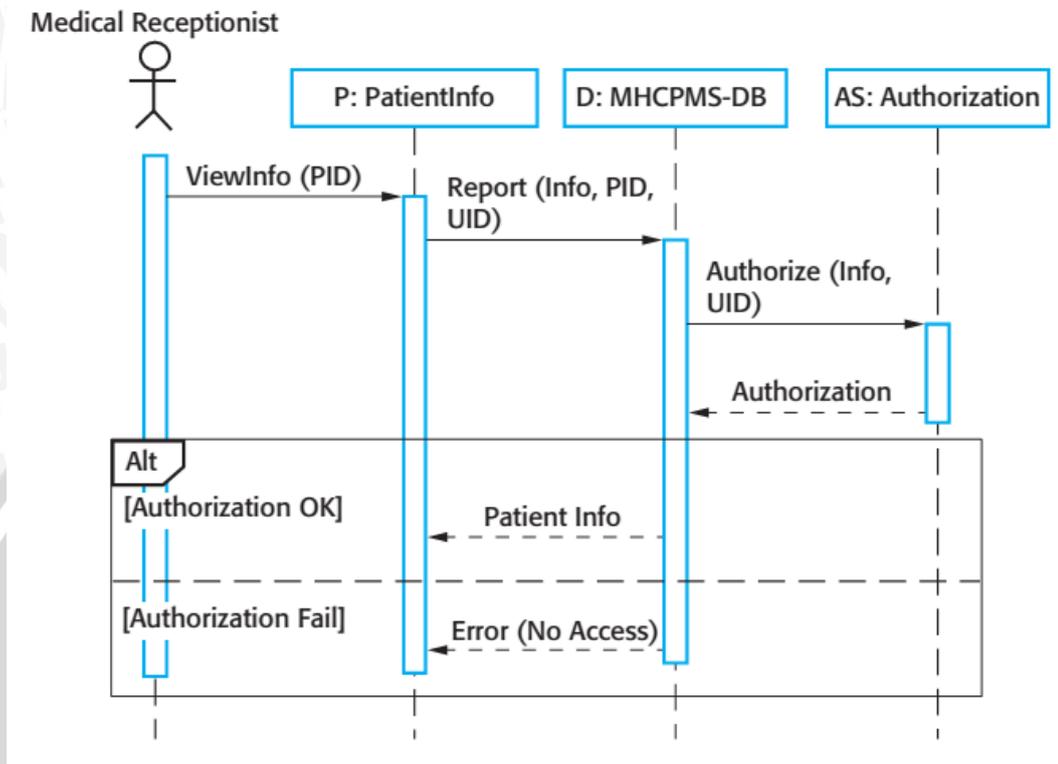
Tabel 2.5 Komponen *sequence diagram*

No	Komponen	Nama Komponen	Keterangan
1.		<i>Object</i>	Merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma.
2.		Actor	Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom.
3.		<i>Lifeline</i>	Mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah objek.
4.		<i>Activation</i>	Dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah lifeline. Activation mengindikasikan sebuah objek yang akan melakukan sebuah aksi.
5.		<i>Boundary</i>	Boundary terletak di antara sistem dengan dunia sekelilingnya. Semua form, laporan-laporan, antar muka ke perangkat keras seperti printer atau scanner dan antar muka ke sistem lainnya adalah termasuk dalam kategori.
6.		<i>Control</i>	Control berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi, atau penanganan kesalahan.
7.		<i>Entity</i>	Entity digunakan menangani informasi yang mungkin akan disimpan secara permanen. Entity bisa juga merupakan sebuah tabel pada struktur basis data.
8.		<i>Message</i>	Message, digambarkan dengan anak panah horizontal antara Activation. Message mengindikasikan komunikasi antara object-object.
9.		<i>Self-message</i>	Self-message atau panggilan mandiri mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri.
10.		<i>Loop</i>	Operator loop adalah <i>fragmen</i> yang dapat mengeksekusi berulang kali dan penjaga menunjukkan dasar iterasi.

Sumber : (Ulum dalam Aulia, 2010)



Dari komponen-komponen tersebut dapat dibangun diagram behaviour yang mewakili masing-masing usecase pada diagram usecase. Contoh diagram sequence dapat dilihat pada Gambar 2.7 berikut,



Gambar 2.7 Contoh *sequence diagram*

Sumber : (Sommerville, 2011)

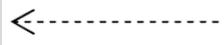
Pada diagram sequence tersebut aktor Medical Receptionist memicu metode ViewInfo dalam instan P dari objek PatientInfo dengan membawa parameter pengenalan (PID). P merupakan objek antarmuka pengguna dengan menampilkan informasi pasien. Kemudian P memanggil database untuk mengambil informasi yang diperlukan. Sebelum memberikan informasi objek D mengotorisasi pada objek AS dengan menggunakan parameter UID dan Info. Pada tahap selanjutnya terdapat dua alternatif yaitu dimana ketika otorisasi berhasil maka D mengembalikan info pasien atau jika otorisasi gagal maka D akan mengembalikan Error (tidak bisa mengakses).

2.8.3 Diagram Kelas

Diagram kelas adalah presentasi grafis dari pandangan statis yang menunjukkan koleksi dari deklaratif (statis) elemen model, seperti kelas, jenis, dan isinya serta hubungannya (Booch dkk, 1999). Diagram kelas merupakan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat diantara mereka. *Class diagram* juga menunjukkan property dan operasi sebuah class dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut (Fowler, 2004

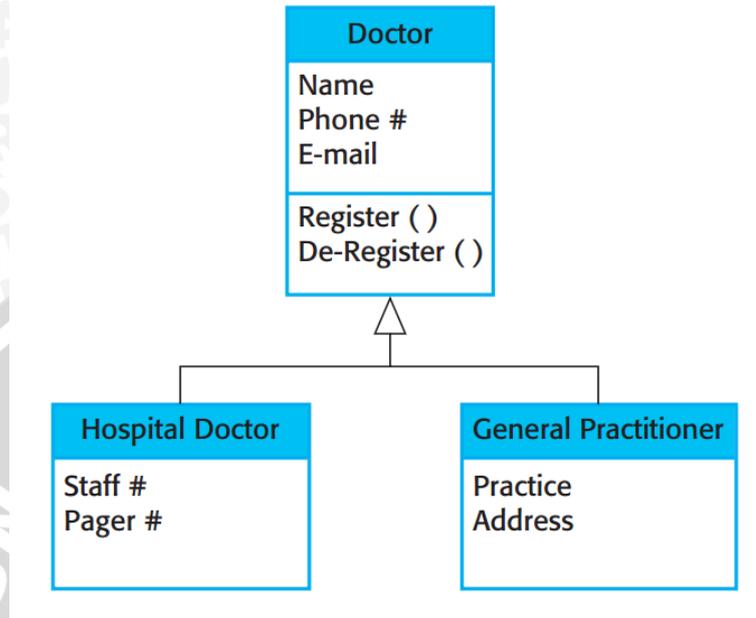
dalam Aulia, 2014). Detail tentang komponen *class diagram* dapat dilihat pada Tabel 2.6 berikut,

Tabel 2.6 Komponen *class diagram*

No	Komponen	Nama Komponen	Keterangan						
1.	<table border="1"> <tr> <td>Nama <i>Class</i></td> </tr> <tr> <td>- atribut</td> </tr> <tr> <td>- atribut</td> </tr> <tr> <td>- atribut</td> </tr> <tr> <td>+ <i>method</i></td> </tr> <tr> <td>+ <i>method</i></td> </tr> </table>	Nama <i>Class</i>	- atribut	- atribut	- atribut	+ <i>method</i>	+ <i>method</i>	<i>Class</i>	Merupakan blok – blok pembangun pada pemrograman berorientasi objek. Sebuah class digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari class. Bagian tengah mendefinisikan property/atribut class. Bagian akhir mendefinisikan method-method dari sebuah class.
Nama <i>Class</i>									
- atribut									
- atribut									
- atribut									
+ <i>method</i>									
+ <i>method</i>									
2.	<u>1..n Owned by 1</u>	<i>Assosiation</i>	Asosiasi merupakan sebuah <i>relationship</i> paling umum antara 2 class, dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 class. Garis ini bisa melambangkan tipe-tipe <i>relationship</i> .						
3.		<i>Compositio n</i>	<i>Composition</i> merupakan kondisi dimana suatu kelas bergantung pada kelas yang lain. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi / <i>solid</i> .						
4.		<i>Dependenc y</i>	<i>Dependency</i> merupakan kondisi dimana suatu kelas menggunakan kelas lain. Penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. <i>Dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.						
5.		<i>Agregratio n</i>	<i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi “mempunyai sebuah” atau “bagian dari”. Sebuah <i>aggregation</i> digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi/tidak solid.						
6.		<i>Generalizat ion</i>	Sebuah relasi <i>generalization</i> sepadan dengan sebuah relasi <i>inheritance</i> pada konsep berorientasi objek. Sebuah <i>generalization</i> dilambangkan dengan sebuah panah dengan kepala panah yang tidak solid yang mengarah ke kelas “ <i>parent</i> ”- nya/induknya.						

Sumber : (Ulum dalam Aulia, 2010)

Dari komponen-komponen tersebut maka dapat dibentuk sebuah kelas diagram pada sistem tertentu. Untuk contoh dari *class diagram* dapat dilihat pada Gambar 2.8 berikut,



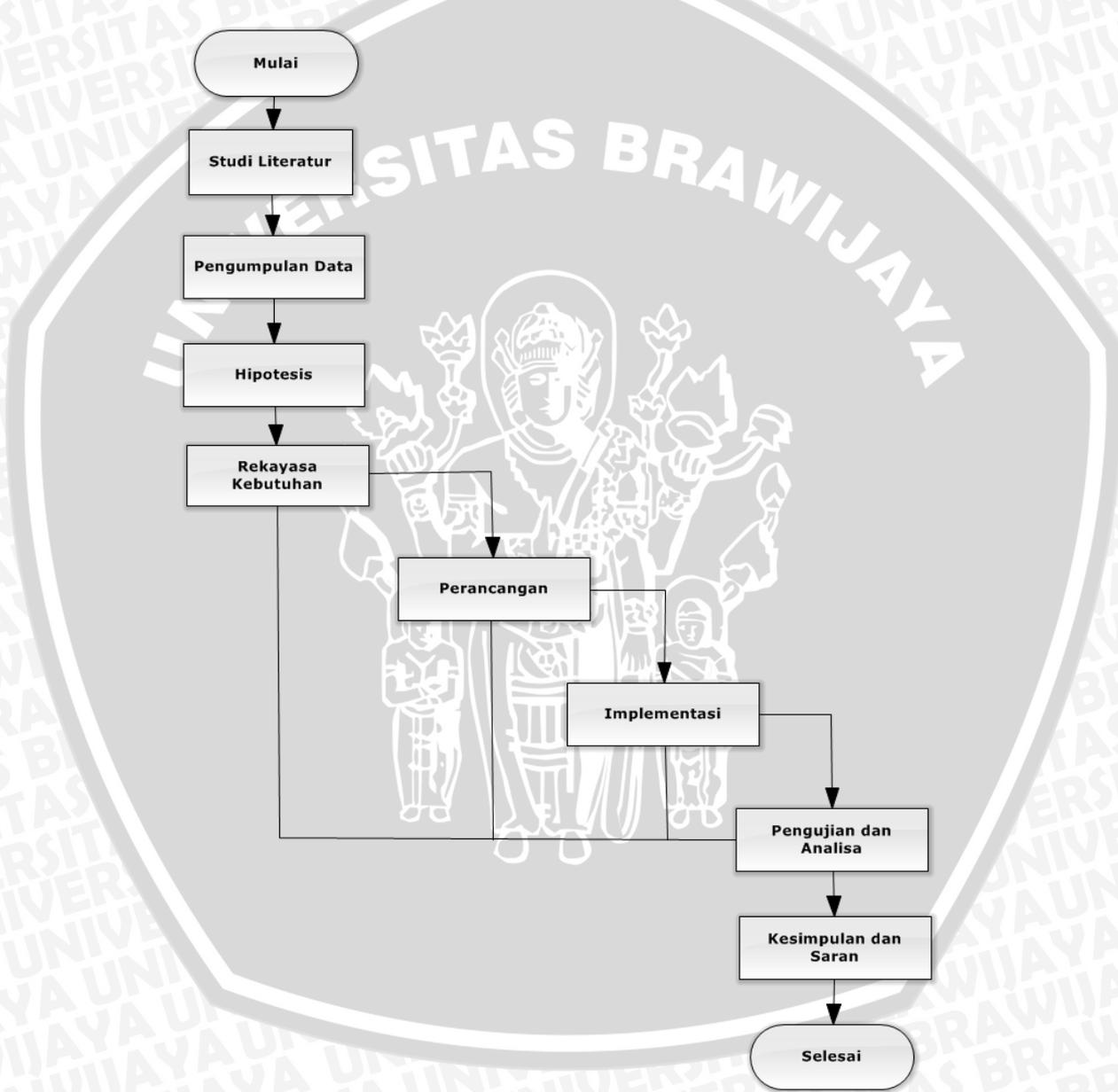
Gambar 2.8 contoh *class diagram*

Dalam contoh *class diagram* pada Gambar 2.8 tersebut terdiri dari kelas Hospital Doctor, general Practitioner yang generalisasi (*extends*) terhadap kelas Doctor. Generalisasi berarti inherit atau bermakna sebagai anak kelas (*is-a*) terhadap kelas orang tua yaitu kelas Doctor (Sommerville,2011).



BAB 3 METODOLOGI

Metodologi penelitian menjelaskan metode yang digunakan dalam pembuatan SIPKIA dengan menggunakan model waterfall. Tahapan metodologi terdiri dari studi literatur, pengumpulan data, rekayasa kebutuhan, perancangan, implementasi, pengujian dan analisa, hipotesis, serta pengambilan kesimpulan. Untuk lebih jelasnya diagram alur metodologi penelitian dapat dilihat pada Gambar 3.1 berikut,



Gambar 3.1 Diagram metodologi penelitian



3.1 Studi Literatur

Mempelajari literatur dari beberapa bidang ilmu yang berhubungan dengan pembuatan Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method*, diantaranya :

- Sistem Informasi
Sistem informasi sebagai program yang dapat mengelola data secara digital melalui perantara komputer yang terisi logika-logika pemrograman.
- Register Kohort
Buku catatan rekam medis dari ibu dan anak yang meliputi register kohort ibu, bayi, balita dan anak.
- Pemetaan Kesehatan Ibu dan Anak
Pemetaan Kesehatan Ibu dan Anak menggunakan program pemerintah Pemantauan Wilayah Setempat Kesehatan Ibu dan Anak (PWS-KIA) dari departemen kesehatan Indonesia.
- Pola Perancangan *Factory Method*
Pola perancangan *factory method* merupakan perancangan yang betripe *creational* pada pola perancangan yang ada. *Factory method* memiliki komponen-komponen *creator*, *concrete creator*, *product*, dan *concrete product*.
- Model pengembangan waterfall
Model pengembangan waterfall merupakan model yang paling lama digunakan dalam pengembangan perangkat lunak. Waterfall memiliki kelebihan yaitu efisiensi dalam pembangunan perangkat lunak karena persyaratan telah dipahami dengan baik dan tidak mungkin berubah secara radikal selama pengembangan sistem.
- Rekayasa Perangkat Lunak
Sebuah bidang profesi yang mendalami cara-cara pengembangan perangkat lunak termasuk pembuatan, pemeliharaan, manajemen organisasi pengembanganan perangkat lunak dan manajemen kualitas.
- Pemrograman bahasa PHP dan framework Codeigniter
Implementasi SIPKIA berbasis web dengan menggunakan framework Codeigniter yang menggunakan bahasa PHP.
- DBMS MySQL
Manajemen data yang digunakan yaitu menggunakan MySql yang support pada framework codeigniter.
- Proses Pengujian Sistem
Pengujian sistem yang dilakukan berupa pengujian unit, pengujian integrasi, pengujian validasi dan pengujian pattern.

Literatur tersebut didapatkan dari buku, jurnal, e-book, penelitian sebelumnya dan dokumentasi projek.

3.2 Pengumpulan Data

Pengumpulan data mempunyai peranan yang sangat penting dalam menyelesaikan skripsi ini karena data yang berkaitan dengan informasi kesehatan ibu dan anak pada bidan desa. Pengumpulan data dilakukan kepada petugas puskesmas kecamatan tumpang dan bidan desa di kecamatan Tumpang. Data pada penelitian ini menggunakan data pertanyaan-pertanyaan yang dapat menentukan kebutuhan sistem informasi pada bidan desa yang terintegrasi di puskesmas kecamatan Tumpang. Pengumpulan data dilakukan dengan dua cara yaitu :

- Metode Pengamatan Langsung (Observasi)
Metode Pengamatan Langsung (Observasi) adalah metode pengambilan data yang dilakukan dengan datang langsung ke tempat tujuan yang diteliti, seperti datang secara langsung untuk melihat dan mengamati tentang informasi kesehatan ibu dan anak di kecamatan Tumpang.
- Metode Wawancara
Metode wawancara adalah metode yang digunakan untuk melakukan wawancara secara langsung kepada pihak terkait seperti kepada petugas puskesmas Tumpang maupun bidan desa di kecamatan Tumpang.

3.3 Hipotesis

Hipotesis atau dugaan sementara pada pembangunan SIPKIA menggunakan pola perancangan *factory pattern* terdiri dari beberapa hipotesis. Berdasarkan pengamatan pada penelitian sebelumnya pada pembangunan SIPKIA menggunakan pola perancangan *factory method* akan memudahkan dalam hal pengembangan karena dibangun menggunakan *design pattern*. Kemudahan tersebut tidak hanya dalam pengembangan kedepannya namun juga akan terasa ketika proses pembangunan SIPKIA yaitu ketika komponen satu telah selesai dan melanjutkan ke komponen lainnya. Karena adanya pola tersebut diharapkan akan mempercepat proses pembangunan SIPKIA dan target yang direncanakan akan tercapai. Selain itu dari sisi usability SIPKIA akan sangat membantu pekerjaan bidan desa dan pegawai puskesmas.

3.4 Rekayasa Kebutuhan

Pada tahap ini dilakukan rekayasa kebutuhan Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method*. Kegiatan analisis kebutuhan bertujuan untuk mengetahui kebutuhan yang diperlukan untuk membangun sistem. Langkah ini dilakukan agar sistem yang dibuat dapat sesuai dengan kebutuhan pengguna. Proses analisis kebutuhan meliputi beberapa tahap elisitasi perangkat lunak, spesifikasi perangkat lunak, validasi dan verifikasi, serta manajemen kebutuhan perangkat lunak. Proses analisis kebutuhan dilakukan dengan metode *Object*

Oriented Analysis dan pemodelan dilakukan menggunakan UML (*Unified Modelling Language*).

3.5 Perancangan Sistem

Perancangan sistem yang akan dibuat berdasarkan analisa kebutuhan sistem yang telah ditentukan sebelumnya seperti merancang pola *factory method*, arsitektur sistem, pemodelan kelas, basis data, perancangan komponen dan perancangan antarmuka pengguna.

3.6 Implementasi Sistem

Implementasi sistem adalah fase membangun sistem yang mengacu pada perancangan sistem dan menerapkan hal yang telah didapatkan dalam proses studi literatur. Fase – fase yang ada dalam implementasi antara lain:

- Implementasi interface, menggunakan software Sublime text 2
- Implementasi basis data, dengan menggunakan sebuah DBMS My SQL pada Server localhost (XAMPP) yang bertujuan untuk memudahkan melakukan manipulasi dan penyimpanan data.
- Implementasi ini akan menghasilkan sebuah sistem informasi dan pemetaan kesehatan ibu dan anak yang terintegrasi dengan puskesmas setempat.

3.7 Pengujian dan Analisa

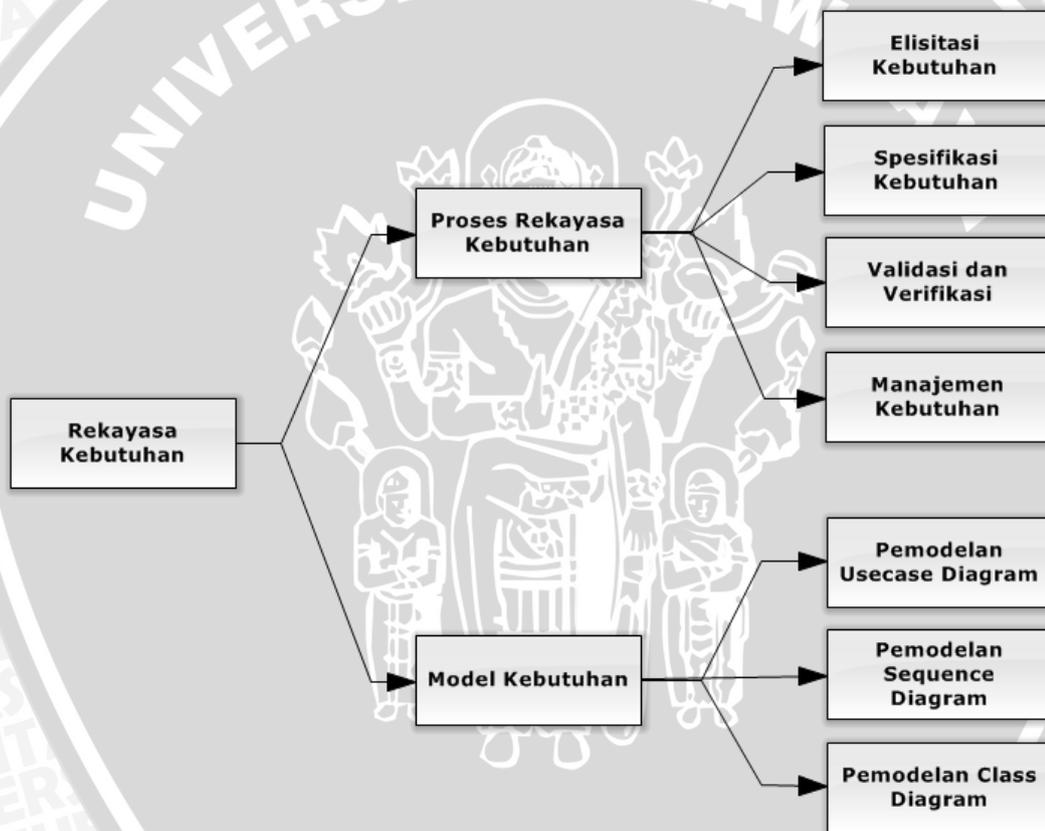
Uji coba sistem dilakukan untuk mengetahui apakah sistem berjalan dengan baik dan sesuai dengan spesifikasi kebutuhan yang telah ditetapkan dapat berjalan dengan baik. Pengujian tersebut terdiri dari pengujian unit, pengujian integrasi, pengujian validasi dan selain itu juga menguji dengan pattern yang digunakan dalam merancangan sistem. Analisa yang dilakukan meliputi pengujian unit, pengujian integrasi, pengujian validasi dan pengujian pattern. Pengujian unit merupakan pengujian terhadap algoritma yang ada dalam sistem dengan menghitung cyclomatic complexity. Pengujian integrasi merupakan gabungan dari pengujian unit. Pengujian validasi berfungsi untuk menguji setiap fungsional sistem sesuai dengan analisis kebutuhan yang dibuat. Analisa pengujian pattern yaitu hasil analisis jika adanya pengembangan dalam sistem yang berhubungan dengan pola pattern tersebut.

3.8 Pengambilan Kesimpulan

Kesimpulan dilakukan setelah semua tahap analisa kebutuhan, perancangan, implementasi dan pengujian serta metode yang diterapkan sudah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis model. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan yang terjadi serta memberikan pertimbangan untuk pengembangan selanjutnya.

BAB 4 REKAYASA KEBUTUHAN

Bab ini akan membahas proses analisa kebutuhan Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method*. Analisa kebutuhan perangkat lunak dibuat untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna. Dalam rekayasa kebutuhan terbagi menjadi proses rekayasa kebutuhan dan model kebutuhan. Proses rekayasa kebutuhan terdiri dari elisitasi, spesifikasi, validasi dan verifikasi dan manajemen kebutuhan. Pemodelan kebutuhan terdiri dari pemodelan *usecase*, *sequence* dan *class diagram*. Tahap-tahap rekayasa kebutuhan Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method* dapat dilihat pada gambar 4.1 berikut,



Gambar 4.1 Diagram rekayasa kebutuhan

4.1 Proses Rekayasa Kebutuhan

Proses rekayasa kebutuhan merupakan tahap untuk mengumpulkan kebutuhan-kebutuhan sistem dalam membangun SIPKIA. Tahap-tahap dalam proses ini adalah elisitasi kebutuhan, spesifikasi kebutuhan, validasi verifikasi dan manajemen kebutuhan.

4.1.1 Elisitasi Kebutuhan

Tahap elisitasi adalah kegiatan dimana developer perangkat lunak bekerja dengan *customers* dan pengguna akhir dari sistem untuk mengetahui tentang domain aplikasi, layanan yang harus disediakan sistem, kinerja yang diperlukan sistem, kendala hardware, dan sebagainya (Sommerville, 2011). Pada tahap elisitasi kebutuhan sistem ini terdiri dari identifikasi stakeholder, teknik elisitasi, dan proses bisnis sistem.

4.1.1.1 Identifikasi Stakeholder

Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method* digunakan sebagai media untuk mempermudah proses registrasi informasi kesehatan ibu dan anak oleh bidan desa dan puskesmas Kecamatan Tumpang. Identifikasi *stakeholder* menjabarkan siapa saja yang terlibat dalam proses registrasi tersebut. Identifikasi stakeholder dapat dilihat lebih lengkap pada Tabel 4.1 berikut,

Tabel 4.1 Identifikasi stakeholder

Stakeholder	Deskripsi
Ibu hamil dan bayi	Ibu hamil dan bayinya merupakan subjek utama dalam administrasi kohort bidan desa
Bidan Desa	Bidan Desa merupakan pihak yang sangat berperan penting dalam SIPKIA. Bidan desa adalah aktor utama yang akan memasukkan data-data kesehatan ibu dan anak.
Kader	Kader posyandu membantu bidan desa dalam menulis administrasi dalam buku bantu register kohort
Pegawai Puskesmas (BKIA Puskesmas)	Pegawai puskesmas bertanggung jawab atas pengecekan data yang telah masuk dalam sistem.
Kepala UPTD Puskesmas	Kepala UPTD Puskesmas bertanggung jawab atas persetujuan proses registrasi secara online

Sesuai dari Tabel 4.1 stakeholder dari sistem administrasi kohort dan pemetaan melibatkan ibu hamil dan anak termasuk bayi balita dan apras, bidan desa, kader posyandu, pegawai puskesmas dan kepala UPTD puskesmas selaku pemberi kebijakan penggunaan sistem.

4.1.1.2 Teknik Elisitasi

Pada penelitian ini elisitasi dilakukan dengan metode wawancara dan pengamatan langsung (observasi). Penggunaan metode wawancara karena langsung berhubungan dengan pemangku kepentingan sistem yang merupakan bagian dari sebagian besar proses rekayasa persyaratan. Dalam wawancara tersebut, tim rekayasa persyaratan menempatkan pertanyaan kepada para pemangku kepentingan dalam hal ini pegawai puskesmas dan bidan desa tentang sistem yang mereka gunakan saat dan sistem yang akan dikembangkan (Sommerville, 2011). Wawancara dilakukan kepada pengguna utama yaitu bidan desa serta pegawai Puskesmas Kecamatan Tumpang bagian KIA. Selain itu identifikasi masalah juga dilakukan dengan pengamatan langsung yang dilakukan ketika ada kegiatan posyandu dimana proses registrasi kohort dilakukan dan proses perekapan oleh bidan desa. Dari kedua metode tersebut maka dapat ditemukan kebutuhan pada Tabel 4.2 berikut :

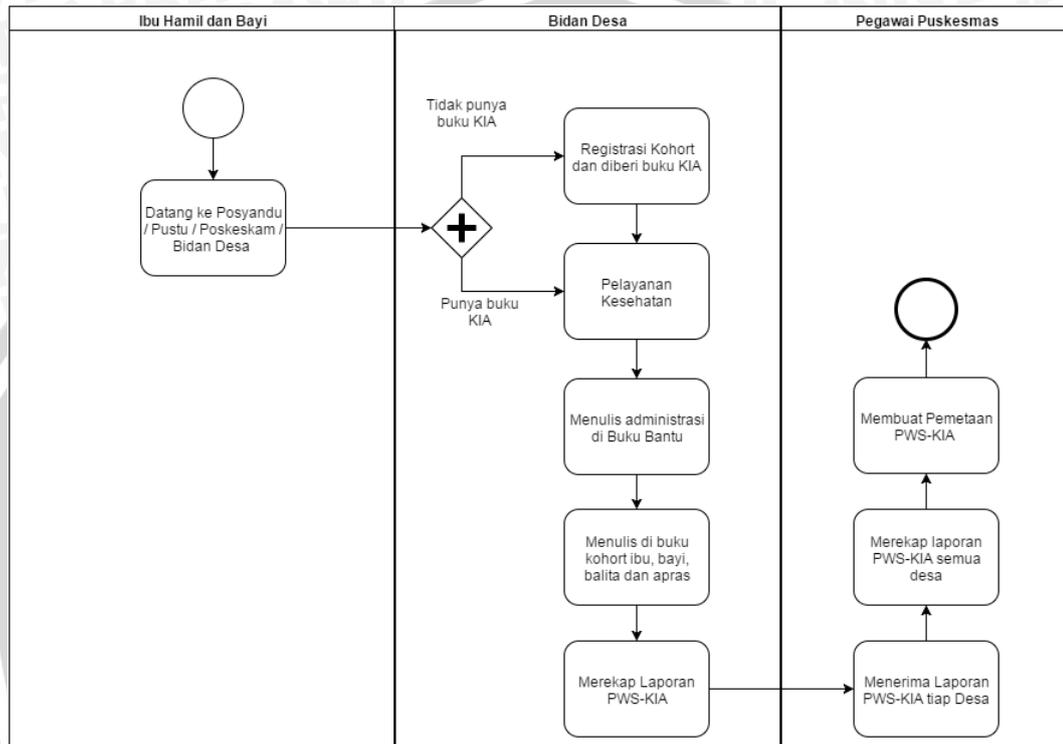
Tabel 4.2 Requirement discovery

No	Temuan Kebutuhan (Requirement Discovery)
1	Proses registrasi kohort bidan desa dapat disimpan dalam sistem
2	Bidan desa dapat mengelola register kohort ibu, bayi, balita dan apras pada sistem
3	Bidan desa dapat menambahkan pelayanan pada register kohort yang ada pada sistem
4	Sistem menyediakan fitur search pada register kohort untuk memudahkan pencarian data register
6	Bidan desa dapat melihat rekapan data KIA masing-masing bulan
7	Pegawai puskesmas dapat melihat rekapan bulanan KIA kolektif dari semua desa
8	Bidan desa dapat melihat pemetaan data KIA masing-masing bulan
9	Pegawai puskesmas dapat melihat pemetaan bulanan KIA kolektif dari semua desa
10	Pegawai puskesmas dapat mengelola user dari sistem
11	Sistem memiliki keamanan yang hanya diakses oleh user yang teridentifikasi
12	Sistem memiliki tampilan <i>user friendly</i>
13	Sistem mudah digunakan

Berdasarkan temuan kebutuhan pada Tabel 4.2 tersebut masih perlu dispesifikasikan lagi seperti temuan kebutuhan bahwa bidan desa dapat mengelola register kohort ibu, bayi, balita dan apras pada sistem akan dispesifikasikan menjadi lihat register, update register, tambah register, hapus register dan lain-lainnya yang akan dibahas lebih lengkap pada sub bab spesifikasi kebutuhan.

4.1.1.3 Analisa Proses Bisnis

Proses registrasi informasi kesehatan ibu dan anak di bidan desa dilakukan secara manual dengan memasukkan data pada buku register kohort, macam dari register kohort sendiri terdiri dari register ibu, bayi, balita dan apras. Selain itu proses perekapan dan pemetaan juga dilakukan secara manual. Proses pendataan registrasi dan pemetaan tersebut dapat dilihat pada tahapan-tahapan yang dapat digambarkan dengan gambar 4.2 berikut,



Gambar 4.2 Proses bisnis sistem informasi dan pemetaan

Proses bisnis registrasi kohort pada Gambar 4.2 dimulai dengan datangnya ibu hamil atau bayi (anak) menuju tempat pelayanan kesehatan seperti posyandu, pustu, puskesmas, bidan desa untuk mendapatkan pelayanan kesehatan. Kemudian ibu atau bayi dilihat apakah mempunyai buku KIA atau tidak. Ketika ibu atau bayi tidak memiliki buku KIA maka akan ditambahkan kedalam buku register kohort dan diberi buku KIA. Setelah mempunyai buku KIA ibu atau bayi akan diberikan pelayanan kesehatan oleh tenaga kesehatan sesuai dengan umur bayi. Setelah proses pelayanan bidan menulis dibuku bantu karena proses penulisan di buku kohort susah untuk dilakukan pada saat pelayanan mengingat proses pelayanan khususnya pada saat di posyandu selalu terdapat banyak antrian. Ketika kegiatan pemeriksaan selesai maka kegiatan bidan selanjutnya adalah menyalin data dari buku bantu ke buku register kohort secara hati-hati. Kemudian merekap laporan dari keempat register kohort yang akan dilaporkan kepada pegawai puskesmas bagian KIA. Ketika pegawai puskesmas telah menerima semua rekapan dari semua desa langkah selanjutnya pegawai puskesmas akan melakukan rekapan sekecamatan dan melakukan pemetaan kesehatan ibu dan anak secara manual.

4.1.2 Spesifikasi Kebutuhan

Spesifikasi kebutuhan adalah proses menuliskan pengguna dan kebutuhan sistem dalam dokumen kebutuhan. Idealnya, pengguna dan persyaratan sistem harus jelas, tidak ambigu, mudah dimengerti, lengkap, dan konsisten (Sommerville,2011).Pada tahap spesifikasi kebutuhan seluruh informasi dari hasil elisitasi akan disempurnakan secara lebih detail dan tepat yang nantinya akan menjadi landasan bagi perancangan dan implementasi. Daftar kebutuhan fungsional Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method* dijelaskan pada Tabel 4.3 berikut,

Tabel 4.3 Daftar kebutuhan fungsional

Kode	Kebutuhan Fungsional	Deskripsi	Prioritas
FR-01	Melihat data register kohort ibu	Sistem dapat menampilkan data register kohort ibu.	Tinggi
FR-02	Menambah data register kohort Ibu	Sistem dapat menambahkan data register kohort ibu.	Tinggi
FR-03	Megubah data register kohort ibu	Sistem dapat mengubah data register kohort ibu.	Sedang
FR-04	Menghapus data register kohort ibu	Sistem dapat menghapus data register kohort ibu.	Rendah
FR-05	Melihat detail pelayanan kohort ibu	Sistem dapat menampilkan pelayanan pada salah satu register ibu	Tinggi
FR-06	Menambah pelayanan kohort ibu	Sistem dapat menambahkan pelayanan pada salah satu register ibu	Tinggi
FR-07	Mengubah pelayanan kohort ibu	Sistem dapat mengubah pelayanan pada salah satu register ibu	Sedang
FR-08	Menghapus pelayanan kohort ibu	Sistem dapat menghapus pelayanan pada salah satu register ibu	Rendah
FR-09	Melihat data register kohort bayi	Sistem dapat menampilkan data register kohort bayi.	Tinggi
FR-10	Menambah data register kohort Bayi	Sistem dapat menambahkan data register kohort bayi.	Tinggi
FR-11	Megubah data register kohort bayi	Sistem dapat mengubah data register kohort bayi.	Sedang
FR-12	Menghapus data register kohort bayi	Sistem dapat menghapus data register kohort bayi.	Rendah

Tabel 4.3 Daftar kebutuhan fungsional (lanjutan)

Kode	Kebutuhan Fungsional	Deskripsi	Prioritas
FR-13	Melihat detail pelayanan kohort bayi	Sistem dapat menampilkan pelayanan pada salah satu register bayi	Tinggi
FR-14	Menambah pelayanan kohort bayi	Sistem dapat menambahkan pelayanan pada salah satu register bayi	Tinggi
FR-15	Mengubah pelayanan kohort bayi	Sistem dapat mengubah pelayanan pada salah satu register bayi	Sedang
FR-16	Menghapus pelayanan kohort bayi	Sistem dapat menghapus pelayanan pada salah satu register bayi	Rendah
FR-17	Melihat data register kohort balita	Sistem dapat menampilkan data register kohort balita.	Tinggi
FR-18	Menambah data register kohort Balita	Sistem dapat menambahkan data register kohort balita.	Tinggi
FR-19	Megubah data register kohort balita	Sistem dapat mengubah data register kohort balita.	Sedang
FR-20	Menghapus data register kohort balita	Sistem dapat menghapus data register kohort balita.	Rendah
FR-21	Melihat detail pelayanan kohort balita	Sistem dapat menampilkan pelayanan pada salah satu register balita	Tinggi
FR-22	Menambah pelayanan kohort balita	Sistem dapat menambahkan pelayanan pada salah satu register balita	Tinggi
FR-23	Mengubah pelayanan kohort balita	Sistem dapat mengubah pelayanan pada salah satu register balita	Sedang
FR-24	Menghapus pelayanan kohort balita	Sistem dapat menghapus pelayanan pada salah satu register balita	Rendah
FR-25	Melihat data register kohort apras	Sistem dapat menampilkan data register kohort apras.	Tinggi
FR-26	Menambah data register kohort Apras	Sistem dapat menambahkan data register kohort apras.	Tinggi
FR-27	Megubah data register kohort apras	Sistem dapat mengubah data register kohort apras.	Sedang
FR-28	Menghapus data register kohort apras	Sistem dapat menghapus data register kohort apras.	Rendah

Tabel 4.3 Daftar kebutuhan fungsional (lanjutan)

Kode	Kebutuhan Fungsional	Deskripsi	Prioritas
FR-29	Melihat detail pelayanan kohort apras	Sistem dapat menampilkan pelayanan pada salah satu register apras	Tinggi
FR-30	Menambah pelayanan kohort apras	Sistem dapat menambahkan pelayanan pada salah satu register apras	Tinggi
FR-31	Mengubah pelayanan kohort apras	Sistem dapat mengubah pelayanan pada salah satu register apras	Sedang
FR-32	Menghapus pelayanan kohort apras	Sistem dapat menghapus pelayanan pada salah satu register apras	Rendah
FR-33	Melihat rekapan PWS-KIA desa	Sistem dapat menampilkan rekapan PWS-KIA pada tiap desa	Tinggi
FR-34	Mencetak rekapan PWS-KIA desa	Sistem dapat mencetak dalam bentuk dokumen PWS-KIA pada tiap desa	Tinggi
FR-35	Melihat rekapan PWS-KIA kecamatan	Sistem dapat menampilkan rekapan PWS-KIA tingkat kecamatan	Tinggi
FR-36	Mencetak rekapan PWS-KIA kecamatan	Sistem dapat mencetak dalam bentuk dokumen PWS-KIA kecamatan	Tinggi
FR-37	Melihat pemetaan PWS-KIA desa	Sistem dapat menampilkan pemetaan PWS-KIA pada tiap desa	Tinggi
FR-38	Mencetak pemetaan PWS-KIA desa	Sistem dapat mencetak dalam bentuk dokumen pemetaan PWS-KIA desa	Tinggi
FR-39	Melihat pemetaan PWS-KIA kecamatan	Sistem dapat menampilkan pemetaan PWS-KIA komulatif sekecamatan	Tinggi
FR-40	Mencetak pemetaan PWS-KIA kecamatan	Sistem dapat mencetak dalam bentuk dokumen pemetaan PWS-KIA komulatif sekecamatan	Tinggi
FR-41	Memperbarui rekapan PWS-KIA	Sistem dapat memperbarui data rekapan dari semua register kohort	Tinggi
FR-42	Memperbarui pemetaan PWS-KIA	Sistem dapat memperbarui data pemetaan dari semua register kohort	Tinggi
FR-43	Melihat user sistem	Sistem dapat menampilkan user sistem	Rendah
FR-44	Menambah user sistem	Sistem dapat menambah user sistem	Rendah
FR-45	Mengedit user sistem	Sistem dapat mengubah user sistem	Rendah

Tabel 4.3 Daftar kebutuhan fungsional (lanjutan)

Kode	Kebutuhan Fungsional	Deskripsi	Prioritas
FR-46	Menghapus user sistem	Sistem dapat menghapus user sistem	Rendah
FR-47	Login	User dapat login ke sistem	Rendah
FR-48	Logout	User dapat logout ke sistem	Rendah

Spesifikasi kebutuhan fungsional pada Tabel 4.3 menjelaskan kebutuhan fungsional apa saja yang dibutuhkan sistem, deskripsi kebutuhan sistem serta prioritas kebutuhan sistem. Sebagai contoh melihat rekapan PWS-KIA desa merupakan kebutuhan dengan prioritas tinggi karena kebutuhan tersebut sangat penting yaitu sistem harus dapat menampilkan rekapan PWS-KIA pada tiap desa. Sedangkan kegiatan mengelola user seperti melihat user, menambah user, mengedit user, dan menghapus user memiliki prioritas rendah karena kegiatan tersebut hanya sebagai pelengkap dari fitur-fitur utama sistem.

4.1.3 Validasi dan Verifikasi

Dari spesifikasi kebutuhan yang telah didefinisikan perlu adanya validasi dan verifikasi. Validasi kebutuhan adalah proses memeriksa bahwa kebutuhan-kebutuhan sistem benar-benar mendefinisikan sistem yang benar-benar diinginkan oleh client (Sommerville, 2011). Pada proses ini memastikan kebutuhan - kebutuhan yang telah disepakati sudah ada pada sistem seperti fitur-fitur yang diperlukan. Proses validasi dan verifikasi pada penelitian ini dilakukan oleh badan desa dan pegawai kecamatan bagian KIA.

Tabel validasi dan verifikasi dari kebutuhan fungsional sistem yang telah dilakukan dapat dilihat pada Tabel 4.4 berikut,

Tabel 4.4 Validasi dan verifikasi kebutuhan fungsional

Kode	Kebutuhan Fungsional	Validasi	Verifikasi
FR-01	Melihat data register kohort ibu	√	√
FR-02	Menambah data register kohort Ibu	√	√
FR-03	Mengubah data register kohort ibu	√	√
FR-04	Menghapus data register kohort ibu	√	√
FR-05	Melihat detail pelayanan kohort ibu	√	√
FR-06	Menambah pelayanan kohort ibu	√	√
FR-07	Mengubah pelayanan kohort ibu	√	√
FR-08	Menghapus pelayanan kohort ibu	√	√

Tabel 4.4 Validasi dan verifikasi kebutuhan fungsional (lanjutan)

Kode	Kebutuhan Fungsional	Validasi	Verifikasi
FR-09	Melihat data register kohort bayi	√	√
FR-10	Menambah data register kohort Bayi	√	√
FR-11	Megubah data register kohort bayi	√	√
FR-12	Menghapus data register kohort bayi	√	√
FR-13	Melihat detail pelayanan kohort bayi	√	√
FR-14	Menambah pelayanan kohort bayi	√	√
FR-15	Mengubah pelayanan kohort bayi	√	√
FR-16	Menghapus pelayanan kohort bayi	√	√
FR-17	Melihat data register kohort balita	√	√
FR-18	Menambah data register kohort Balita	√	√
FR-19	Megubah data register kohort balita	√	√
FR-20	Menghapus data register kohort balita	√	√
FR-21	Melihat detail pelayanan kohort balita	√	√
FR-22	Menambah pelayanan kohort balita	√	√
FR-23	Mengubah pelayanan kohort balita	√	√
FR-24	Menghapus pelayanan kohort balita	√	√
FR-25	Melihat data register kohort apras	√	√
FR-26	Menambah data register kohort Apras	√	√
FR-27	Megubah data register kohort apras	√	√
FR-28	Menghapus data register kohort apras	√	√
FR-29	Melihat detail pelayanan kohort apras	√	√
FR-30	Menambah pelayanan kohort apras	√	√
FR-31	Mengubah pelayanan kohort apras	√	√
FR-32	Menghapus pelayanan kohort apras	√	√
FR-33	Melihat rekapan PWS-KIA desa	√	√
FR-34	Mencetak rekapan PWS-KIA desa	√	√
FR-35	Melihat rekapan PWS-KIA kecamatan	√	√
FR-36	Mencetak rekapan PWS-KIA kecamatan	√	√

Tabel 4.4 Validasi dan verifikasi kebutuhan fungsional (lanjutan)

Kode	Kebutuhan Fungsional	Validasi	Verifikasi
FR-37	Melihat pemetaan PWS-KIA desa	√	√
FR-38	Mencetak pemetaan PWS-KIA desa	√	√
FR-39	Melihat pemetaan PWS-KIA kecamatan	√	√
FR-40	Mencetak pemetaan PWS-KIA kecamatan	√	√
FR-41	Memperbarui rekapan PWS-KIA	√	√
FR-42	Memperbarui rekapan PWS-KIA	√	√
FR-43	Melihat user sistem	√	√
FR-44	Menambah user sistem	√	√
FR-45	Mengedit user sistem	√	√
FR-46	Menghapus user sistem	√	√
FR-47	Login	√	√
FR-48	Logout	√	√

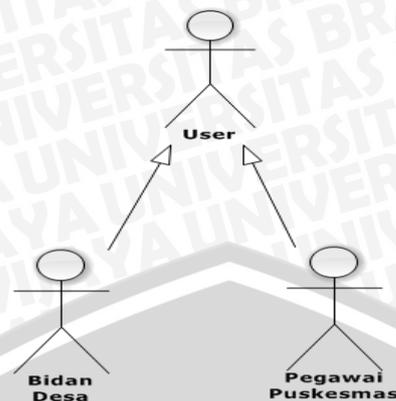
Dari hasil validasi dan verifikasi yang telah tertera pada Tabel 4.5 tersebut dari semua kebutuhan fungsional telah tervalidasi dan terverifikasi semua dengan nilai valid.

4.1.4 Manajemen Kebutuhan

Kebutuhan sistem perangkat lunak besar selalu berubah. Salah satu alasan untuk ini adalah bahwa sistem biasanya dikembangkan untuk mengatasi masalah-masalah yang tidak bisa sepenuhnya didefinisikan. Karena masalah tidak dapat didefinisikan secara penuh, persyaratan software tidak lengkap (Sommerville, 2011). Manajemen kebutuhan adalah tahap lanjutan yang dikerjakan setelah sebelumnya menyelesaikan proses validasi dan verifikasi. Fungsi utama dilakukannya manajemen kebutuhan ini adalah untuk mengidentifikasi, mengontrol, mencari kebutuhan dan perubahan kebutuhan yang diperlukan seiring berjalannya proses pengerjaan. Dalam manajemen kebutuhan SIPKIA terdiri dari identifikasi aktor dan identifikasi kebutuhan.

4.1.4.1 Identifikasi Aktor

Identifikasi aktor bertujuan untuk menggambarkan pengguna yang akan terlibat dalam penggunaan sistem dan membantu memberikan gambaran yang jelas tentang apa yang harus dikerjakan sistem. Identifikasi aktor yang akan berperan dalam sistem dapat dilihat pada Gambar 4.3 berikut,



Gambar 4.3 Identifikasi aktor

Penjelasan Gambar 4.3 mengenai deskripsi dari aktor dapat dilihat pada Tabel 4.5 berikut,

Tabel 4.5 Identifikasi user

Aktor	Spesifikasi	Deskripsi
User	Bidan Desa	Bidan Desa merupakan pihak yang sangat berperan penting dalam sistem. Bidan desa adalah aktor utama yang akan memasukkan data-data kesehatan ibu dan anak.
	Pegawai Puskesmas	Pegawai puskesmas merupakan pihak yang akan menerima laporan dari hasil registrasi di posyandu atau bidan desa masing-masing. Selain itu staff puskesmas dapat melihat hasil pemetaan PWS-KIA dan mencetak Pemetaan tersebut.

4.1.4.2 Identifikasi Kebutuhan

Identifikasi kebutuhan merupakan tahap dimana kebutuhan yang telah didefinisikan harus dapat diidentifikasi dengan mudah. Identifikasi tersebut yaitu terdiri dari penyetaraan antara kebutuhan sistem dan kebutuhan interface. Identifikasi kebutuhan SIPKIA dapat dilihat pada Tabel 4.6 berikut,

Tabel 4.6 Identifikasi kebutuhan

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi	Kode Interface
FR-01	Melihat data register kohort ibu	Sistem dapat menampilkan data register kohort ibu.	IR-01
FR-02	Menambah data register kohort Ibu	Sistem dapat menambahkan data register kohort ibu.	IR-02



Tabel 4.6 Identifikasi kebutuhan (lanjutan)

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi	Kode Interface
FR-03	Mengubah data register kohort ibu	Sistem dapat mengubah data register kohort ibu.	IR-03
FR-04	Menghapus data register kohort ibu	Sistem dapat menghapus data register kohort ibu.	IR-04
FR-05	Melihat detail pelayanan kohort ibu	Sistem dapat menampilkan pelayanan pada salah satu register ibu	IR-05
FR-06	Menambah pelayanan kohort ibu	Sistem dapat menambahkan pelayanan pada salah satu register ibu	IR-06
FR-07	Mengubah pelayanan kohort ibu	Sistem dapat mengubah pelayanan pada salah satu register ibu	IR-07
FR-08	Menghapus pelayanan kohort ibu	Sistem dapat menghapus pelayanan pada salah satu register ibu	IR-08
FR-09	Melihat data register kohort bayi	Sistem dapat menampilkan data register kohort bayi.	IR-09
FR-10	Menambah data register kohort Bayi	Sistem dapat menambahkan data register kohort bayi.	IR-10
FR-11	Mengubah data register kohort bayi	Sistem dapat mengubah data register kohort bayi.	IR-11
FR-12	Menghapus data register kohort bayi	Sistem dapat menghapus data register kohort bayi.	IR-12
FR-13	Melihat detail pelayanan kohort bayi	Sistem dapat menampilkan pelayanan pada salah satu register bayi	IR-13
FR-14	Menambah pelayanan kohort bayi	Sistem dapat menambahkan pelayanan pada salah satu register bayi	IR-14
FR-15	Mengubah pelayanan kohort bayi	Sistem dapat mengubah pelayanan pada salah satu register bayi	IR-15
FR-16	Menghapus pelayanan kohort bayi	Sistem dapat menghapus pelayanan pada salah satu register bayi	IR-16
FR-17	Melihat data register kohort balita	Sistem dapat menampilkan data register kohort balita.	IR-17
FR-18	Menambah data register kohort Balita	Sistem dapat menambahkan data register kohort balita.	IR-18

Tabel 4.6 Identifikasi kebutuhan (lanjutan)

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi	Kode Interface
FR-19	Mengubah data register kohort balita	Sistem dapat mengubah data register kohort balita.	IR-19
FR-20	Menghapus data register kohort balita	Sistem dapat menghapus data register kohort balita.	IR-20
FR-21	Melihat detail pelayanan kohort balita	Sistem dapat menampilkan pelayanan pada salah satu register balita	IR-21
FR-22	Menambah pelayanan kohort balita	Sistem dapat menambahkan pelayanan pada salah satu register balita	IR-22
FR-23	Mengubah pelayanan kohort balita	Sistem dapat mengubah pelayanan pada salah satu register balita	IR-23
FR-24	Menghapus pelayanan kohort balita	Sistem dapat menghapus pelayanan pada salah satu register balita	IR-24
FR-25	Melihat data register kohort apras	Sistem dapat menampilkan data register kohort apras.	IR-25
FR-26	Menambah data register kohort Apras	Sistem dapat menambahkan data register kohort apras.	IR-26
FR-27	Mengubah data register kohort apras	Sistem dapat mengubah data register kohort apras.	IR-27
FR-28	Menghapus data register kohort apras	Sistem dapat menghapus data register kohort apras.	IR-28
FR-29	Melihat detail pelayanan kohort apras	Sistem dapat menampilkan pelayanan pada salah satu register apras	IR-29
FR-30	Menambah pelayanan kohort apras	Sistem dapat menambahkan pelayanan pada salah satu register apras	IR-30
FR-31	Mengubah pelayanan kohort apras	Sistem dapat mengubah pelayanan pada salah satu register apras	IR-31
FR-32	Menghapus pelayanan kohort apras	Sistem dapat menghapus pelayanan pada salah satu register apras	IR-32
FR-33	Melihat rekapan PWS-KIA desa	Sistem dapat menampilkan rekapan PWS-KIA pada tiap desa	IR-33

Tabel 4.6 Identifikasi kebutuhan (lanjutan)

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi	Kode Interface
FR-34	Mencetak rekapan PWS-KIA desa	Sistem dapat mencetak dalam bentuk dokumen PWS-KIA pada tiap desa	IR-34
FR-35	Melihat rekapan PWS-KIA kecamatan	Sistem dapat menampilkan rekapan PWS-KIA tingkat kecamatan	IR-35
FR-36	Mencetak rekapan PWS-KIA kecamatan	Sistem dapat mencetak dalam bentuk dokumen PWS-KIA kecamatan	IR-36
FR-37	Melihat pemetaan PWS-KIA desa	Sistem dapat menampilkan pemetaan PWS-KIA pada tiap desa	IR-37
FR-38	Mencetak pemetaan PWS-KIA desa	Sistem dapat mencetak dalam bentuk dokumen pemetaan PWS-KIA desa	IR-38
FR-39	Melihat pemetaan PWS-KIA kecamatan	Sistem dapat menampilkan pemetaan PWS-KIA kumulatif sekecamatan	IR-39
FR-40	Mencetak pemetaan PWS-KIA kecamatan	Sistem dapat mencetak dalam bentuk dokumen pemetaan PWS-KIA kumulatif sekecamatan	IR-40
FR-41	Memperbarui rekapan PWS-KIA	Sistem dapat memperbarui data rekapan dari semua register kohort	IR-41
FR-42	Memperbarui pemetaan PWS-KIA	Sistem dapat memperbarui data pemetaan dari semua register kohort	IR-42
FR-43	Melihat user sistem	Sistem dapat menampilkan user sistem	IR-43
FR-44	Menambah user sistem	Sistem dapat menambah user sistem	IR-44
FR-45	Mengedit user sistem	Sistem dapat mengubah user sistem	IR-45
FR-46	Menghapus user sistem	Sistem dapat menghapus user sistem	IR-46
FR-47	Login	User dapat login ke sistem	IR-47
FR-48	Logout	User dapat logout ke sistem	IR-48

Identifikasi kebutuhan pada Tabel 4.6 memetakan kebutuhan sistem dengan kebutuhan antarmuka. Pemetaan ini akan mempermudah jika adanya kemungkinan perubahan kebutuhan sehingga dapat diidentifikasi dari kebutuhan antarmukanya.

4.2 Model Kebutuhan

Setelah proses rekayasa kebutuhan maka perlu dimodelkan untuk memudahkan pembacaan kebutuhan dalam tahap perancangan. Proses pemodelan kebutuhan ini menggunakan *Unified Modelling Language (UML)* yang terdiri dari *usecase diagram*, *sequence diagram*, dan *class diagram*.

4.2.1 Pemodelan Usecase Diagram

Dalam konteks UML, tahap konseptualisasi dilakukan dengan pembuatan *use case diagram* yang merupakan deskripsi bagaimana perangkat lunak akan digunakan oleh penggunanya. Pada bagian ini akan diuraikan bagaimana menggambarkan *use case diagram* pada Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method*,

4.2.1.1 Diagram Use Case Sistem

Diagram *use case* menggambarkan ruang lingkup sistem yang sedang dibangun. Diagram *use case* terdiri dari dua komponen utama, yakni *use case* dan aktor. *Use case* meliputi semua hal yang ada pada sistem, sedangkan aktor meliputi semua hal diluar sistem. Aktor termasuk seseorang atau apa saja yang berhubungan dengan sistem yang dibangun. *Use case* dibuat berdasarkan kebutuhan fungsional sistem. Pemetaan kebutuhan fungsional dan *use case* dapat dilihat pada Tabel 4.7 berikut ini,

Tabel 4.7 Pemetaan kebutuhan fungsional dengan *use case*

Kode Kebutuhan	Kebutuhan	Kode Use Case	Nama Use Case	Aktor
FR-01	Melihat data register kohort ibu	UC-01	Lihat kohort ibu	Bidan desa
FR-02	Menambah data register kohort ibu	UC-02	Tambah kohort ibu	Bidan desa
FR-03	Mengubah data register kohort ibu	UC-03	Ubah kohort ibu	Bidan desa
FR-04	Menghapus data register kohort ibu	UC-04	Hapus kohort ibu	Bidan desa
FR-05	Melihat detail pelayanan kohort ibu	UC-05	Lihat pelayanan ibu	Bidan desa
FR-06	Menambah pelayanan kohort ibu	UC-06	Tambah pelayanan ibu	Bidan desa
FR-07	Mengubah pelayanan kohort ibu	UC-07	Ubah pelayanan ibu	Bidan desa
FR-08	Menghapus pelayanan kohort ibu	UC-08	Hapus pelayanan ibu	Bidan desa

Tabel 4.7 Pemetaan kebutuhan fungsional dengan *use case* (lanjutan)

Kode Kebutuhan	Kebutuhan	Kode Use Case	Nama Use Case	Aktor
FR-09	Melihat data register kohort bayi	UC-09	Lihat kohort bayi	Bidan desa
FR-10	Menambah data register kohort Bayi	UC-10	Tambah kohort bayi	Bidan desa
FR-11	Mengubah data register kohort bayi	UC-11	Ubah kohort bayi	Bidan desa
FR-12	Menghapus data register kohort bayi	UC-12	Hapus kohort bayi	Bidan desa
FR-13	Melihat detail pelayanan kohort bayi	UC-13	Lihat pelayanan bayi	Bidan desa
FR-14	Menambah pelayanan kohort bayi	UC-14	Tambah pelayanan bayi	Bidan desa
FR-15	Mengubah pelayanan kohort bayi	UC-15	Ubah pelayanan bayi	Bidan desa
FR-16	Menghapus pelayanan kohort bayi	UC-16	Hapus pelayanan bayi	Bidan desa
FR-17	Melihat data register kohort balita	UC-17	Lihat kohort balita	Bidan desa
FR-18	Menambah data register kohort Balita	UC-18	Tambah kohort balita	Bidan desa
FR-19	Mengubah data register kohort balita	UC-19	Ubah kohort balita	Bidan desa
FR-20	Menghapus data register kohort balita	UC-20	Hapus kohort balita	Bidan desa
FR-21	Melihat detail pelayanan kohort balita	UC-21	Lihat pelayanan balita	Bidan desa
FR-22	Menambah pelayanan kohort balita	UC-22	Tambah pelayanan balita	Bidan desa
FR-23	Mengubah pelayanan kohort balita	UC-23	Ubah pelayanan balita	Bidan desa
FR-24	Menghapus pelayanan kohort balita	UC-24	Hapus pelayanan balita	Bidan desa
FR-25	Melihat data register kohort apras	UC-25	Lihat kohort apras	Bidan desa

Tabel 4.7 Pemetaan kebutuhan fungsional dengan *use case* (lanjutan)

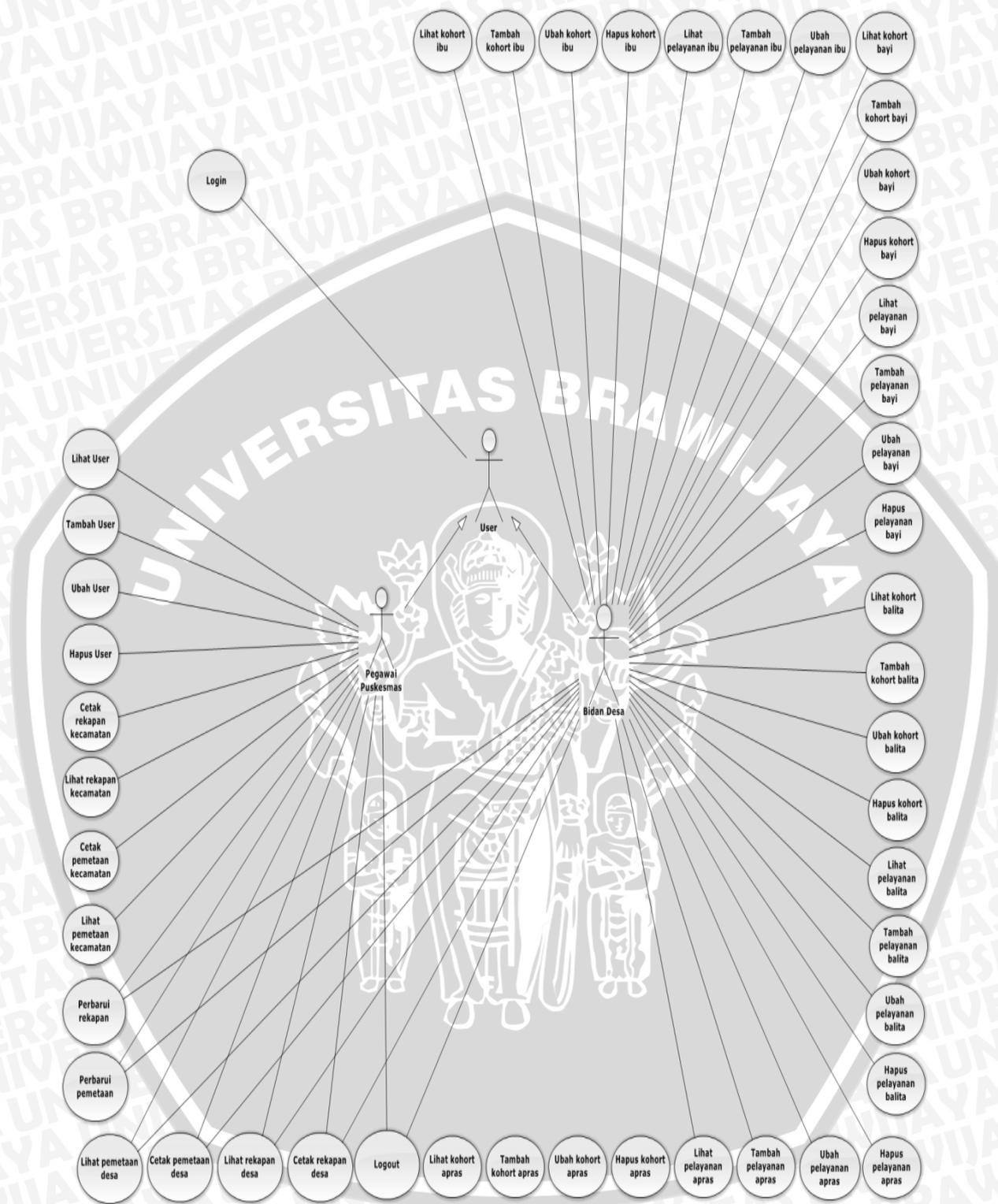
Kode Kebutuhan	Kebutuhan	Kode Use Case	Nama Use Case	Aktor
FR-26	Menambah data register kohort Apras	UC-26	Tambah kohort apras	Bidan desa
FR-27	Mengubah data register kohort apras	UC-27	Ubah kohort apras	Bidan desa
FR-28	Menghapus data register kohort apras	UC-28	Hapus kohort apras	Bidan desa
FR-29	Melihat detail pelayanan kohort apras	UC-29	Lihat pelayanan apras	Bidan desa
FR-30	Menambah pelayanan kohort apras	UC-30	Tambah pelayanan apras	Bidan desa
FR-31	Mengubah pelayanan kohort apras	UC-31	Ubah pelayanan apras	Bidan desa
FR-32	Menghapus pelayanan kohort apras	UC-32	Hapus pelayanan apras	Bidan desa
FR-33	Melihat rekapan PWS-KIA desa	UC-33	Lihat rekapan desa	Bidan desa dan pegawai puskesmas
FR-34	Mencetak rekapan PWS-KIA desa	UC-34	Cetak rekapan desa	Bidan desa dan pegawai puskesmas
FR-35	Melihat rekapan PWS-KIA kecamatan	UC-35	Lihat rekapan kecamatan	Pegawai puskesmas
FR-36	Mencetak rekapan PWS-KIA kecamatan	UC-36	Cetak rekapan kecamatan	Pegawai puskesmas
FR-37	Melihat pemetaan PWS-KIA desa	UC-37	Lihat pemetaan desa	Bidan desa dan pegawai puskesmas
FR-38	Mencetak pemetaan PWS-KIA desa	UC-38	Cetak pemetaan desa	Bidan desa dan pegawai puskesmas
FR-39	Melihat pemetaan PWS-KIA kecamatan	UC-39	Lihat pemetaan kecamatan	Pegawai puskesmas
FR-40	Mencetak pemetaan PWS-KIA kecamatan	UC-40	Cetak pemetaan kecamatan	Pegawai puskesmas
FR-41	Memperbarui rekapan PWS-KIA	UC-41	Perbarui rekapan	Bidan desa dan pegawai puskesmas
FR-42	Memperbarui pemetaan PWS-KIA	UC-42	Perbarui pemetaan	Bidan desa dan pegawai puskesmas

Tabel 4.7 Pemetaan kebutuhan fungsional dengan *use case* (lanjutan)

Kode Kebutuhan	Kebutuhan	Kode Use Case	Nama Use Case	Aktor
FR-43	Melihat user sistem	UC-43	Lihat user	Pegawai puskesmas
FR-44	Menambah user sistem	UC-44	Tambah user	Pegawai puskesmas
FR-45	Mengedit user sistem	UC-45	Ubah user	Pegawai puskesmas
FR-46	Menghapus user sistem	UC-46	Hapus user	Pegawai puskesmas
FR-47	Login	UC-47	Login	User
FR-48	Logout	UC-48	Logout	Bidan desa dan pegawai puskesmas

Dari pemetaan kebutuhan fungsional dan *use case* pada Tabel 4.7 tersebut akan teridentifikasi aktor dalam kebutuhan fungsional yang telah didefinisikan diawal. SIPKIA memiliki beberapa kegiatan utama. Penggambaran kegiatan tersebut dapat dilihat dalam Gambar 4.4 diagram *use case* berikut,





Gambar 4.4 Diagram use case sistem

4.2.1.2 Skenario Use Case

Penjelasan beberapa *use case diagram* pada Gambar 4.4 dapat dilihat pada tabel spesifikasi *use case* berikut,

1. [UC-01] Lihat kohort ibu

Tabel 4.8 Skenario *use case* lihat kohort ibu

Skenario Kasus pada Sistem	
Nomor FR	FR-01
Nama	Lihat kohort ibu
Tujuan	Bidan desa dapat melihat data-data yang masuk dalam register kohort ibu
Deskripsi	Sistem dapat menampilkan data register kohort ibu.
Aktor	Bidan Desa
Skenario Utama	
Kondisi Awal	Aktor telah login ke sistem
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol menu kohort ibu	2. Sistem menampilkan tabel register kohort ibu
Skenario Alternatif 1: Jika sesi login tidak valid	
	2. Sistem menampilkan halaman login
Skenario Alternatif 2: Jika sesi desa tidak valid	
	2. Sistem memberikan alert bahwa user tidak mempunyai hak akses
Kondisi Akhir	Sistem menampilkan data-data register kohort ibu yang tersimpan di database

Pada Tabel 4.8 menggambarkan skenario dari *use case* lihat register kohort ibu. Tujuan dari kebutuhan ini adalah bidan desa dapat melihat data-data yang masuk dalam register kohort ibu. Skenario diawali dengan aktor login ke dalam sistem. Kemudian aktor menekan tombol menu kohort ibu dan sistem akan menampilkan tabel register kohort ibu. Ada dua skenario alternatif yaitu terjadi jika sesi login user tidak valid maka sistem akan menampilkan halaman login sedangkan alternatif kedua yaitu ketika sesi desa tidak valid yakni user tidak memiliki hak pada desa yang dipilih maka sistem akan memberikan alert bahwa user tidak mempunyai hak akses pada register yang dipilih. Kondisi akhir dari skenario ini adalah sistem menampilkan data-data register kohort ibu yang tersimpan di database.

2. [UC-02] Tambah kohort ibu

Tabel 4.9 Skenario *use case* tambah kohort ibu

Skenario Kasus pada Sistem	
Nomor FR	FR-02
Nama	Tambah kohort ibu
Tujuan	Bidan desa dapat menambah data register kohort ibu
Deskripsi	Sistem dapat menambahkan data register kohort ibu.
Aktor	Bidan Desa
Skenario Utama	
Kondisi Awal	Aktor telah login ke sistem
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol menu kohort ibu	2. Sistem menampilkan data register kohort ibu
3. Aktor menekan tombol tambah register ibu	4. Sistem menampilkan form tambah register ibu
5. Aktor memasukkan data pada form tambah register ibu	
6. Aktor menekan tombol simpan	7. Sistem menyimpan data baru ke database
Skenario Alternatif 1: Jika sesi login tidak valid	
	2. Sistem menampilkan halaman login
Skenario Alternatif 2: Jika sesi desa tidak valid	
	7. Sistem memberikan alert bahwa user tidak mempunyai hak akses
Kondisi Akhir	Sistem menyimpan data masukan dari aktor ke database dan sistem redirect ke tampilan lihat register kohort ibu

Pada Tabel 4.9 menggambarkan skenario dari *use case* tambah register kohort ibu. Tujuan dari kebutuhan ini adalah bidan desa dapat melihat data-data yang masuk dalam register kohort ibu. Skenario diawali dengan aktor login ke dalam sistem. Kemudian aktor menekan tombol menu kohort ibu dan sistem akan menampilkan tabel register kohort ibu. Kemudian aktor menekan tombol tambah register ibu dan sistem akan menampilkan form tambah register ibu. Langkah selanjutnya aktor memasukkan data ibu baru pada form tambah register ibu dan menekan tombol simpan sehingga sistem akan menyimpan data baru ke dalam database. Ada dua skenario alternatif yaitu terjadi jika sesi login user tidak valid maka sistem akan menampilkan halaman login sedangkan alternatif kedua yaitu ketika sesi desa tidak valid yakni user tidak memiliki hak pada desa yang dipilih maka sistem akan memberikan alert bahwa user tidak mempunyai hak akses pada register yang dipilih. Kondisi akhir dari skenario ini adalah sistem menyimpan data masukan

dari aktor ke database dan sistem redirect ke tampilan lihat register kohort ibu.

3. [UC-03] Ubah kohort ibu

Tabel 4.10 Skenario *use case* ubah kohort ibu

Skenario Kasus pada Sistem	
Nomor FR	FR-03
Nama	Ubah kohort ibu
Tujuan	Bidan desa dapat mengubah data register kohort ibu
Deskripsi	Sistem dapat mengubah data register kohort ibu.
Aktor	Bidan Desa
Skenario Utama	
Kondisi Awal	Aktor telah login ke sistem
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol menu kohort ibu	2. Sistem menampilkan data register kohort ibu
3. Aktor menekan tombol ubah register	4. Sistem menampilkan form ubah register kohort ibu
5. Aktor memasukkan data baru pada form ubah register kohort ibu	
6. Aktor menekan tombol simpan	7. Sistem menyimpan data baru ke database
Skenario Alternatif 1: Jika sesi login tidak valid	
	2. Sistem menampilkan halaman login
Skenario Alternatif 2: Jika sesi desa tidak valid	
	7. Sistem memberikan alert bahwa user tidak mempunyai hak akses
Kondisi Akhir	Sistem menyimpan data baru dari aktor ke database dan sistem redirect ke tampilan lihat register kohort ibu

Pada Tabel 4.10 menggambarkan skenario dari *use case* ubah register kohort ibu. Tujuan dari kebutuhan ini adalah bidan desa dapat mengubah data register kohort ibu. Skenario diawali dengan aktor login ke dalam sistem. Kemudian aktor menekan tombol menu kohort ibu dan sistem akan menampilkan tabel register kohort ibu. Kemudian aktor menekan tombol ubah register ibu pada salah satu register dan sistem akan menampilkan form ubah register ibu. Langkah selanjutnya aktor memasukkan data baru pada form ubah register ibu dan menekan tombol simpan sehingga sistem akan menyimpan data baru ke dalam database. Ada dua skenario alternatif yaitu terjadi jika sesi login user tidak valid maka sistem akan menampilkan halaman login sedangkan alternatif kedua yaitu ketika sesi desa tidak valid yakni user tidak memiliki hak pada desa yang dipilih maka sistem akan memberikan alert

bahwa user tidak mempunyai hak akses pada register yang dipilih. Kondisi akhir dari skenario ini adalah sistem menyimpan data baru dari aktor ke database dan sistem redirect ke tampilan lihat register kohort ibu

4. [UC-04] Hapus kohort ibu

Tabel 4.11 Skenario *use case* hapus kohort ibu

Skenario Kasus pada Sistem	
Nomor FR	FR-04
Nama	Hapus kohort ibu
Tujuan	Bidan desa dapat menghapus data register kohort ibu
Deskripsi	Sistem dapat menghapus data register kohort ibu.
Aktor	Bidan Desa
Skenario Utama	
Kondisi Awal	Aktor telah login ke sistem
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol menu kohort ibu	2. Sistem menampilkan data register kohort ibu
3. Aktor menekan tombol hapus register	4. Sistem menampilkan konfirmasi hapus register
5. Aktor mengkonfirmasi penghapusan	6. Sistem menghapus data dari database
Skenario Alternatif 1: Jika sesi login tidak valid	
	2. Sistem menampilkan halaman login
Skenario Alternatif 2: Jika sesi desa tidak valid	
	6. Sistem memberikan alert bahwa user tidak mempunyai hak akses
Kondisi Akhir	Sistem menghapus data yang dipilih aktor dan sistem redirect ke tampilan lihat register kohort ibu

Pada Tabel 4.11 menggambarkan skenario dari *use case* hapus register kohort ibu. Tujuan dari kebutuhan ini adalah bidan desa dapat menghapus data register kohort ibu. Skenario diawali dengan aktor login ke dalam sistem. Kemudian aktor menekan tombol menu kohort ibu dan sistem akan menampilkan tabel register kohort ibu. Kemudian aktor menekan tombol hapus register pada salah satu register dan sistem akan menampilkan konfirmasi hapus register ibu. Langkah selanjutnya aktor mengkonfirmasi penghapusan dan sistem akan menghapus data dari database. Ada dua skenario alternatif yaitu terjadi jika sesi login user tidak valid maka sistem akan menampilkan halaman login sedangkan alternatif kedua yaitu ketika sesi desa tidak valid yakni user tidak memiliki hak pada desa yang dipilih maka sistem akan memberikan alert bahwa user tidak mempunyai hak akses pada register yang dipilih. Kondisi akhir dari skenario ini adalah sistem menghapus

data yang dipilih aktor dan sistem redirect ke tampilan lihat register kohort ibu.

5. [UC-33] Lihat rekapan desa

Tabel 4.12 Skenario use case lihat rekapan desa

Skenario Kasus pada Sistem	
Nomor FR	FR-33
Nama	Lihat rekapan desa
Tujuan	Bidan desa dan pegawai puskesmas dapat melihat rekapan desa
Deskripsi	Sistem dapat menampilkan rekapan PWS-KIA pada tiap desa
Aktor	Bidan Desa dan pegawai puskesmas
Skenario Utama	
Aktor Pegawai puskesmas	
Kondisi Awal	Aktor telah login ke sistem
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol menu lihat rekapan desa	2. Sistem menampilkan menu top down pilihan desa
3. Pilih salah satu desa pada sub menu topdown	4. Sistem menampilkan rekapan desa yang dipilih
Skenario Alternatif 1: Jika sesi login tidak valid	
	4. Sistem menampilkan halaman login
Aktor Bidan Desa	
Kondisi Awal	Aktor telah login ke sistem
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol menu lihat rekapan desa	2. Sistem menampilkan rekapan pada desa yang dipilih
Skenario Alternatif 1: Jika sesi login tidak valid	
	2. Sistem menampilkan halaman login
Kondisi Akhir	Sistem menampilkan data rekapan desa

Pada Tabel 4.12 menggambarkan skenario dari *use case* lihat rekapan desa. Tujuan dari kebutuhan ini Bidan desa dan pegawai puskesmas dapat melihat rekapan desa. Skenario diawali dengan aktor login ke dalam sistem. Pada skenario ini terdiri dari dua aktor yaitu bidan desa dan pegawai puskesmas. Pada sisi bidan desa aktor menekan tombol lihat rekapan maka sistem akan menampilkan data rekapan. Pada sisi pegawai puskesmas aktor menekan tombol lihat rekapan desa dan sistem akan menampilkan menu top down pilihan desa dan aktor akan memilih salah satu desa yang akan dilihat sehingga rekapan pada desa yang dipilih yang akan ditampilkan. Skenario alternatif yaitu terjadi jika sesi login user tidak valid maka sistem akan menampilkan halaman login. Kondisi akhir dari skenario ini adalah sistem menampilkan data rekapan desa.

6. [UC-34] Cetak rekapan desa

Tabel 4.13 Skenario *use case* cetak rekapan desa

Skenario Kasus pada Sistem	
Nomor FR	FR-34
Nama	Cetak rekapan desa
Tujuan	Bidan desa dan pegawai puskesmas dapat mencetak rekapan desa
Deskripsi	Sistem dapat mencetak dalam bentuk dokumen PWS-KIA pada tiap desa
Aktor	Bidan Desa dan pegawai puskesmas
Skenario Utama	
Kondisi Awal	Aktor telah login ke sistem dan sistem berada di halaman rekapan desa
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol export	2. Sistem menampilkan jenis tipe file
3. Memilih salah satu tipe file	4. Sistem otomatis mengunduh file export
Kondisi Akhir	Sistem menampilkan data unduhan rekapan desa

Pada Tabel 4.13 menggambarkan skenario dari *use case* cetak rekapan desa. Tujuan dari kebutuhan ini bidan desa dan pegawai puskesmas dapat mencetak rekapan desa. Skenario diawali dengan aktor telah login ke sistem dan sistem berada di halaman rekapan desa. Pada skenario ini terdiri dari dua aktor yaitu bidan desa dan pegawai puskesmas. Namun skenario yang terjadi tidak memiliki perbedaan yaitu aktor akan menekan tombol export kemudian sistem menampilkan jenis tipe file. Kemudian aktor memilih salah satu tipe file dan sistem otomatis mengunduh file export. Kondisi akhir dari skenario ini adalah sistem menampilkan data unduhan rekapan desa.



7. [UC-37] Lihat pemetaan desa

Tabel 4.14 Skenario *use case* pemetaan desa

Skenario Kasus pada Sistem	
Nomor FR	FR-37
Nama	Lihat pemetaan desa
Tujuan	Bidan desa dan pegawai puskesmas dapat melihat pemetaan desa
Deskripsi	Sistem dapat menampilkan pemetaan PWS-KIA pada tiap desa
Aktor	Bidan Desa dan pegawai puskesmas
Skenario Utama	
Aktor Pegawai puskesmas	
Kondisi Awal	Aktor telah login ke sistem
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol menu pemetaan desa	2. Sistem menampilkan menu top down pilihan desa
3. Pilih salah satu desa pada menu topdown	4. Sistem menampilkan pemetaan pada desa yang dipilih
Skenario Alternatif 1: Jika sesi login tidak valid	
	4. Sistem menampilkan halaman login
Aktor Bidan Desa	
Kondisi Awal	Aktor telah login ke sistem
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol menu lihat pemetaan desa	2. Sistem menampilkan pemetaan desa
Skenario Alternatif 1: Jika sesi login tidak valid	
	2. Sistem menampilkan halaman login
Kondisi Akhir	Sistem menampilkan data pemetaan desa

Pada Tabel 4.14 menggambarkan skenario dari *use case* lihat pemetaan desa. Tujuan dari kebutuhan ini bidan desa dan pegawai puskesmas dapat melihat pemetaan desa. Skenario diawali dengan aktor login ke dalam sistem. Pada skenario ini terdiri dari dua aktor yaitu bidan desa dan pegawai puskesmas. Pada sisi bidan desa aktor menekan tombol lihat pemetaan maka sistem akan menampilkan data pemetaan. Pada sisi pegawai puskesmas aktor menekan tombol lihat pemetaan desa dan sistem akan menampilkan menu top down pilihan desa dan aktor akan memilih salah satu desa yang akan dilihat sehingga pemetaan pada desa yang dipilih yang akan ditampilkan. Skenario alternatif yaitu terjadi jika sesi login user tidak valid maka sistem akan menampilkan halaman login. Kondisi akhir dari skenario ini adalah sistem menampilkan data pemetaan desa

8. [UC-38] Cetak pemetaan desa

Tabel 4.15 Skenario *use case* cetak pemetaan desa

Skenario Kasus pada Sistem	
Nomor FR	FR-38
Nama	Cetak pemetaan desa
Tujuan	Bidan desa dan pegawai puskesmas dapat mencetak pemetaan desa
Deskripsi	Sistem dapat mencetak dalam bentuk dokumen PWS-KIA pada tiap desa
Aktor	Bidan Desa dan pegawai puskesmas
Skenario Utama	
Kondisi Awal	Aktor telah login ke sistem dan sistem berada di halaman pemetaan desa
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol export	2. Sistem menampilkan jenis file export
3. Memilih salah satu tipe file	4. Sistem otomatis mengunduh file export
Kondisi Akhir	File akan terunduh sesuai pilihan

Pada Tabel 4.15 menggambarkan skenario dari *use case* cetak rekapan desa. Tujuan dari kebutuhan ini bidan desa dan pegawai puskesmas dapat mencetak pemetaan desa. Skenario diawali dengan aktor telah login ke sistem dan sistem berada di halaman pemetaan desa. Pada skenario ini terdiri dari dua aktor yaitu bidan desa dan pegawai puskesmas. Namun skenario yang terjadi tidak memiliki perbedaan yaitu aktor akan menekan tombol export kemudian sistem menampilkan jenis tipe file. Kemudian aktor memilih salah satu tipe file dan sistem otomatis mengunduh file export. Kondisi akhir dari skenario ini adalah file akan terunduh sesuai pilihan.

4.2.2 Pemodelan *Sequence Diagram*

Diagram *sequence* digunakan untuk menelusuri eksekusi skenario dalam konteks yang sama sebagai diagram komunikasi yang menunjukkan bagaimana aluran data saling berinteraksi dalam beberapa behavior. Diagram *sequence* dibuat berdasarkan *use case* diagram sistem. Pemetaan *use case* dan *sequence* dapat dilihat pada Tabel 4.16 berikut,



Tabel 4.16 Pemetaan *use case* dengan *sequence*

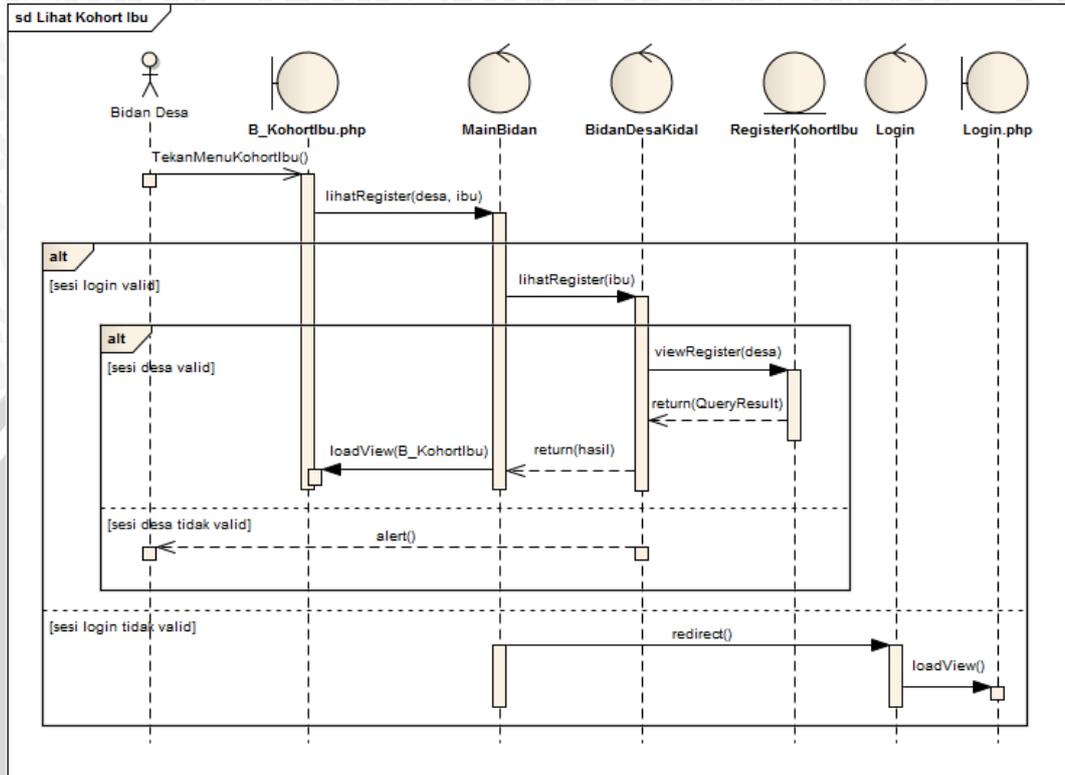
Kode Use Case	Nama Use Case	Kode Sequence	Nama Use Case	Aktor
UC-01	Lihat kohort ibu	SC-01	Lihat kohort ibu	Bidan desa
UC-02	Tambah kohort ibu	SC-02	Tambah kohort ibu	Bidan desa
UC-03	Ubah kohort ibu	SC-03	Ubah kohort ibu	Bidan desa
UC-04	Hapus kohort ibu	SC-04	Hapus kohort ibu	Bidan desa
UC-05	Lihat pelayanan ibu	SC-05	Lihat pelayanan ibu	Bidan desa
UC-06	Tambah pelayanan ibu	SC-06	Tambah pelayanan ibu	Bidan desa
UC-07	Ubah pelayanan ibu	SC-07	Ubah pelayanan ibu	Bidan desa
UC-08	Hapus pelayanan ibu	SC-08	Hapus pelayanan ibu	Bidan desa
UC-09	Lihat kohort bayi	SC-09	Lihat kohort bayi	Bidan desa
UC-10	Tambah kohort bayi	SC-10	Tambah kohort bayi	Bidan desa
UC-11	Ubah kohort bayi	SC-11	Ubah kohort bayi	Bidan desa
UC-12	Hapus kohort bayi	SC-12	Hapus kohort bayi	Bidan desa
UC-13	Lihat pelayanan bayi	SC-13	Lihat pelayanan bayi	Bidan desa
UC-14	Tambah pelayanan bayi	SC-14	Tambah pelayanan bayi	Bidan desa
UC-15	Ubah pelayanan bayi	SC-15	Ubah pelayanan bayi	Bidan desa
UC-16	Hapus pelayanan bayi	SC-16	Hapus pelayanan bayi	Bidan desa
UC-17	Lihat kohort balita	SC-17	Lihat kohort balita	Bidan desa
UC-18	Tambah kohort balita	SC-18	Tambah kohort balita	Bidan desa
UC-19	Ubah kohort balita	SC-19	Ubah kohort balita	Bidan desa
UC-20	Hapus kohort balita	SC-20	Hapus kohort balita	Bidan desa
UC-21	Lihat pelayanan balita	SC-21	Lihat pelayanan balita	Bidan desa
UC-22	Tambah pelayanan balita	SC-22	Tambah pelayanan balita	Bidan desa
UC-23	Ubah pelayanan balita	SC-23	Ubah pelayanan balita	Bidan desa
UC-24	Hapus pelayanan balita	SC-24	Hapus pelayanan balita	Bidan desa
UC-25	Lihat kohort apras	SC-25	Lihat kohort apras	Bidan desa
UC-26	Tambah kohort apras	SC-26	Tambah kohort apras	Bidan desa
UC-27	Ubah kohort apras	SC-27	Ubah kohort apras	Bidan desa
UC-28	Hapus kohort apras	SC-28	Hapus kohort apras	Bidan desa
UC-29	Lihat pelayanan apras	SC-29	Lihat pelayanan apras	Bidan desa

Tabel 4.16 Pemetaan *use case* dengan *sequence* (lanjutan)

Kode Use Case	Nama Use Case	Kode Sequence	Nama Use Case	Aktor
UC-30	Tambah pelayanan apras	SC-30	Tambah pelayanan apras	Bidan desa
UC-31	Ubah pelayanan apras	SC-31	Ubah pelayanan apras	Bidan desa
UC-32	Hapus pelayanan apras	SC-32	Hapus pelayanan apras	Bidan desa
UC-33	Lihat rekapan desa	SC-33	Lihat rekapan desa	Bidan desa dan pegawai puskesmas
UC-34	Cetak rekapan desa	SC-34	Cetak rekapan desa	Bidan desa dan pegawai puskesmas
UC-35	Lihat rekapan kecamatan	SC-35	Lihat rekapan kecamatan	Pegawai puskesmas
UC-36	Cetak rekapan kecamatan	SC-36	Cetak rekapan kecamatan	Pegawai puskesmas
UC-37	Lihat pemetaan desa	SC-37	Lihat pemetaan desa	Bidan desa dan pegawai puskesmas
UC-38	Cetak pemetaan desa	SC-38	Cetak pemetaan desa	Bidan desa dan pegawai puskesmas
UC-39	Lihat pemetaan kecamatan	SC-39	Lihat pemetaan kecamatan	Pegawai puskesmas
UC-40	Cetak pemetaan kecamatan	SC-40	Cetak pemetaan kecamatan	Pegawai puskesmas
UC-41	Perbarui rekapan	SC-41	Perbarui rekapan	Bidan desa dan pegawai puskesmas
UC-42	Perbarui pemetaan	SC-42	Perbarui pemetaan	Bidan desa dan pegawai puskesmas
UC-43	Lihat user	SC-43	Lihat user	Pegawai puskesmas
UC-44	Tambah user	SC-44	Tambah user	Pegawai puskesmas
UC-45	Ubah user	SC-45	Ubah user	Pegawai puskesmas
UC-46	Hapus user	SC-46	Hapus user	Pegawai puskesmas
UC-47	Login	SC-47	Login	User
UC-48	Logout	SC-48	Logout	Bidan desa dan pegawai puskesmas

Dari pemetaan tersebut maka dapat dimodelkan dalam diagram sequence. Berikut beberapa diagram sequence pada SIPKIA,

1. [SC_01] Lihat Kohort Ibu

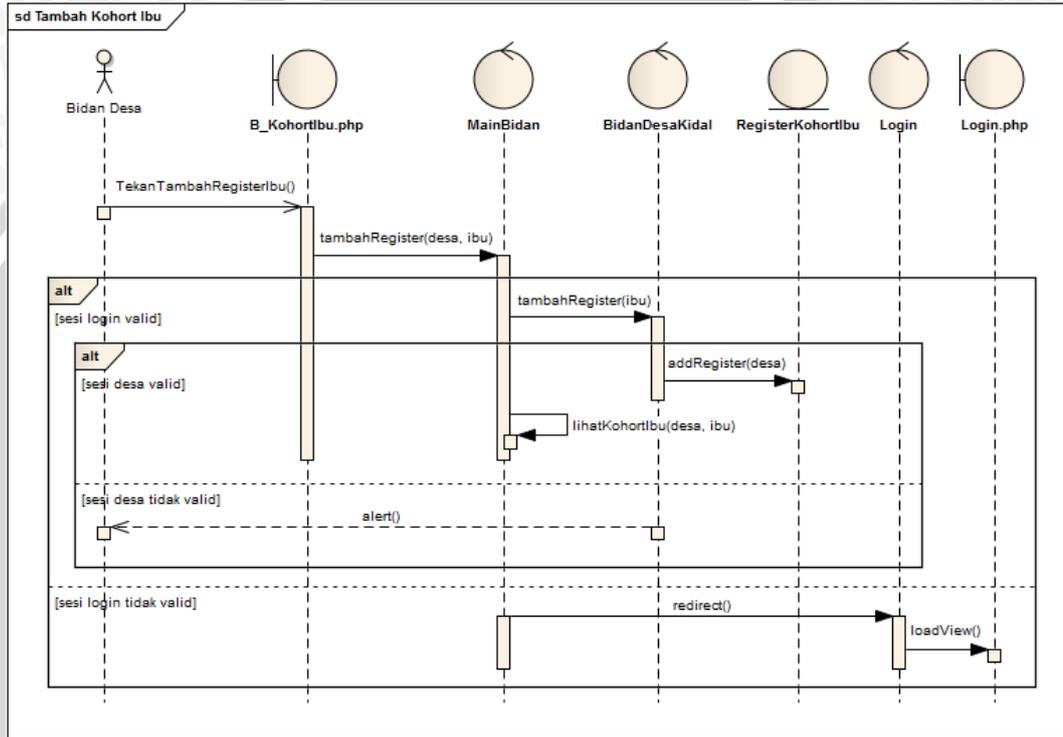


Gambar 4.5 Diagram sequence lihat kohort ibu

Diagram sequence lihat kohort ibu pada Gambar 4.5 diawali aktor bidan desa menekan tombol menu kohort ibu pada boundary B_KohortIbu.php. Selanjutnya sistem akan mendirect ke fungsi lihatRegister() pada objek MainBidan sebagai controller. Setelah alur masuk dalam objek MainBidan sistem akan mengecek validasi user apakah sesi login valid. Jika sesi login tidak valid maka sistem akan direct ke controller login dan menampilkan halaman login. Tetapi jika sesi login valid maka sistem akan meneruskan ke alur selanjutnya. Pada fungsi lihatRegister() di MainBidan terdapat parameter desa dengan jenis kohort, dan pada kasus ini jenis kohort adalah ibu. Setelah itu sistem akan mengecek objek bidan desa mana yang akan dibuat berdasarkan parameter desa. Pada kasus ini asumsinya adalah bidan desa kidal sehingga MainBidan akan memanggil fungsi lihatRegister pada objek BidanDesaKidal dengan parameter ibu. Ketika masuk objek BidanDesaKidal sistem akan mengecek apakah sesi desa valid dalam arti yang meminta adalah benar-benar bidan desa kidal bukan desa lain. Jika ternyata sesi desa tidak valid maka sistem akan memberikan alert atau pesan jika pengguna tidak memiliki hak akses ke bidan desa tersebut. Tetapi jika sistem valid maka akan diteruskan ke tahap selanjutnya. Selanjutnya karena jenis kohort yang diminta adalah ibu maka

BidanDesaKidal akan membuat objek RegisterKohortIbu dan memanggil fungsi viewRegister(). Setelah itu fungsi viewRegister() akan mengembalikan nilai yang diminta kepada objek BidanDesaKidal yang selanjutnya bidanDesaKidal juga meneruskan nilai kembalian ke objek MainBidan dan pada akhirnya MainBidan memanggil boundary B_KohortIbu.php dan mengirim data kembalian dari BidanDesaKidal sehingga tampilan halaman lihat kohort ibu denga data-data register kohort ibu.

2. [SC_02] Tambah Kohort Ibu



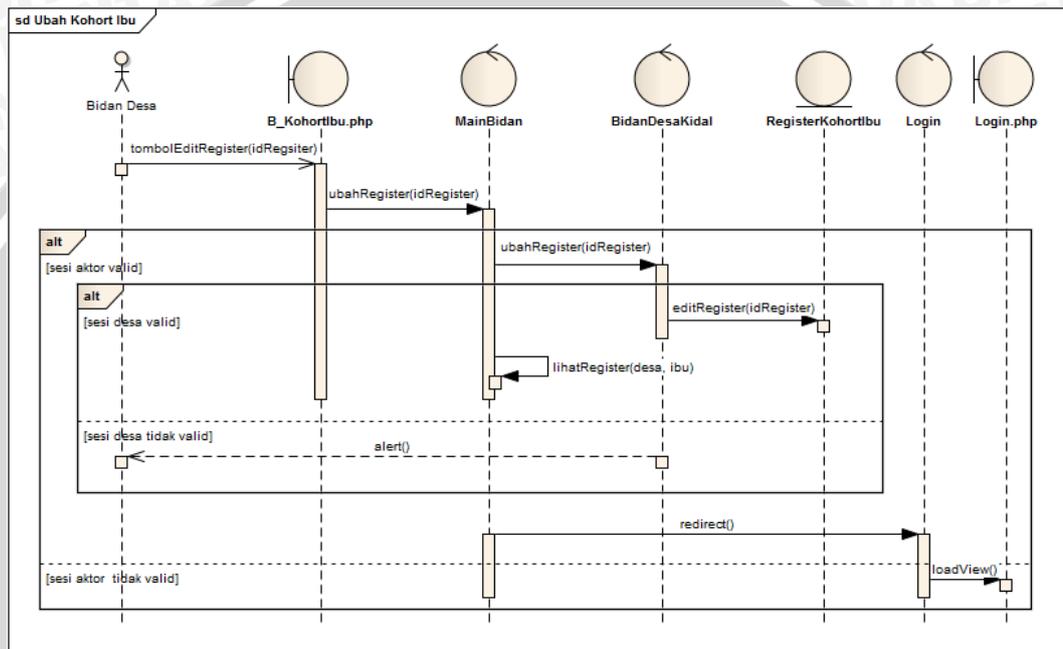
Gambar 4.6 Diagram sequence tambah kohort ibu

Diagram sequence tambah kohort ibu pada Gambar 4.6 diawali aktor bidan desa menekan tombol tambah register dan tombol simpan pada form tambah register kohort ibu di boundary B_KohortIbu.php. Selanjutnya sistem akan mendirect ke fungsi tambahRegister() pada objek MainBidan sebagai controller. Setelah alur masuk dalam objek MainBidan sistem akan mengecek validasi user apakah sesi login valid. Jika sesi login tidak valid maka sistem akan direct ke controller login dan menampilkan halaman login. Tetapi jika sesi login valid maka sistem akan meneruskan ke alur selanjutnya. Pada fungsi tambahRegister() di MainBidan terdapat parameter desa dengan jenis kohort, dan pada kasus ini jenis kohort adalah ibu. Setelah itu sistem akan mengecek objek bidan desa mana yang akan dibuat berdasarkan parameter desa. Pada kasus ini asumsinya adalah bidan desa kidal sehingga MainBidan akan memanggil fungsi tambahRegister pada objek BidanDesaKidal dengan parameter ibu. Ketika masuk objek BidanDesaKidal sistem akan mengecek



apakah sesi desa valid dalam arti yang meminta adalah benar-benar bidan desa kidal bukan desa lain. Jika ternyata sesi desa tidak valid maka sistem akan memberikan alert atau pesan jika pengguna tidak memiliki hak akses ke bidan desa tersebut. Tetapi jika sistem valid maka akan diteruskan ke tahap selanjutnya. Selanjutnya karena jenis kohort yang akan ditambahkan adalah ibu maka BidanDesaKidal akan membuat objek RegisterKohortIbu dan memanggil fungsi addRegister(). Setelah itu fungsi addRegister() akan mengelola data masukan dari aktor ke dalam database.

[SC_03] Ubah Kohort Ibu

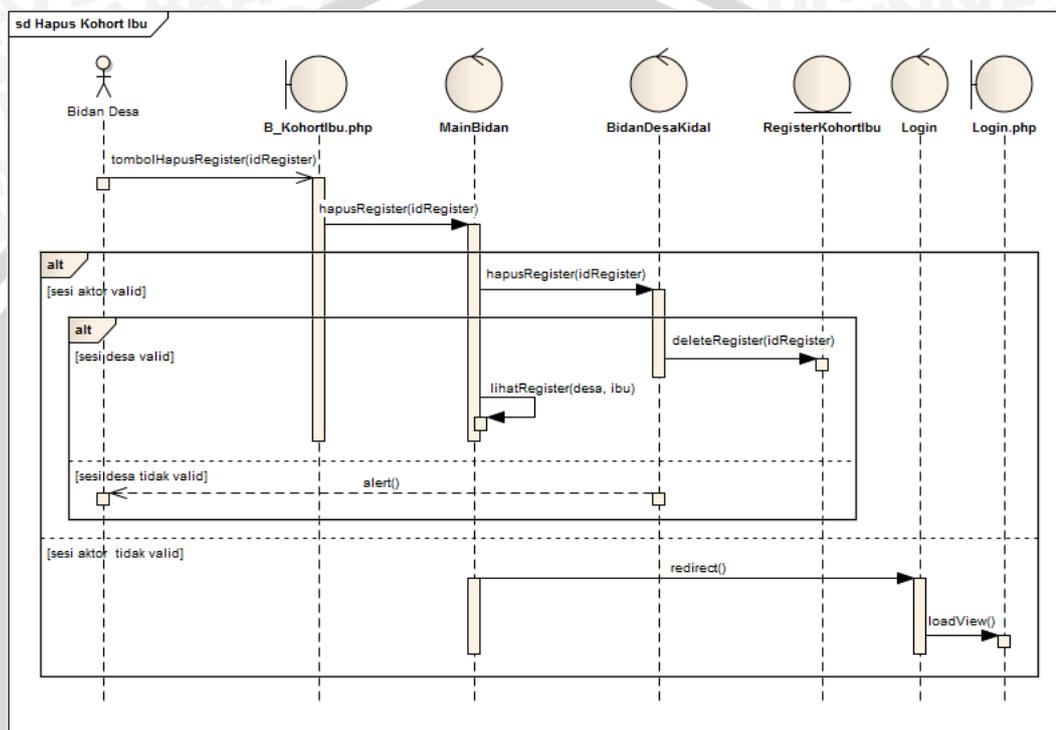


Gambar 4.7 Diagram sequence ubah kohort ibu

Diagram sequence ubah kohort ibu pada Gambar 4.7 diawali aktor bidan desa menekan tombol edit register dan tombol simpan pada form ubah register kohort ibu di boundary B_KohortIbu.php. Selanjutnya sistem akan mendirect ke fungsi ubahRegister() pada objek MainBidan sebagai controller. Setelah alur masuk dalam objek MainBidan sistem akan mengecek validasi user apakah sesi login valid. Jika sesi login tidak valid maka sistem akan direct ke controller login dan menampilkan halaman login. Tetapi jika sesi login valid maka sistem akan meneruskan ke alur selanjutnya. Pada fungsi ubahRegister() di MainBidan terdapat parameter desa dengan jenis kohort, dan pada kasus ini jenis kohort adalah ibu. Setelah itu sistem akan mengecek objek bidan desa mana yang akan dibuat berdasarkan parameter desa. Pada kasus ini asumsinya adalah bidan desa kidal sehingga MainBidan akan memanggil fungsi ubahRegister pada objek BidanDesaKidal dengan parameter ibu. Ketika masuk objek BidanDesaKidal sistem akan mengecek apakah sesi desa valid dalam arti yang meminta adalah benar-benar bidan desa kidal bukan desa lain. Jika ternyata sesi desa tidak valid maka sistem akan memberikan alert atau pesan jika pengguna tidak memiliki

hak akses ke bidan desa tersebut. Tetapi jika sistem valid maka akan diteruskan ke tahap selanjutnya. Selanjutnya karena jenis kohort yang akan diubah adalah ibu maka BidanDesaKidal akan membuat objek RegisterKohortIbu dan memanggil fungsi editRegister(). Setelah itu fungsi editRegister() akan mengelola data perubahan yang telah diedit oleh aktor untuk dimasukkan ke dalam database.

3. [SC_04] Hapus Kohort Ibu



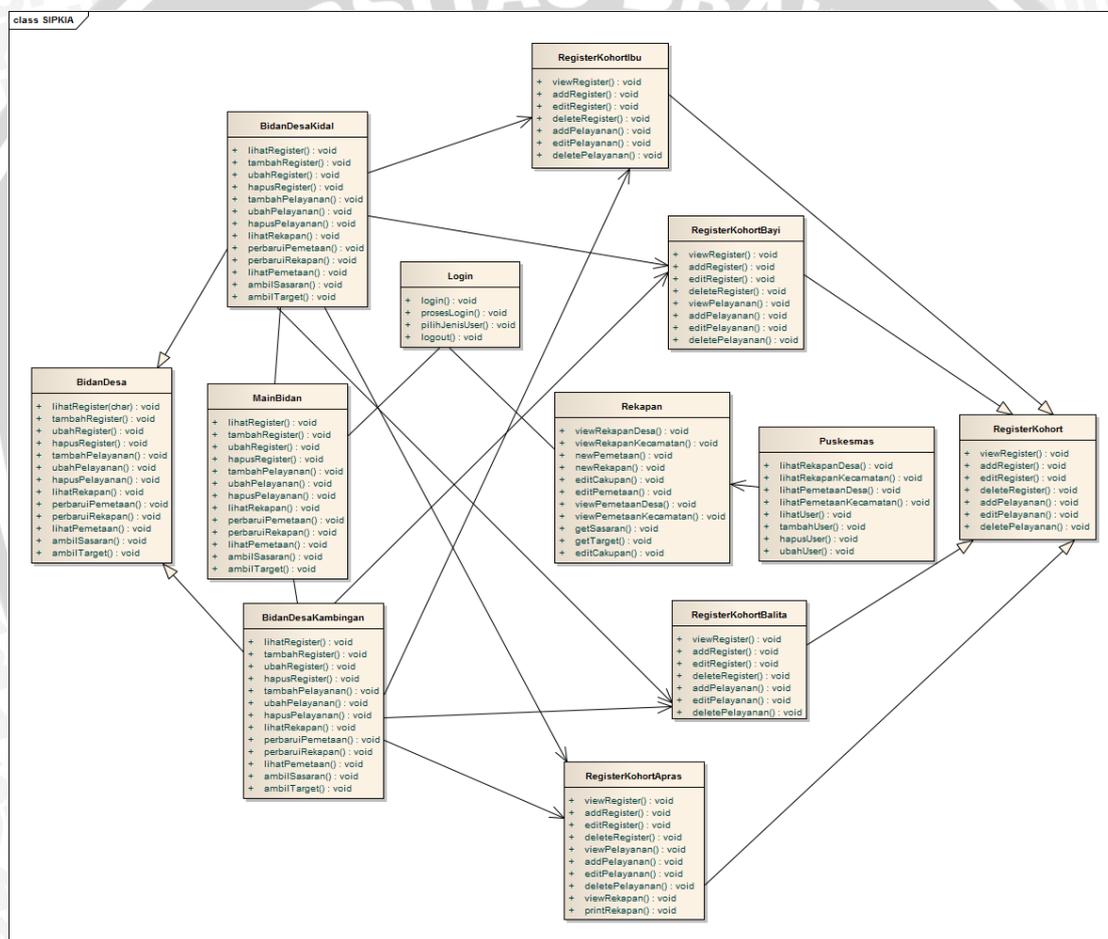
Gambar 4.8 Diagram sequence hapus kohort ibu

Diagram sequence hapus kohort ibu pada Gambar 4.8 diawali aktor bidan desa menekan tombol hapus register dan tombol konfirmasi ya pada boundary B_KohortIbu.php. Selanjutnya sistem akan mendirect ke fungsi hapusRegister() pada objek MainBidan sebagai controller. Setelah alur masuk dalam objek MainBidan sistem akan mengecek validasi user apakah sesi login valid. Jika sesi login tidak valid maka sistem akan direct ke controller login dan menampilkan halaman login. Tetapi jika sesi login valid maka sistem akan meneruskan ke alur selanjutnya. Pada fungsi hpausRegister() di MainBidan terdapat parameter desa dengan jenis kohort, dan pada kasus ini jenis kohort adalah ibu. Setelah itu sistem akan mengecek objek bidan desa mana yang akan dibuat berdasarkan parameter desa. Pada kasus ini asumsinya adalah bidan desa kidal sehingga MainBidan akan memanggil fungsi hapusRegister pada objek BidanDesaKidal dengan parameter ibu. Ketika masuk objek BidanDesaKidal sistem akan mengecek apakah sesi desa valid dalam arti yang meminta adalah benar-benar bidan desa kidal bukan desa lain. Jika ternyata sesi desa tidak valid maka sistem

akan memberikan alert atau pesan jika pengguna tidak memiliki hak akses ke bidan desa tersebut. Tetapi jika sistem valid maka akan diteruskan ke tahap selanjutnya. Selanjutnya karena jenis kohort yang akan dihapus adalah ibu maka BidanDesaKidal akan membuat objek RegisterKohortIbu dan memanggil fungsi deleteRegister(). Setelah itu fungsi deleteRegister() akan menghapus data register ibu yang dipilih dari database.

4.2.3 Pemodelan Class Diagram

Dari pendefinisian *use case* dan *sequence* maka dapat dilakukan analisis kelas-kelas apa yang perlu dibuat dalam pembangunan SIPKIA. Analisis diagram kelas dari Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method* dapat dilihat pada Gambar 4.9 berikut,



Gambar 4.9 Analisis class diagram

Dari hasil analisis yang dilakukan pada Gambar 4.9 kelas yang dibuat antara lain adalah kelas BidanDesa yang akan diextends oleh masing-masing bidan desa pada sebuah desa. Sampel dalam kelas diagram tersebut adalah BidanDesaKidal dan BidanDesaKambingan. Namun dalam implementasinya jumlah bidan desa ada lima belas. Selanjutnya yaitu kelas RegisterKohort yang akan di extends oleh kelas



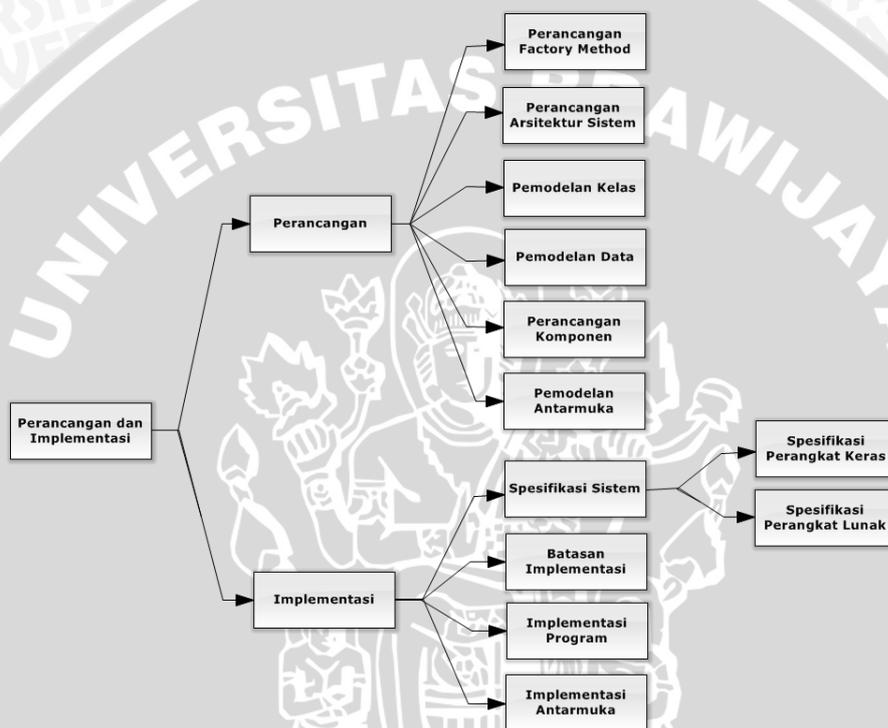
Repository.ub.ac.id

RegisterKohortIbu, RegisterKohortBayi, RegisterKohortBalita dan RegisterKohortApras. Sedangkan kelas yang akan bertindak sebagai kelas utama atau client adalah kelas MainBidan dan Puskesmas. Kelas tersebut yang akan memanggil kelas-kelas yang telah dibuat dari awal. Selain itu terdapat kelas Login yang akan mengelola validasi user dan ada kelas Rekapian yang akan mengelola rekapian dan pemetaan dari semua register kohort.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini membahas perancangan dan implementasi berdasarkan hasil analisis kebutuhan SIPKIA. Proses perancangan sistem yang dilakukan meliputi perancangan pola *factory method*, perancangan arsitektur sistem, pemodelan kelas, pemodelan data, perancangan komponen dan perancangan antarmuka pengguna. Implementasi sistem terdiri dari penjelasan tentang spesifikasi sistem, batasan implementasi, implementasi program, dan implementasi antarmuka. Tahapan-tahapan pada proses implementasi dapat dilihat pada Gambar 5.1 berikut,



Gambar 5.1 Diagram perancangan dan implementasi

5.1 Perancangan Sistem

Proses perancangan sistem yang dilakukan meliputi perancangan pola *factory method*, perancangan arsitektur sistem, pemodelan kelas, pemodelan data, dan perancangan antarmuka pengguna.

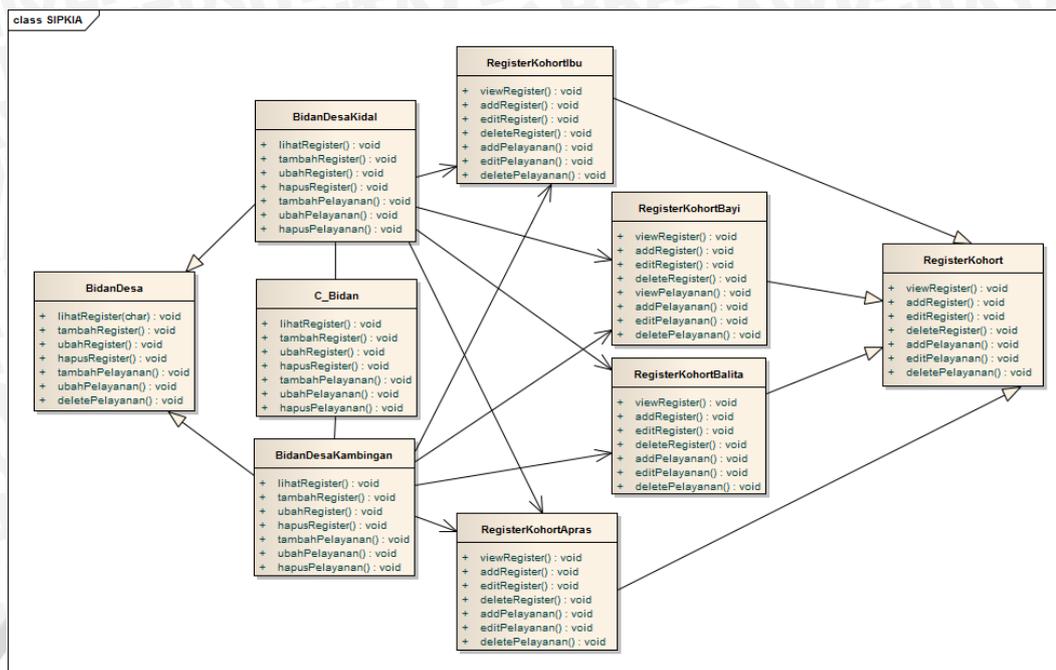
5.1.1 Perancangan Pola *Factory Method*

Dalam penelitian ini menggunakan perancangan pola *factory method* yang memiliki beberapa komponen penyusun pola. Komponen-komponen dalam pola *factory method* adalah *creator*, *concrete creator*, *product*, dan *concrete product*. Penggunaan komponen-komponen dalam Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak ini dapat dilihat pada Tabel 5.1 berikut,

Tabel 5.1 Komponen perancangan pola *factory method*

Komponen	Kelas	Deskripsi
Creator	BidanDesa	Bidan desa merupakan <i>creator</i> dalam sistem ini yang akan membuat register kohort pada masing-masing desa
<i>Concrete Creator</i>	BidanDesaTumpang, BidanDesaMalangsuko, BidanDesaJeru, BidanDesaWringinsongo, BidanDesaTulusbesar, BidanDesaBenjor, BidanDesaDuwet, BidanDesaDampul, BidanDesaPulungdowo, BidanDesaPandanajeng, BidanDesaBokor, BidanDesaSlamet, BidanDesaKidal, BidanDesaKambangan, BidanDesaNgingit	Bidan desa pada desa masing-masing adalah <i>creator</i> nyata yang akan mengelola register kohort yang terdiri dari kohort ibu, bayi, balita dan anak.
Product	RegisterKohort	Register kohort adalah produk yang akan dibuat oleh bidan desa
<i>Concrete Product</i>	RegisterKohortIbu, RegisterKohortBayi, RegisterKohortBalita, RegisterKohortApras	Produk nyata yang dibuat oleh bidan desa pada masing-masing desa yaitu register kohort ibu, register kohort bayi, register kohort balita dan register kohort apras
Client	C_Bidan	Client yang akan memanggil fungsi <i>factory method</i> yang disediakan oleh <i>concrete creator</i>

Dari komponen-komponen yang terbentuk tersebut maka dapat dilakukan pemodelan kelas untuk pola perancangan *factory method*. Bagian dari pemodelan kelas pola perancangan *factory method* pada SIPKIA dapat dilihat pada gambar 5.2 berikut,



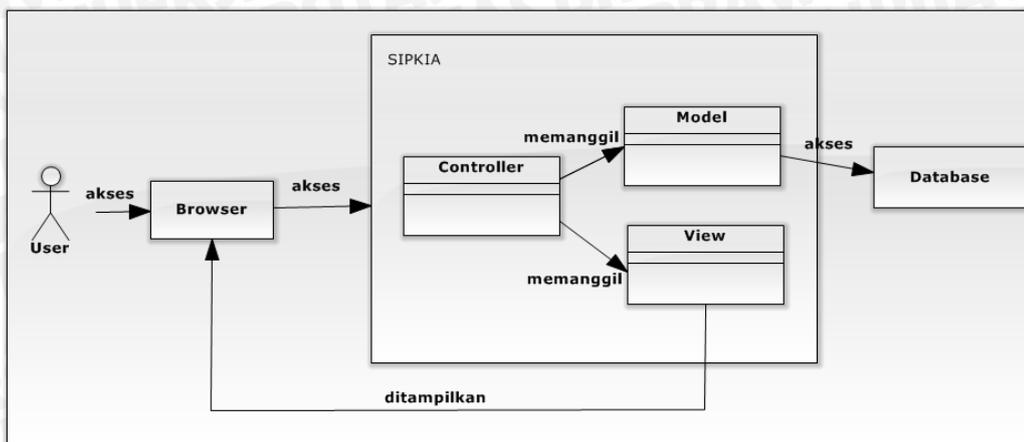
Gambar 5.2 Class diagram perancangan pola factory method

Gambar 5.2 merupakan diagram kelas SIPKIA dengan perancangan factory method. Sesuai dengan komponen yang ada dalam pola factory method SIPKIA juga memiliki komponen-komponen tersebut. Komponen *creator* pada diagram kelas tersebut berupa kelas BidanDesa dimana kelas ini akan di inherit (extends) oleh komponen *concrete creator* yaitu dalam hal ini diwakili oleh kelas BidanDesaKida dan kelas BidanDesaKambangan. Pada kedua kelas ini menyediakan method factory yang akan menerima panggilan dari client. Client akan memanggil method factory tersebut dan method tersebut akan menruskan ke kelas yang dibutuhkan tanpa diketahui oleh client kelas apa yang akan dipanggil. Kelas yang akan dipanggil oleh method tersebut merupakan komponen *concrete product* yang inherit (extends) ke komponen *product*. Pada perancangan SIPKIA kelas *product* adalah RegisterKohort, sedangkan *concrete productnya* adalah kelas RegisterKohortIbu, RegisterKohortBayi, RegisterKohortBalita dan RegisterKohortApras. Dan Client yang akan memanggil factory method yang disediakan oleh concrete cretor adalah kelas C_Bidan.

5.1.2 Perancangan Arsitektur Sistem

Perancangan arsitektur dilakukan untuk menggambarkan jalanya sistem dari awal, yakni dari proses pengguna mengakses sistem, pemrosesan data, hingga menghasilkan *output* bagi pengguna. Perancangan arsitektur Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak pemetaan dapat dilihat pada Gambar 5.3 berikut,



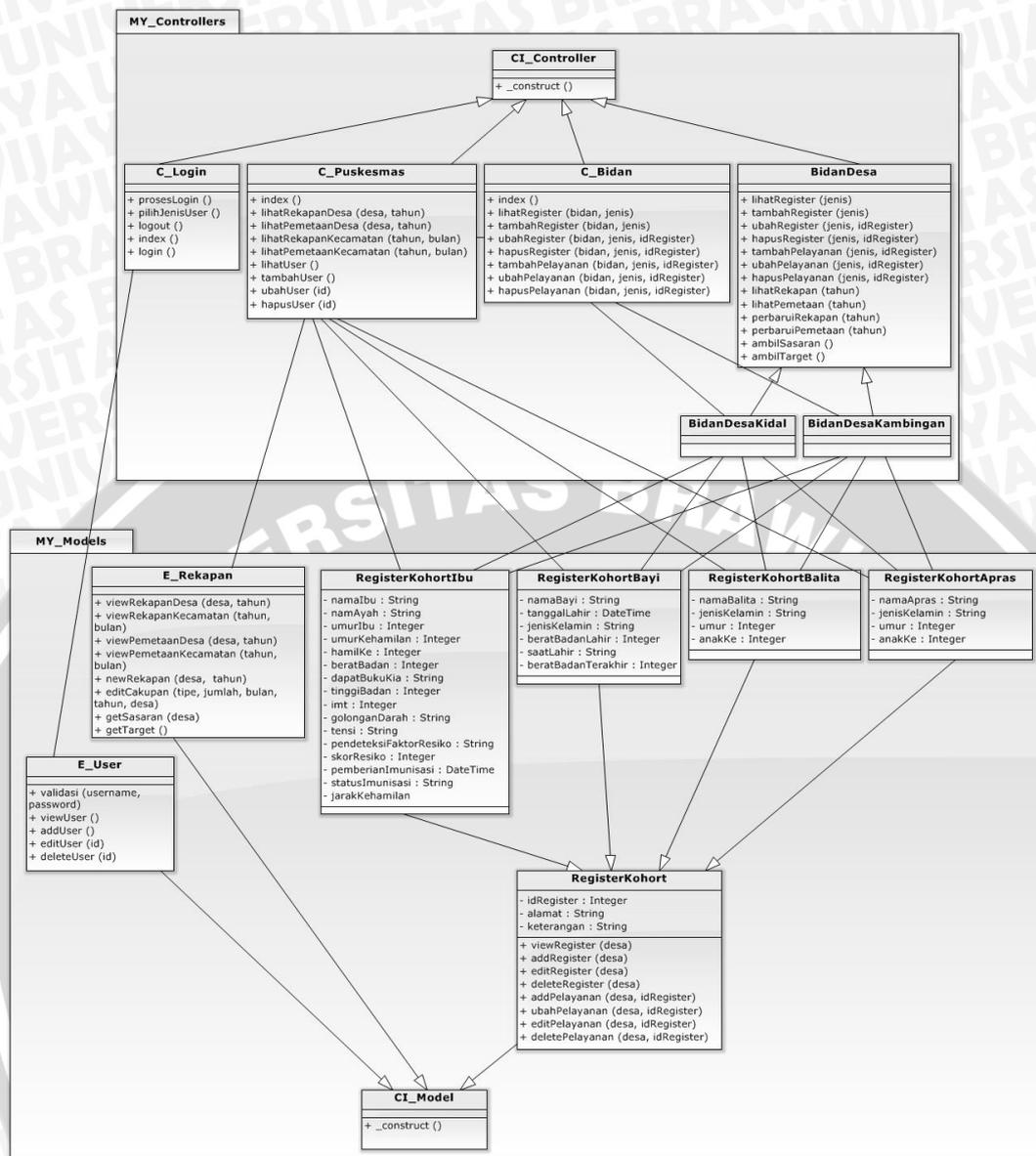


Gambar 5.3 Arsitektur sistem

Pada Gambar 5.3 sistem ini dapat dijalankan oleh pengguna melalui beberapa proses, yaitu user mengakses sistem melalui browser. Sistem menggunakan framework Code Igniter (CI) sehingga browser akan mengakses sistem sesuai file pada Code Igniter yang menggunakan MVC (Model View Controller). Dalam CI, *controller* akan mengambil model yang berisi fungsi-fungsi untuk pengolahan data, diantaranya proses CRUD (*Create, Read, Update, Delete*). Data yang telah dipanggil di dalam model akan ditampilkan view didalam browser.

5.1.3 Pemodelan Kelas

Diagram kelas menggambarkan sistem dan relasi-relasi yang ada didalamnya. Suatu kelas dalam sistem memiliki relasi dengan kelas lainnya dan dapat memiliki pewarisan (*inheritance*) atau ketergantungan (*dependency*) dengan komponen yang dimilikinya. Sistem informasi dan pemetaan menggunakan MVC (*Model, View, Controller*) sehingga merepresentasikan ketiga komponen tersebut. Diagram *model* SIPKIA bidang desa dapat dilihat pada Gambar 5.4 berikut,



Gambar 5.4 Class diagram SIPKIA

Berdasarkan gambar 5.4 diagram kelas SIPKIA terdiri dari dua *package* kelas yaitu MY_Models dan MY_Controllers. Pada paket MY_Models terdiri dari kelas-kelas model yang semuanya generalisasi atau mewarisi dari kelas CI_Model. Kelas-kelas pada package MY_Models berfungsi untuk operasi data pada database baik memasukkan data ke database atau mengambil data dari database. Kelas model yakni kelas CI_Model memiliki anak kelas E_User yang fungsinya untuk keperluan kelola data user dan validasi user ketika ingin login ke SIPKIA, kelas E_Rekapan berfungsi untuk pengelolaan data rekapan dan pemetaan PWS-KIA, kelas RegisterKohort sebagai parent class dari masing-masing kohort, kelas RegisterKohortIbu berfungsi pengelolaan data register kohort ibu, kelas RegisterKohortBayi berfungsi pengelolaan data register kohort bayi, kelas



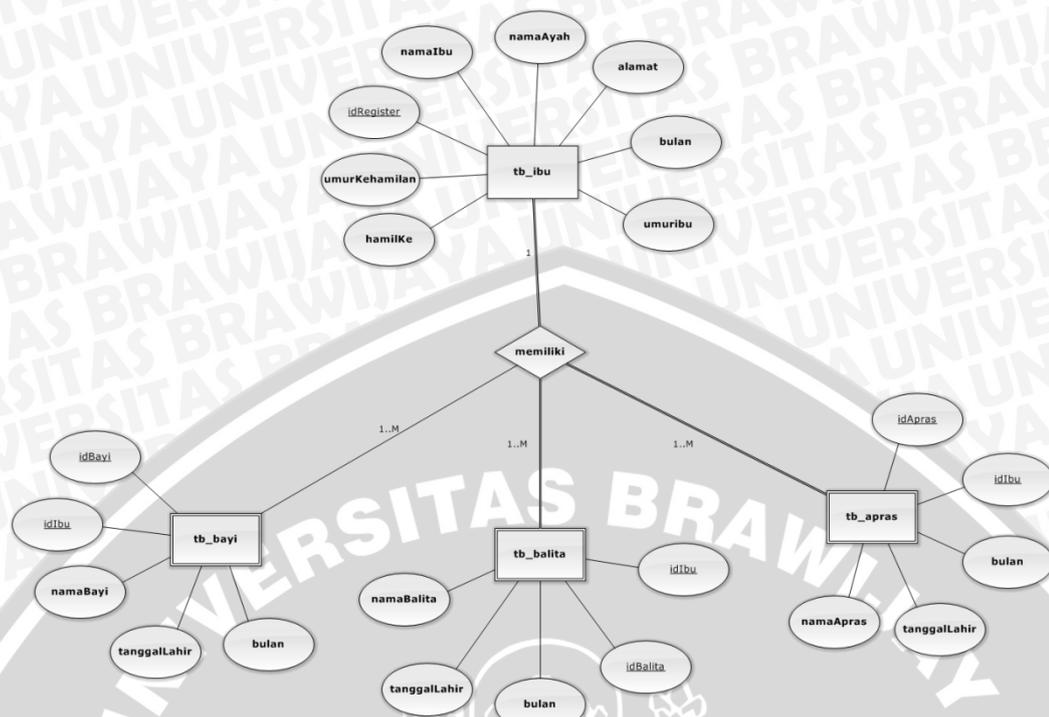
RegisterKohortBalita berfungsi pengelolaan data register kohort balita dan kelas RegisterKohortApras berfungsi pengelolaan data register kohort apras.

Package yang satunya adalah MY_Controllers yang terdiri dari kelas-kelas controller yang semuanya generalisasi atau mewarisi dari kelas CI_Controller. Kelas-kelas pada package MY_Controllers berfungsi penghubung dan pengontrol dari data dalam model dengan tampilan pada view. Kelas controller yakni kelas CI_Controller memiliki anak kelas C_Login yang fungsinya untuk keperluan operasi ketika user ingin login ke SIPKIA, kelas C_Puskesmas berfungsi untuk manajemen pada fitur yang dimiliki pegawai puskesmas, C_Bidan berfungsi untuk manajemen pada yang dimiliki bidan desa, kelas BidanDesa berfungsi sebagai parent class, kelas BidanDesaKidal dan BidanDesaKambingan berfungsi untuk manajemen pada fitur register kohort ibu, bayi, balita dan apras. Pada masing-masing kelas controller tersebut semuanya terhubung dengan model yang memiliki kepentingan.

5.1.4 Pemodelan Data

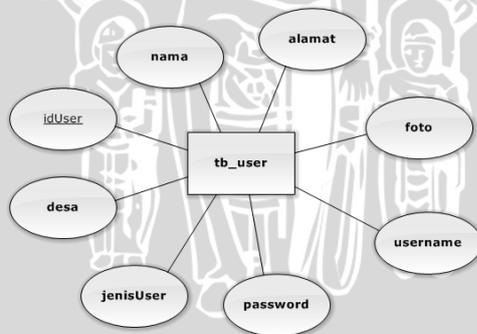
Pemodelan data bertujuan merancang manajemen data yang akan digunakan dalam sistem. Manajemen data mencakup pemodelan basis data yang terdiri dari data relevan yang akan digunakan dalam sistem dan didukung oleh *Database Management System* (DBMS). Pemodelan data dalam sistem ini digambarkan dalam *Entity Relationship Diagram* (ERD). ERD SIPKIA dapat dilihat pada Gambar 5.5 dan Gambar 5.6 berikut,





Gambar 5.5 ERD pengelolaan data register kohort

Pada ERD Gambar 5.5 tersebut **tb_ibu** mempunyai relasi dengan **tb_bayi**, **tb_balita** dan **tb_apras** dimana masing-masing relasi tersebut akan mengelola data register kohort ibu, bayi, balita dan apras. Register ibu merupakan parent dari register bayi, balita dan apras.

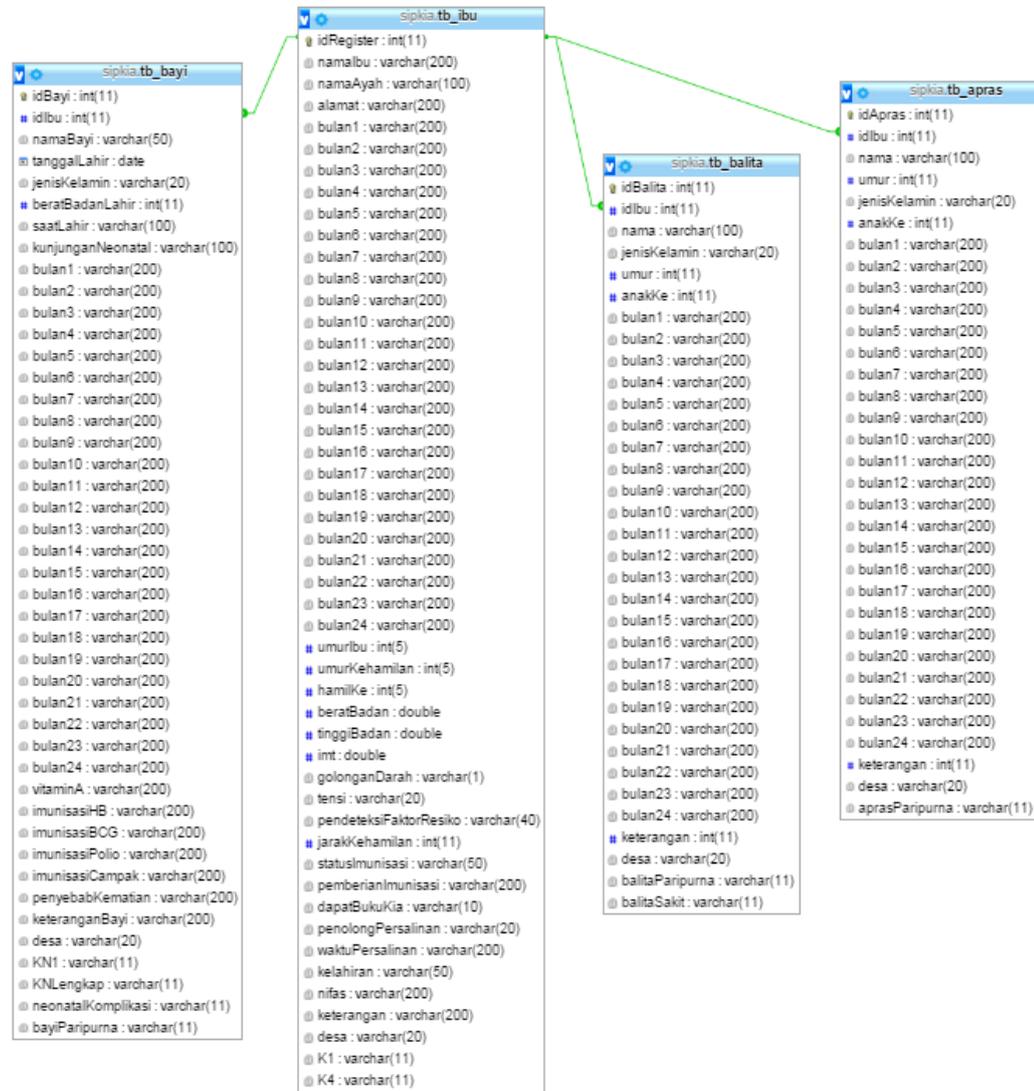


Gambar 5.6 ERD pengelolaan data user sistem

Pada ERD pengelolaan data user sistem pada Gambar 5.6 terdiri dari **idUser**, **username**, **password** dan atribut lainnya yang menggambarkan seorang user dari sistem.

5.1.4.1 Pemodelan Database

Berdasarkan dari ERD diagram pada maka dapat dimodelkan dalam database. Dalam pemodelan database terdiri dari pemodelan data register kohort dan pemodelan data user dan pemetaan SIPKIA. Pemodelan database dapat dilihat pada gambar 5.7 dan 5.8 berikut,



Gambar 5.7 ERD pengelolaan data register kohort

Pada tb_ibu mempunyai relasi dengan tb_bayi, tb_balita dan apras. Relasi tersebut berdasarkan ERD yang telah dibuat yaitu seorang ibu adalah parent dari bayi, balita, dan apras. Pemodelan database pada Gambar 5.7 berisi atribut-atribut dalam register kohort seperti profil register, data pemeriksaan, data pelayanan dan data-data rekam medis lainnya.

sipkia.tb_user	sipkia.tb_rekapan	sipkia.tb_pemetaan
<ul style="list-style-type: none"> idUser : int(11) nama : varchar(100) alamat : varchar(200) foto : varchar(200) username : varchar(40) password : varchar(200) jenisUser : varchar(20) desa : varchar(20) 	<ul style="list-style-type: none"> idRekapan : int(11) bulan : int(11) tahun : int(11) desa : varchar(20) K1Lalu : int(11) K1Ini : int(11) K1Kom : int(11) K1Presentase : double K1R : int(11) K4Lalu : int(11) K4Ini : int(11) K4Kom : int(11) K4Presentase : double K4R : int(11) deteksiRestiLalu : int(11) deteksiRestiIni : int(11) deteksiRestiKom : int(11) deteksiRestiPresentase : double deteksiRestiR : int(11) restiKomLalu : int(11) restiKomIni : int(11) restiKomKom : int(11) restiKomPresentase : double restiKomR : int(11) komKebLalu : int(11) komKebIni : int(11) komKebKom : int(11) komKebPresentase : double komKebR : int(11) persalinanNakesLalu : int(11) persalinanNakesIni : int(11) persalinanNakesKom : int(11) persalinanNakesPresentase : double persalinanNakesR : int(11) persalinanFaskesLalu : int(11) persalinanFaskesIni : int(11) persalinanFaskesKom : int(11) persalinanFaskesPresentase : double persalinanFaskesR : int(11) pelayananNifasLalu : int(11) pelayananNifasIni : int(11) pelayananNifasKom : int(11) pelayananNifasPresentase : int(11) 	<ul style="list-style-type: none"> idPemetaan : int(11) K1Lalu : varchar(20) K4Lalu : varchar(20) deteksiRestiLalu : varchar(20) restiKomLalu : varchar(20) komKebLalu : varchar(20) persalinanNakesLalu : varchar(20) persalinanFaskesLalu : varchar(20) pelayananNifasLalu : varchar(20) KN1Lalu : varchar(20) KNLengkapLalu : varchar(20) neonatalKomplikasiLalu : varchar(20) bayiParipurnaLalu : varchar(20) balitaParipurnaLalu : varchar(20) aprasParipurnaLalu : varchar(20) balitaSakitLalu : varchar(20) desa : varchar(20) bulan : int(11) tahun : int(11) K1Target : varchar(20) K4Target : varchar(20) deteksiRestiTarget : varchar(20) restiKomTarget : varchar(20) komKebTarget : varchar(20) persalinanNakesTarget : varchar(20) persalinanFaskesTarget : varchar(20) pelayananNifasTarget : varchar(20) KN1Target : varchar(20) KNLengkapTarget : varchar(20) neonatalKomplikasiTarget : varchar(20) bayiParipurnaTarget : varchar(20) balitaParipurnaTarget : varchar(20) aprasParipurnaTarget : varchar(20) balitaSakitTarget : varchar(20) K1Status : varchar(20) K4Status : varchar(20) deteksiRestiStatus : varchar(20) restiKomStatus : varchar(20) komKebStatus : varchar(20) persalinanNakesStatus : varchar(20) persalinanFaskesStatus : varchar(20) pelayananNifasStatus : varchar(20) KN1Status : varchar(20)
<ul style="list-style-type: none"> idSasaran : int(11) desa : varchar(50) sasaranBumil : int(11) sasaranBumilRisti : int(11) sasaranBulin : int(11) sasaranBayi : int(11) sasaranNeoRisti : int(11) sasaranBalita : int(11) sasaranApras : int(11) 		
<ul style="list-style-type: none"> tahun : int(11) K1 : int(11) K4 : int(11) deteksiResti : int(11) restiKom : int(11) komKeb : int(11) persalinanNakes : int(11) persalinanFaskes : int(11) pelayananNifas : int(11) KN1 : int(11) KNLengkap : int(11) neonatalKomplikasi : int(11) bayiParipurna : int(11) balitaParipurna : int(11) aprasParipurna : int(11) balitaSakit : int(11) 		

Gambar 5.8 ERD pengelolaan data user dan pemetaan sipkia

Pada ERD Gambar 5.8 tabel tb_user merupakan tabel yang berisi pengelolaan data dari user SIPKIA. Tabel tb_sasaran berisi data sasaran ibu hamil, ibu bersalin dan sasaran bayi. Tabel tb_target berisi data target dari masing-masing cakupan PWS-KIA. Selanjutnya tb_rekapan dan tb_pemetaan adalah tabel yang mengelola data rekapan dari register kohort dan pemetaan kesehatan ibu dan anak.

5.1.4.2 Struktur Tabel

Berdasarkan rancangan database diatas, berikut struktur tabel yang terdapat pada database sistem dan penjelasannya dapat dilihat pada Tabel 5.2, Tabel 5.3, dan Tabel 5.4 berikut.

1. Tabel tb_ibu

Tabel 5.2 Atribut data register ibu

<i>Field</i>	<i>Type</i>	<i>Deskripsi</i>	<i>Keterangan</i>
idRegister	integer	Nomor identitas dari masing-masing register	PK
namaIbu	varchar(200)	Nama ibu	
namaAyah	varchar(100)	Nama ayah	
alamat	varchar(200)	Alamat	
umurIbu	integer	Umur ibu	
umurKehamilan	integer	Umur kehamilan dalam minggu	
hamilKe	integer	Hamil keberapa	
beratBadan	integer	Berat badan	
tinggiBadan	integer	Tinggi badan	
imt	integer	Indeks masa tubuh	
golonganDarah	varchar(1)	Golongan darah	
tensi	varchar(20)	Tensi darah	
pendeteksiFaktorResiko	varchar(40)	Pendeteksi faktor resiko	
skorResiko	integer	Skor resiko	
jarakKehamilan	integer	Jarak kehamilan	
statusImunisasi	varchar(50)	Status imunisasi	
pemberianImunisasi	varchar(200)	Pemberian imunisasi	
dapatBukuKia	varchar(10)	Dapat Buku KIA	
penolongPeralinan	varchar(50)	Penolong persalinan	
waktuPeralinan	varchar(200)	Waktu persalinan	
kelahiran	varchar(50)	Waktu kelahiran	
nifas1	varchar(50)	Pelayanan nifas 1	
nifas2	varchar(50)	Pelayanan nifas 2	
nifas3	varchar(50)	Pelayanan nifas 3	
bulan	varchar(200)	Pelayanan	
keterangan	varchar(200)	Keterangan	
desa	varchar(20)	Desa	
K1	varchar(11)	Cakupan K1	
K4	varchar(11)	Cakupan K4	
deteksiResti	varchar(11)	Cakupan deteksi resiko tinggi	
restikom	varchar(11)	Resiko Komplikasi	
komKeb	varchar(11)	Komplikasi Keidanan	
pelayananNakes	varchar(11)	Pelayanan tenaga kesehatan	
pelayananFaskes	varchar(11)	Pelayanan fasilitas kesehatan	
pelayananNifas	varchar(11)	Pelayanan nifas	

2. Tabel tb_bayi

Tabel 5.3 Atribut data register bayi

<i>Field</i>	<i>Type</i>	<i>Deskripsi</i>	<i>Keterangan</i>
idBayi	integer	Nomor identitas dari register bayi	PK
idIbu	integer	Nomor identitas dari ibu	FK
namaBayi	varchar(50)	Nama bayi	
tanggalLahir	varchar(100)	Tanggal lahir bayi	
jenisKelamin	varchar(20)	Jenis kelamin	
beratBadanLahir	integer	berat badan lahir dalam gram	
saatLahir	varchar(100)	Kondisi saat lahir	
kunjunganNeonatal1	varchar(100)	Kunjungan neonatal 1	
kunjunganNeonatal2	varchar(100)	Kunjungan neonatal 2	
kunjunganNeonatal3	varchar(100)	Kunjungan neonatal 3	
bulan	varchar(200)	Pelayanan	
vitaminA	varchar(200)	Tanggal Vitamin A	
imunisasiHB	varchar(200)	Tanggal imunisasi HB	
imunisasiBCG	varchar(200)	Tanggal imunisasi BCG	
imunisasiPolio	varchar(200)	Tanggal imunisasi Polio	
imunisasiCampak	varchar(200)	Tanggal imunisasi Campak	
penyebabKematian	varchar(200)	Penyebab kematian	
keteranganBayi	varchar(200)	Keterangan	
KN1	varchar(11)	Cakupan KN1	
KNLengkap	varchar(11)	Cakupan KN Lengkap	
neonatalKomplikasi	varchar(11)	Cakupan neonatal Komplikasi	
bayiParipurna	varchar(11)	Cakupan bayi paripurna	
beratBadanTerakhir	integer	Cakupan	

3. Tabel tb_user

Tabel 5.4 Atribut data user

<i>Field</i>	<i>Type</i>	<i>Deskripsi</i>	<i>Keterangan</i>
idUser	integer	Nomor identitas user	PK
username	varchar(40)	Username	<i>Unique</i>
password	varchar(200)	Password	
nama	varchar(100)	Nama user	
alamat	varchar(200)	Alamat	
foto	varchar(200)	Link foto user	
jenisUser	varchar(20)	Jenis user	
desa	varchar(20)	Desa	

5.1.5 Perancangan Komponen

Perancangan komponen merupakan kegiatan mengambil setiap komponen dalam sistem tentang bagaimana komponen tersebut akan berjalan (Sommerville,2011).Model desain dapat digunakan untuk mempermudah proses implementasi oleh programmer. Perancangan komponen berisi algoritma-

algoritma yang akan diimplementasikan pada kelas-kelas yang didefinisikan. Algoritma yang dituliskan merupakan pseudocode yang menjembatani bahasa manusia dengan code program. Standart yang digunakan merujuk pada web users.csc.calpoly.ed yang ditulis oleh Dr. John Dalbey. Dalam perancangan komponen SIPKIA sampel yang diambil yaitu algoritma pada kelas BidanDesaKidal, kelas RegisterKohortIbu dan kelas E_Rekapan.

5.1.5.1 Perancangan Komponen Client

Dalam pola *factory method* terdapat kelas client untuk mengakses polanya. Pada SIPKIA kelas client adalah kelas C_Bidan. Cuplikan fungsi dari kelas C_Bidan pada method lihat Register dapat dilihat pada Gambar 5.9 berikut,

Fungsi Lihat Register
Fungsi lihatRegister(bidan,jenis) Jika parameter bidan sama dengan desa kidal Buat objek dari kelas BidanDesaKidal Panggil fungsi lihatRegister dengan parameter jenis menggunakan objek yang telah dibuat Panggil fungsi lihatRegister dengan parameter ibu menggunakan objek yang telah dibuat Masukkan hasil pemanggilan fungsi kedalam array Kembalikan nilai array tersebut Lakukan hal tersebut untuk semua desa

Gambar 5.9 Perancangan komponen client

5.1.5.2 Perancangan Komponen Concrete Creator

Pada kelas BidanDesaKidal terdapat fungsi-fungsi *factory method* sebagai kelas *concrete creator*. Algoritma fungsi tambah register kelas BidanDesaKida dapat dilihat pada Gambar 5.10 berikut,

Fungsi Tambah Register
Fungsi tambahRegister(jenis) kasus kunci adalah variabel jenis jika kasus adalah ibu buat objek dari kelas RegisterKohortIbu panggil kelas addregister dengan menggunakan objek yang telah dibuat dengan parameter desa keluar dari percabangan jika kasus adalah bayi buat objek dari kelas RegisterKohortBayi panggil kelas addregister dengan menggunakan objek yang telah dibuat dengan parameter desa keluar dari percabangan jika kasus adalah balita buat objek dari kelas RegisterKohortBalita panggil kelas addregister dengan menggunakan objek yang telah dibuat dengan parameter desa keluar dari percabangan



jika kasus adalah apras
 buat objek dari kelas RegisterKohortApras
 panggil kelas addregister dengan menggunakan objek yang telah
 dibuat dengan parameter desa
 keluar dari percangan
 jika kasus tidak sama dengan kasus yang didefinisikan
 tampilkan "salah parameter"
 keluar dari percabangan

Gambar 5.10 Perancangan komponen *concrete creator*

5.1.5.3 Perancangan Komponen *Concrete Product*

Pada kelas RegisterKohortIbu terdapat fungsi-fungsi sebagai kelas *concrete product*. Algoritma fungsi tambah register pada kelas RegisterKohortIbu dapat dilihat pada Gambar 5.11 berikut,

Fungsi Tambah Register

Fungsi tambahRegister(desa)

Variabel namaIbu sama dengan masukan dari form nama ibu
 Variabel namaAyah sama dengan masukan dari form nama ayah
 Variabel alamat sama dengan masukan dari form alamat
 Variabel umurIbu sama dengan masukan dari form umur ibu
 Variabel umurKehamilan sama dengan masukan dari form umur kehamilan
 Variabel hamilKe sama dengan masukan dari form hamil ke
 Variabel beratBadan sama dengan masukan dari form berat badan
 Variabel dapatBukuKia sama dengan masukan dari form dapat buku kia
 Variabel tinggiBadan sama dengan masukan dari form tinggi badan
 Variabel imt adalah hasil dari berat badan dibagi tinggi badan dibagi seratus dikali
 tinggi badan dibagi seratus
 Bulatkan nilai variabel imt
 Variabel golongan darah sama dengan masukan dari form golongan darah
 Variabel tensi sama dengan masukan dari form tensi
 Variabel pendeteksiFaktorResiko sama dengan masukan dari form pendeteksi faktor
 resiko
 Variabel skorResiko sama dengan masukan dari form skor resiko
 Variabel pemberianImuniasi sama dengan masukan dari form pemberian imunisasi
 Variabel statusImunisasi sama dengan masukan dari form status imunisasi
 Variabel jarakKehamilan sama dengan masukan dari form jarak kehamilan
 Variabel keterangan sama dengan masukan dari form jarak keterangan
 Variabel hamil sama dengan nilai pembulatan umurKehamilan dibagi 4
 Variabel perkiraan sama dengan nilai pembulatan dari (36-hamil)/4 digabung bulan
 sekarang
 Jika isi variabel pendeteksiFaktorResiko adalah masyarakat
 Variabel tanggal adalah waktu sekarang
 Update tabel tb_ibu ke database dimana idRegister sama dengan
 idRegisterMasukan dengan data baru pada kolom deteksiResti dengan nilai tanggal
 Masukkan semua data masukan dari form kedalam array dengan nama data
 Masukkan variabel array data ke tabel tb_ibu pada database

Gambar 5.11 Perancangan komponen *concrete product*

5.1.5.4 Perancangan Komponen Rekapam Register

Pada kelas E_Rekapam terdapat fungsi-fungsi sebagai kelas untuk merekap data register kohort. Algoritma fungsi newRekapam pada kelas E_Rekapam dapat dilihat pada Gambar 5.12 berikut,

Fungsi Perbarui Rekapam
<p>Fungsi newRekapam(desas,tahun) ambil data pada tb_ibu,tb_bayi,tb_balita dan tb_apras dimana desa sama dengan parameter desa pada masing-masing cakupan cari berapa jumlah cakupan pada bulan ini dan parameter tahun pada masing-masing cakupan edit cakupan dengan jumlah cakupan yang terekap pada bulan ini dan parameter tahun</p>

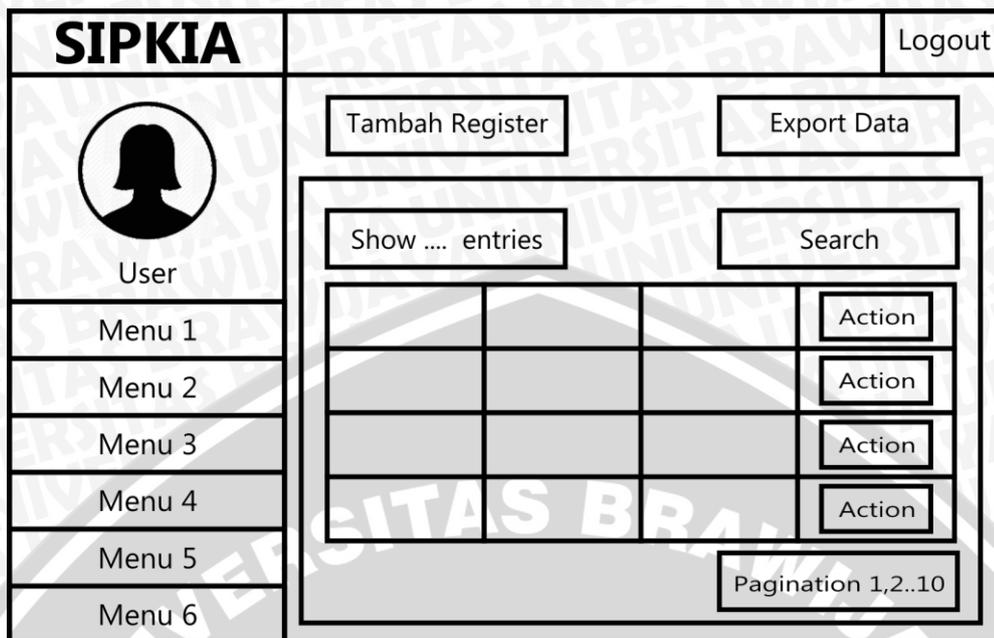
Gambar 5.12 Perancangan komponen rekapam register

5.1.6 Perancangan Antarmuka Pengguna

Proses ini bertujuan untuk menghasilkan antarmuka sistem agar *output* yang dihasilkan sistem dapat diterima dengan baik oleh pengguna. *Output* yang dihasilkan diantaranya adalah register kohort ibu, register kohort bayi, register kohort balita, register kohort apras dan berupa rekapam data dan pemetaan pada puskesmas kecamatan.

5.1.6.1 Antarmuka Halaman Lihat Register

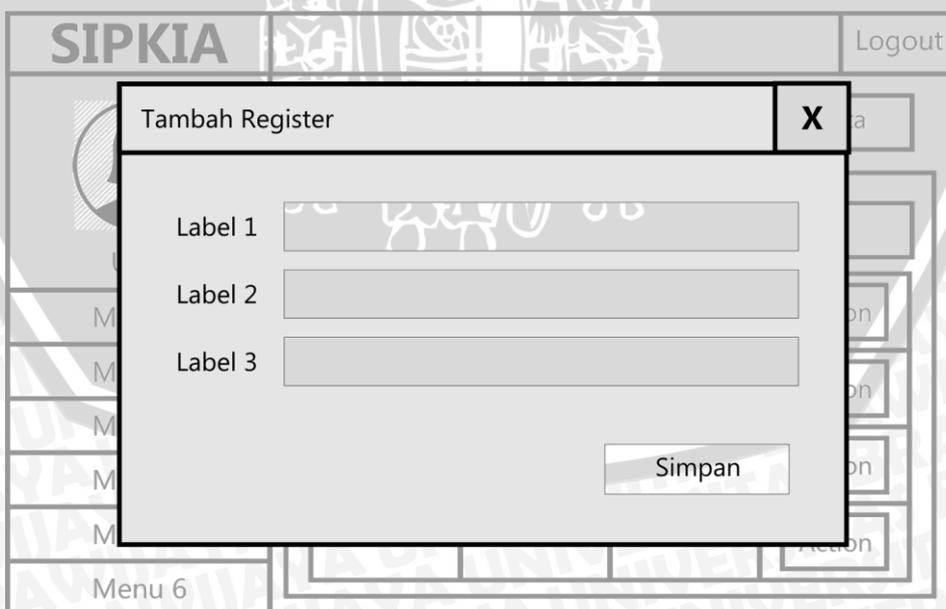
Halaman ini merupakan halaman untuk melihat data-data yang telah dimasukkan pada register kohort ibu, register kohort bayi, register kohort balita, dan register kohort apras. Keempat register tersebut memiliki tampilan halaman yang hampir sama. Pada halaman ini akan menampilkan data-data register yang telah masuk dalam database. Tombol-tombol yang ada dalam hamalan ini adalah tombol menu navigasi, tambah register, export data, logout dan aksi-aksi yang dapat diberikan pada tiap baris tabel. Aksi tersebut berupa lihat detail register, ubah register, hapus register, lihat pelayanan dan tambah pelayanan. Selain itu juga ada form seacrh untuk mencari data register dan juga ada tombol pagination untuk melihat pada tabel selanjutnya. Rancangan antarmuka halaman lihat register dapat dilihat pada Gambar 5.13 berikut,



Gambar 5.13 Antarmuka halaman lihat register

5.1.6.2 Antarmuka Tambah Register

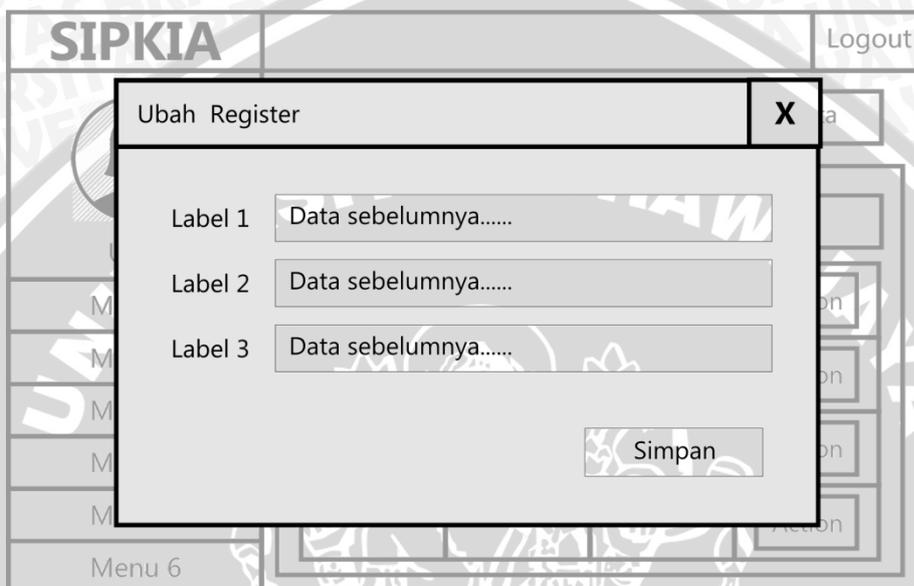
Halaman ini merupakan halaman atau form untuk menambahkan atau membuat pada register kohort ibu, register kohort bayi, register kohort balita, dan register kohort apras. Keempat form penambahan register tersebut adalah modals atau pop up yang memiliki tampilan yang hampir sama. Dalam modals terdiri dari form yang harus diisi oleh bidan desa berupa data register seperti nama ibu, alamat, umur kehamilan dan lain-lain. Rancangan antarmuka halaman tambah register kohort dapat dilihat pada Gambar 5.14 berikut,



Gambar 5.14 Antarmuka tambah register kohort

5.1.6.3 Antarmuka Ubah Register

Halaman ini merupakan halaman atau form untuk mengubah register kohort ibu, register kohort bayi, register kohort balita, dan register kohort apras. Keempat form perubahan register tersebut adalah modals atau pop up yang memiliki tampilan yang hampir sama. Pada modals ubah register terdiri dari form yang dapat dirubah isinya oleh bidan desa seperti form nama bayi, alamat, keterangan dan lain-lain. Rancangan antarmuka halaman ubah register kohort dapat dilihat pada Gambar 5.15 berikut,



The image shows a screenshot of a web application interface. At the top left, there is a header with the text 'SIPKIA'. In the top right corner, there is a 'Logout' button. A modal window titled 'Ubah Register' is open in the center. The modal has a close button 'X' in the top right corner. Inside the modal, there are three input fields, each preceded by a label: 'Label 1', 'Label 2', and 'Label 3'. Each input field contains the text 'Data sebelumnya.....'. At the bottom right of the modal, there is a button labeled 'Simpan'. The background of the application is slightly blurred, showing a sidebar with menu items and a main content area.

Gambar 5.15 Antarmuka ubah register

5.1.6.4 Antarmuka Hapus Register

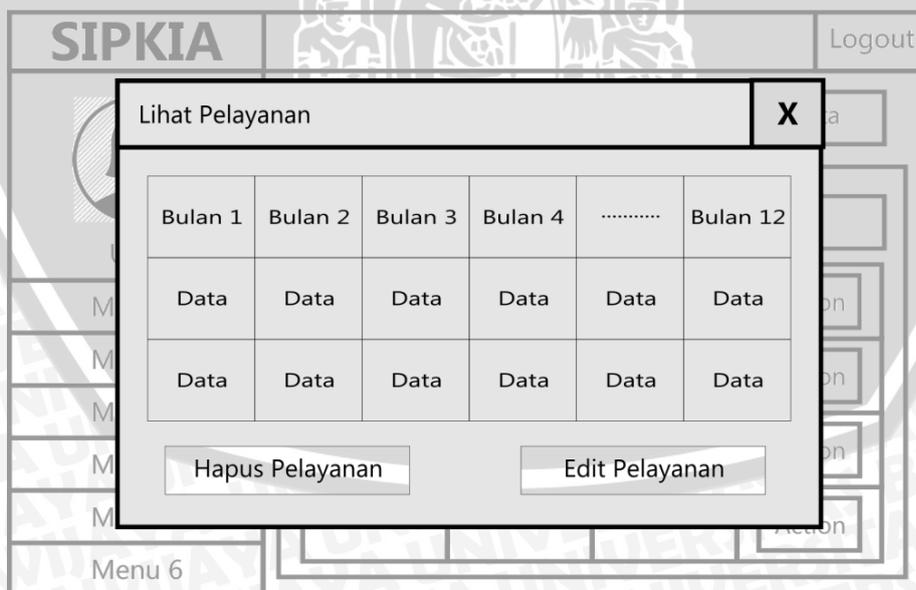
Halaman ini merupakan halaman untuk menghapus register kohort ibu, register kohort bayi, register kohort balita, dan register kohort apras. Keempat halaman hapus register tersebut adalah modals atau pop up yang memiliki tampilan yang hampir sama. Modals dari hapus register ini merupakan modals untuk konfirmasi untuk penghapusan data register yang diinginkan oleh bidan desa. Rancangan antarmuka halaman hapus register kohort dapat dilihat pada Gambar 5.16 berikut,



Gambar 5.16 Antarmuka hapus register

5.1.6.5 Antarmuka Halaman Lihat Pelayanan

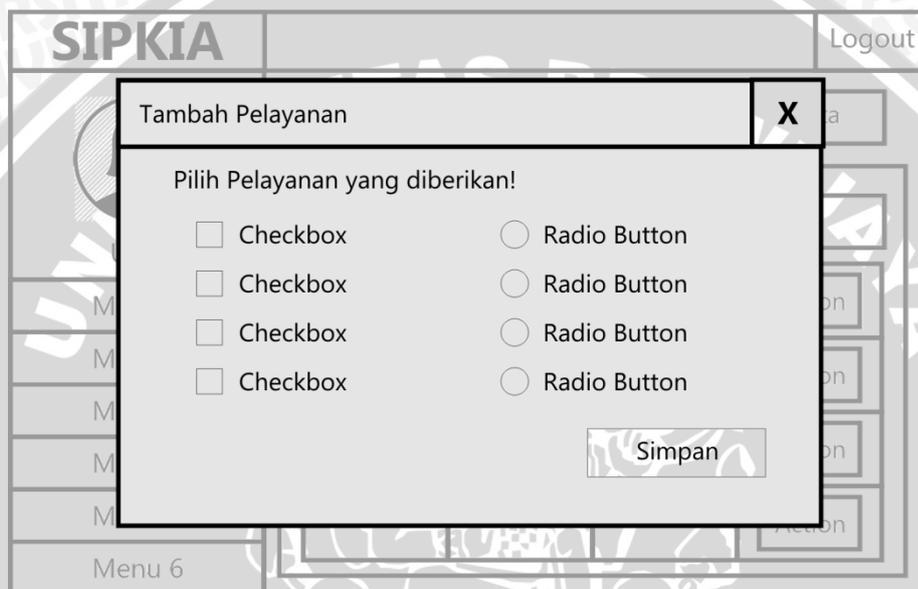
Halaman ini merupakan halaman untuk melihat pelayanan-pelayanan yang telah dimasukkan pada register kohort ibu, register kohort bayi, register kohort balita, dan register kohort apras. Keempat pelayanan tersebut memiliki tampilan halaman yang hampir sama. Pada halaman lihat pelayanan sistem akan menampilkan data pelayanan yang telah diberikan. Terdapat dua tab yang terdiri dari tab tahun. Selain itu juga ada tombol edit pelayanan dan hapus pelayanan untuk mengelola data pelayanan. Rancangan antarmuka halaman lihat pelayanan dapat dilihat pada Gambar 5.17 berikut,



Gambar 5.17 Halaman lihat pelayanan

5.1.6.6 Antarmuka Tambah Pelayanan

Halaman ini merupakan halaman atau form untuk menambahkan pelayanan pada register kohort ibu, register kohort bayi, register kohort balita, dan register kohort apras. Keempat form penambahan pelayanan tersebut adalah modals atau pop up yang memiliki tampilan yang hampir sama. Pada modals tambah pelayanan terdiri dari form untuk menambahkan pelayanan yang diberikan. Form tersebut lebih banyak berupa checkbox dan radio button. Bidan desa hanya perlu mengklik form tersebut ketika memberikan pelayanan seperti pelayanan vitamin A, imunisasi, dan lain-lain. Rancangan antarmuka halaman tambah pelayanan dapat dilihat pada Gambar 5.18 berikut,



Gambar 5.18 Antarmuka tambah pelayanan

5.2 Implementasi

Pada sub bab ini membahas implementasi berdasarkan hasil analisis kebutuhan dan proses perancangan SIPKIA. Pembahasan pada sub bab ini terdiri dari penjelasan tentang spesifikasi sistem, batasan implementasi, implementasi program, dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Hasil analisis kebutuhan dan perancangan yang telah diuraikan pada sub bab perancangan menjadi acuan untuk melakukan implementasi sistem yang diharapkan akan berfungsi sesuai dengan kebutuhan pengguna. Spesifikasi sistem terdiri dari spesifikasi perangkat keras dan perangkat lunak sebagai berikut,

5.2.1.1 Spesifikasi Perangkat Keras

Pembangunan SIPKIA menggunakan spesifikasi perangkat keras dapat dilihat pada Tabel 5.5 berikut,

Tabel 5.5 Spesifikasi perangkat keras

Nama Komponen	Spesifikasi
Prosesor	Intel (R) Pentium
Memori (RAM)	2 GB
Harddisk	500 GB
Kartu Grafis	Intel (R) HD Graphics
Monitor	14.0", Resolusi 1366 x 768

5.2.1.2 Spesifikasi Perangkat Lunak

Pembangunan SIPKIA menggunakan spesifikasi perangkat lunak dapat dilihat pada Tabel 5. 6 berikut,

Tabel 5.6 Spesifikasi perangkat lunak

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 7 Ultimate
Browser	Google Chrome versi 42.0.2311.152, Firefox 33.1.1
Code Editor	Sublime Text 2.0.2, Adobe Dreamweaver CS5
Image Editor	Corel Draw 12
Basis Data	MySQL 5.6
Pemodelan, UML	Ideas Modeler 6.52, Enterprise Architect 8.0
Framework	Code Igniter 2.1.4, Bootstrap
Dokumentasi	MS Word 2010

5.2.2 Batasan Implementasi

Beberapa batasan implementasi Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method* diantaranya :

1. SIPKIA dirancang dan dibangun menggunakan bahasa pemrograman PHP, *framework* Code Igniter dan Bootstrap untuk pembuatan website, serta menggunakan MySQL sebagai pendukung basis data.
2. SIPKIA memiliki fitur utama berupa kelola data register kohort, rekapan data register kohort dan pemetaan kesehatan ibu dan anak. Register kohort tersebut terdiri dari register kohort ibu, bayi, balita dan apras.
3. Data yang digunakan dalam SIPKIA diambil dari Puskesmas Tumpang Bagian Kesehatan Ibu dan Anak (BKIA). Selain itu data juga diambil langsung bidan desa setempat.

5.2.3 Implementasi Program

Proses implementasi SIPKIA dilakukan dengan menggunakan bahasa pemrograman PHP dengan *framework* Code Igniter yang menggunakan konsep MVC dan *framework* bootstrap sebagai template CSS dan javascript. Pada

perancangan sebelumnya telah didefinisikan kelas-kelas yang masuk dalam pola perancangan *factory method*. Komponen yang terlibat dalam *factory method* adalah *creator* dan *product* serta client yang akan memanggil *factory method* tersebut. Berikut implementasi program dari masing-masing komponen *factory method*,

5.2.3.1 Implementasi Program Komponen Client

Dalam pola *factory method* dibutuhkan client untuk mengakses kelas yang ada dalam pola. Pada SIPKIA kelas client yang dibuat bernama kelas C_Bidan. Cuplikan code program yang diambil yaitu pada fungsi lihat register. Lihat register kohort merupakan kegiatan untuk melihat data registrasi kohort yang telah dimasukkan ke database. Pada kelas C_Bidan akan memanggil fungsi *factory method* pada kelas BidanDesa yang dipilih. Client tidak mengetahui kelas sebenarnya yang dibutuhkan untuk melihat register. Client hanya memanggil fungsi lihatRegister dengan malampirkan parameter bidan yaitu bidan desa mana yang memanggil dan parameter jenis yaitu kohort apa yang diinginkan oleh client. Implementasi program pada komponen client dapat dilihat pada Gambar 5.19 berikut,

```
public function lihatRegister($bidan,$jenis)
{
    if ($bidan=="kidal") {
        $factoryKidal = new BidanDesaKidal();
        $hasil['data']=$factoryKidal->lihatRegister($jenis);
        $hasil['ibu']=$factoryKidal->lihatRegister('ibu');
    }else if ($bidan=="kambingan") {
        $factoryKambingan = new BidanDesaKambingan();
        $hasil['data']=$factoryKambingan->lihatRegister($jenis);
        $hasil['ibu']=$factoryKambingan->lihatRegister('ibu');
    }else if ($bidan=="ngingit") {
        $factoryNgingit = new BidanDesaNgingit();
        $hasil['data']=$factoryNgingit->lihatRegister($jenis);
        $hasil['ibu']=$factoryNgingit->lihatRegister('ibu');
    }else if ($bidan=="pandanajeng") {
        $factoryPandanajeng = new BidanDesaPandanajeng();
        $hasil['data']=$factoryPandanajeng->lihatRegister($jenis);
        $hasil['data']=$factoryPandanajeng->lihatRegister($jenis);
    }else if ($bidan=="pulungdowo") {
        $factoryPulungdowo = new BidanDesaPulungdowo();
        $hasil['data']=$factoryPulungdowo->lihatRegister($jenis);
        $hasil['data']=$factoryPulungdowo->lihatRegister($jenis);
    }else if ($bidan=="bokor") {
        $factoryBokor = new BidanDesaBokor();
        $hasil['data']=$factoryBokor->lihatRegister($jenis);
        $hasil['data']=$factoryBokor->lihatRegister($jenis);
    }else if ($bidan=="slamet") {
        $factorySlamet = new BidanDesaSlamet();
        $hasil['data']=$factorySlamet->lihatRegister($jenis);
        $hasil['data']=$factorySlamet->lihatRegister($jenis);
    }else if ($bidan=="tumpang") {
        $factoryTumpang = new BidanDesaTumpang();
        $hasil['data']=$factoryTumpang->lihatRegister($jenis);
        $hasil['data']=$factoryTumpang->lihatRegister($jenis);
    }
}
```

```

}else if ($bidan=="dampul") {
    $factoryDampul = new BidanDesaDampul();
    $hasil['data']=$factoryDampul->lihatRegister($jenis);
    $hasil['data']=$factoryDampul->lihatRegister($jenis);
}else if ($bidan=="duwet") {
    $factoryDuwet = new BidanDesaDuwet();
    $hasil['data']=$factoryDuwet->lihatRegister($jenis);
    $hasil['data']=$factoryDuwet->lihatRegister($jenis);
}else if ($bidan=="benjor") {
    $factoryBenjor = new BidanDesaBenjor();
    $hasil['data']=$factoryBenjor->lihatRegister($jenis);
    $hasil['data']=$factoryBenjor->lihatRegister($jenis);
}else if ($bidan=="malangsuko") {
    $factoryMalangsuko = new BidanDesaMalangsuko();
    $hasil['data']=$factoryMalangsuko->lihatRegister($jenis);
    $hasil['data']=$factoryMalangsuko->lihatRegister($jenis);
}else if ($bidan=="wringinsongo") {
    $factoryWringinsongo = new BidanDesaWringinsongo();
    $hasil['data']=$factoryWringinsongo->lihatRegister($jenis);
    $hasil['data']=$factoryWringinsongo->lihatRegister($jenis);
}else if ($bidan=="jeru") {
    $factoryJeru = new BidanDesaJeru();
    $hasil['data']=$factoryJeru->lihatRegister($jenis);
    $hasil['data']=$factoryJeru->lihatRegister($jenis);
}else if ($bidan=="tulusbesar") {
    $factoryTulusbesar = new BidanDesaTulusBesar();
    $hasil['data']=$factoryTulusbesar->lihatRegister($jenis);
    $hasil['data']=$factoryTulusbesar->lihatRegister($jenis);
}
$view="";
if ($jenis=="ibu") {
    $view="B_KohortIbu";
}else if ($jenis=="bayi") {
    $view="B_KohortBayi";
}else if ($jenis=="balita") {
    $view="B_KohortBalita";
}else if ($jenis=="apras") {
    $view="B_KohortApras";
}
$this->load->view($view, $hasil);
}

```

Gambar 5.19 Kode program komponen client

5.2.3.2 Implementasi Program Komponen *Concrete Creator*

Concrete creator dalam SIPKIA merupakan bidan desa pada masing-masing desa di Kecamatan Tumpang. *Concrete creator* merupakan method yang akan mengarahkan ke kelas yang diminta oleh client. Pada kasus ini cuplikan program yang diambil adalah fungsi tambah register pada bidan desa kidal. Tambah register kohort merupakan kegiatan untuk menambah atau memasukkan data registrasi kohort baru yang dapat dilakukan oleh bidan desa. Pada fungsi ini memiliki parameter jenis yang merupakan jenis kohort yang diinginkan oleh client. Method ini akan meneruskan fungsi (create) pada komponen *concrete product* yaitu register kohort ibu, bayi, balita atau apras. Implementasi program pada komponen *concrete creator* dapat dilihat pada Gambar 5.20 berikut,

```

function tambahRegister($jenis) {
    switch ($jenis) {
        case "ibu":
            $register = new RegisterKohortIbu;
            $hasil=$register->addRegister("kidal");
            break;
        case "bayi":
            $register = new RegisterKohortBayi;
            $register->addRegister("kidal");
            break;
        case "balita":
            $register = new RegisterKohortBalita;
            $register->addRegister("kidal");
            break;
        case "apras":
            $register = new RegisterKohortApras;
            $register->addRegister("kidal");
            break;
        default:
            echo "Salah Parameter";
            break;
    }
}

```

Gambar 5.20 Kode program komponen *concrete creator*

5.2.3.3 Implementasi Program Komponen *Concrete Product*

Concrete product dalam SIPKIA merupakan register kohort ibu, bayi, balita dan apras. *Concrete product* merupakan kelas yang melayani permintaan dari *concrete creator* yang secara tidak langsung dipanggil oleh client. Pada kasus ini cuplikan program yang diambil adalah fungsi tambah register pada register kohort ibu. Dalam implementasinya *concrete product* merupakan kelas model yang memiliki hubungan transaksi data dengan database. Fungsi add register memiliki parameter desa dimana ibu atau anak pada desa mana yang akan ditambahkan oleh bidan desa. Pada akhir fungsi program akan memasukkan data register baru ke tabel tb_ibu pada database. Implementasi program pada komponen *concrete product* dapat dilihat pada Gambar 5.21 berikut,

```

function addRegister($desa) {
    $namaIbu=$this->input->post('namaIbu');
    $namaAyah=$this->input->post('namaAyah');
    $alamat=$this->input->post('alamat');
    $umurIbu=$this->input->post('umurIbu');
    $umurKehamilan=$this->input->post('umurKehamilan');
    $hamilKe=$this->input->post('hamilKe');
    $beratBadan=$this->input->post('beratBadan');
    $dapatBukuKia=$this->input->post('dapatBukuKia');
    $tinggiBadan=$this->input->post('tinggiBadan');
    $imt=$beratBadan/((($tinggiBadan/100)*($tinggiBadan/100)));
    $imt= floor($imt);
    $golonganDarah=$this->input->post('golonganDarah');
    $tensi=$this->input->post('tensi');
    $pendeteksiFaktorResiko=$this->input->post('pendeteksiFaktorResiko');
    $skorResiko=$this->input->post('skorResiko');
}

```

```

$ pemberianImunisasi=$this->input->post('pemberianImunisasi');
$statusImunisasi=$this->input->post('statusImunisasi');
$jarakKehamilan=$this->input->post('jarakKehamilan');
$keterangan=$this->input->post('keterangan');

$shamil=floor($umurKehamilan/4);
$perkiraan=floor((36-$shamil)/4)+date('m');
$perkiraan="bulan".$perkiraan;

if ($pendeteksiFaktorResiko=="Masyarakat") {
    $bulan=date('d-m-Y');
    $data = array(
        'deteksiResti' => $bulan
    );
    $this->db->where('idRegister', $idRegister);
    $this->db->update('tb_ibu', $data);
}
if ($pemberianImunisasi=="Diberi imunisasi TT") {
    $tanggal=date('d-m-Y');
    $pemberianImunisasi=$pemberianImunisasi." pada ".$tanggal;
}

$data = array(
    'namaIbu' => $namaIbu,
    'namaAyah' => $namaAyah,
    'alamat' => $alamat,
    'desa' => $desa,
    'umurIbu' => $umurIbu,
    'umurKehamilan' => $umurKehamilan,
    'hamilKe' => $hamilKe,
    'beratBadan' => $beratBadan,
    'tinggiBadan' => $tinggiBadan,
    'imt' => $imt,
    'golonganDarah' => $golonganDarah,
    'tensi' => $tensi,
    'pendeteksiFaktorResiko' => $pendeteksiFaktorResiko,
    'skorResiko' => $skorResiko,
    'pemberianImunisasi' => $pemberianImunisasi,
    'statusImunisasi' => $statusImunisasi,
    'skorResiko' => $skorResiko,
    'jarakKehamilan' => $jarakKehamilan,
    'dapatBukuKia' => $dapatBukuKia,
    'keterangan' => $keterangan,
    $perkiraan => "Perkiraan kelahiran"
);
$this->db->insert('tb_ibu', $data);
}

```

Gambar 5.21 Kode program komponen *concrete product*

5.2.4 Implementasi Antarmuka

Antarmuka sistem merupakan sarana yang digunakan pengguna untuk berinteraksi dengan sistem. Berikut penjelasan mengenai implementasi antarmuka yang telah dibuat pada SIPKIA,

5.2.4.1 Implementasi Antarmuka Lihat Register

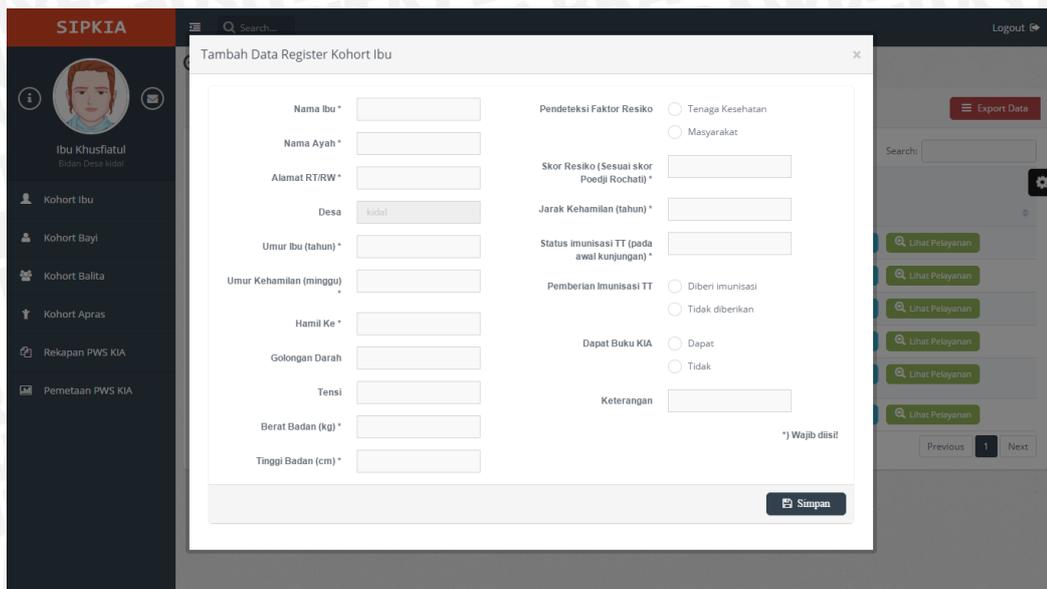
Antarmuka lihat register merupakan halaman utama dari masing-masing register kohort yaitu register kohort ibu, bayi, balita dan anak. Halaman ini berisi data register kohort yang telah terdaftar, berisi tombol aksi dari data, dan pagination. Implementasi antarmuka lihat register dapat dilihat pada Gambar 5.22 berikut,

ID Register	Nama Ibu	Nama Suami	Alamat (RTRW)	Umur Ibu (tahun)	Umur Kehamilan (minggu)	Hamil Ke	IMT	Jarak Kehamilan	Kelola	Pelayanan
1	Ulina	Samsul	21/1	42	10	4	24	4	[Icons]	+ Tambah Pelayanan Lihat Pelayanan
2	Mariana	Slamet	36/2	33	20	2	24	8	[Icons]	+ Tambah Pelayanan Lihat Pelayanan
3	Markonah	Hari	36/2	23	20	2	24	5	[Icons]	+ Tambah Pelayanan Lihat Pelayanan
4	Sullati	Sullanto	12/3	33	7	3	29	4	[Icons]	+ Tambah Pelayanan Lihat Pelayanan
7	Sri Wahyuningsih	Siptoni	01/01	35	3	2	32	4	[Icons]	+ Tambah Pelayanan Lihat Pelayanan
10	Khoirul'umah	Khunaini	07/12	37	4	3	31	4	[Icons]	+ Tambah Pelayanan Lihat Pelayanan

Gambar 5.22 Antarmuka halaman lihat register

5.2.4.2 Implementasi Antarmuka Tambah Register

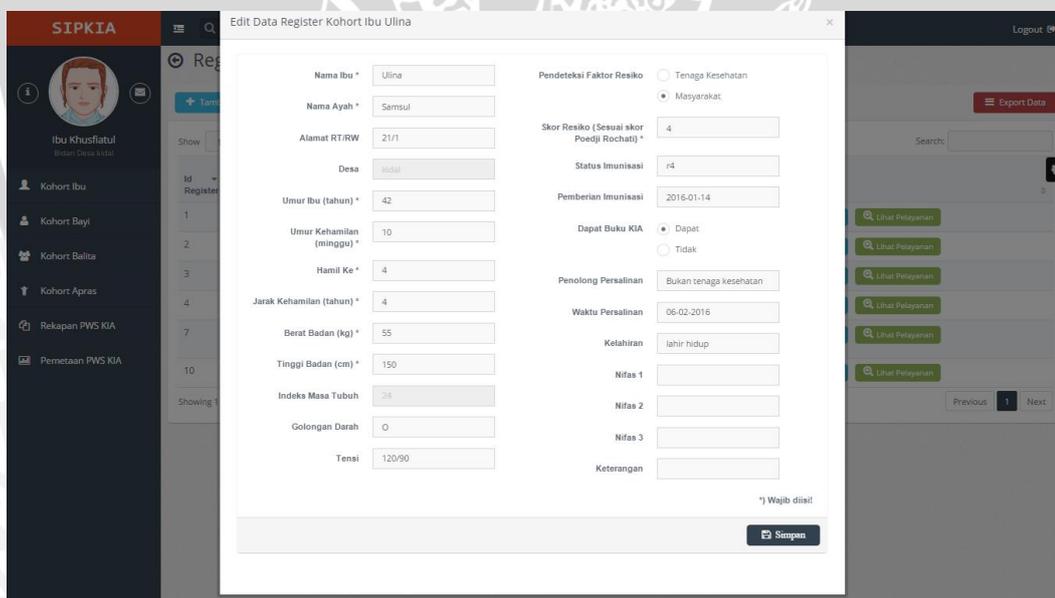
Antarmuka tambah register merupakan halaman untuk menambah data register baru yaitu register kohort ibu, bayi, balita dan anak. Halaman ini berupa pop up atau modals dimana user disuruh untuk mengisi form penambahan register baru. Implementasi antarmuka tambah register dapat dilihat pada Gambar 5.23 berikut,



Gambar 5.23 Antarmuka halaman tambah register

5.2.4.3 Implementasi Antarmuka Ubah Register

Antarmuka ubah register merupakan halaman untuk mengubah data register yang telah terdaftar yaitu register kohort ibu, bayi, balita dan anak. Halaman ini berupa pop up atau modals dimana user disuruh untuk mengisi form perubahan data dari data terakhir. Implementasi antarmuka ubah register dapat dilihat pada Gambar 5.24 berikut,

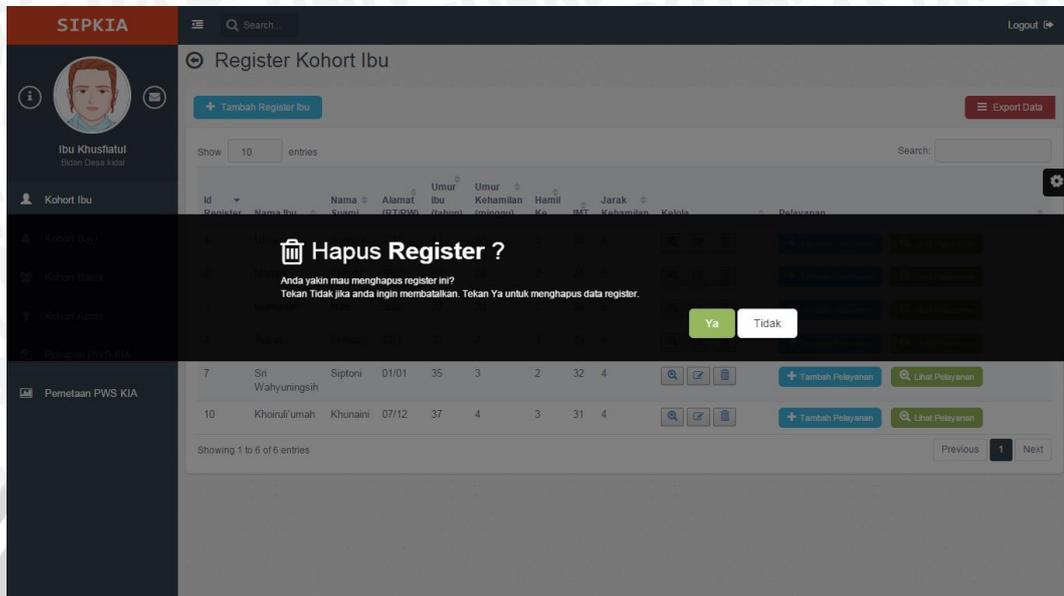


Gambar 5.24 Antarmuka halaman ubah register

5.2.4.4 Implementasi Antarmuka Hapus Register

Antarmuka hapus register merupakan halaman untuk menghapus data register yang telah terdaftar yaitu register kohort ibu, bayi, balita dan anak.

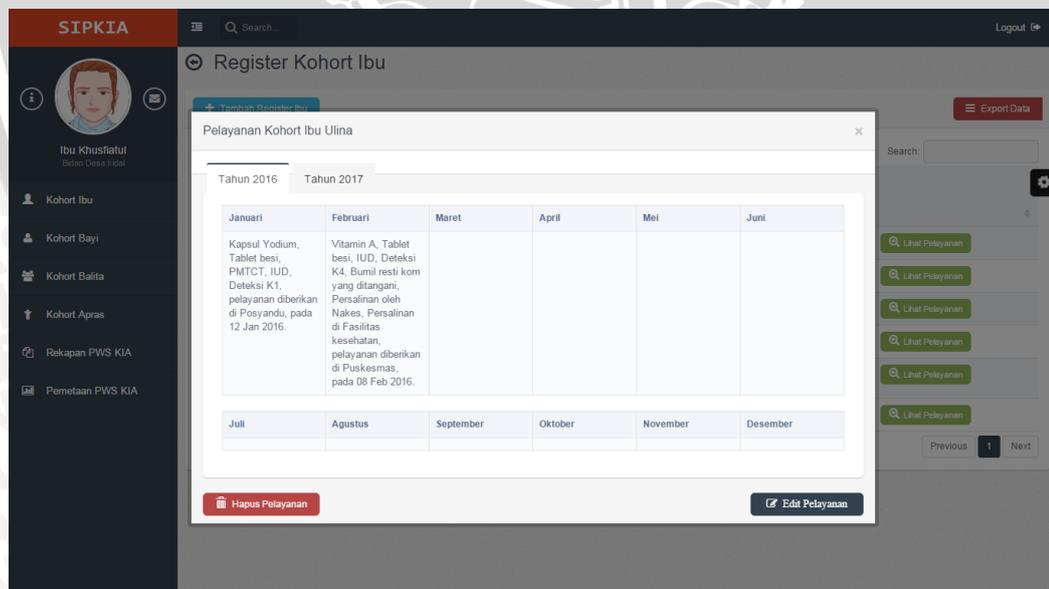
Halaman ini berupa pop up atau modals dimana user disuruh mengkonfirmasi hapus register. Implementasi antarmuka hapus register dapat dilihat pada Gambar 5.25 berikut,



Gambar 5.25 Antarmuka halaman hapus register

5.2.4.5 Implementasi Antarmuka Lihat Pelayanan

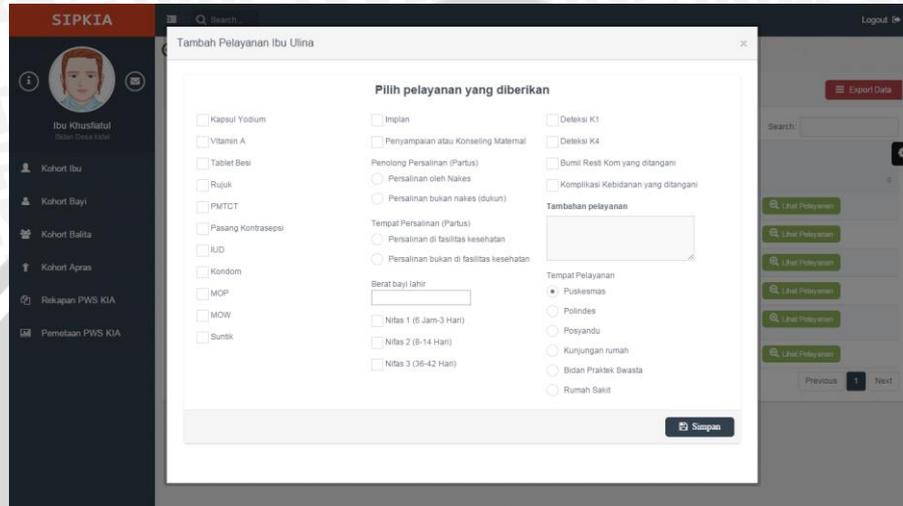
Antarmuka lihat pelayanan merupakan halaman pelayanan yang telah diberikan pada salah satu register. Implementasi antarmuka lihat pelayanan dapat dilihat pada Gambar 5.26 berikut,



Gambar 5.26 Antarmuka halaman lihat pelayanan

5.2.4.6 Implementasi Antarmuka Tambah Pelayanan

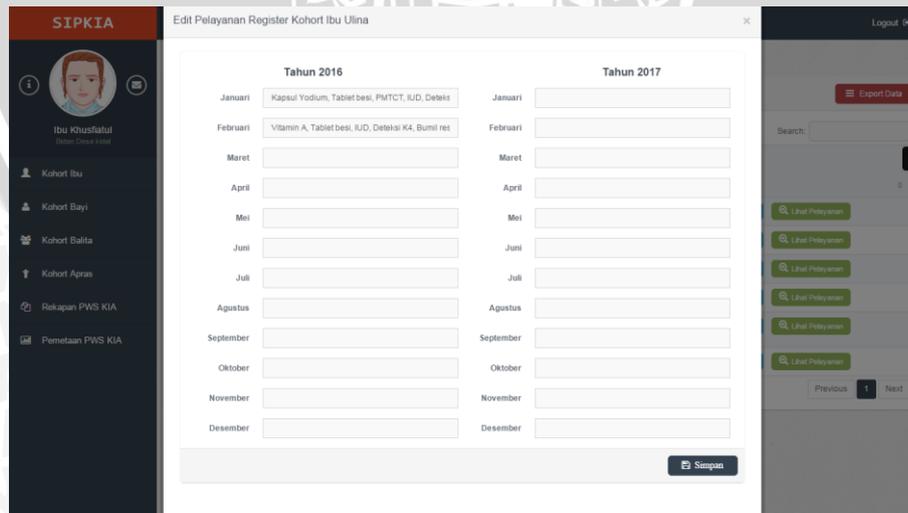
Antarmuka tambah pelayanan merupakan halaman untuk menambah data pelayanan pada register kohort ibu, bayi, balita dan anak. Halaman ini berupa pop up atau modals dimana user disuruh untuk mengisi form tambah pelayanan. Implementasi antarmuka tambah pelayanan dapat dilihat pada Gambar 5.27 berikut,



Gambar 5.27 Antarmuka halaman tambah pelayanan

5.2.4.7 Implementasi Antarmuka Ubah Pelayanan

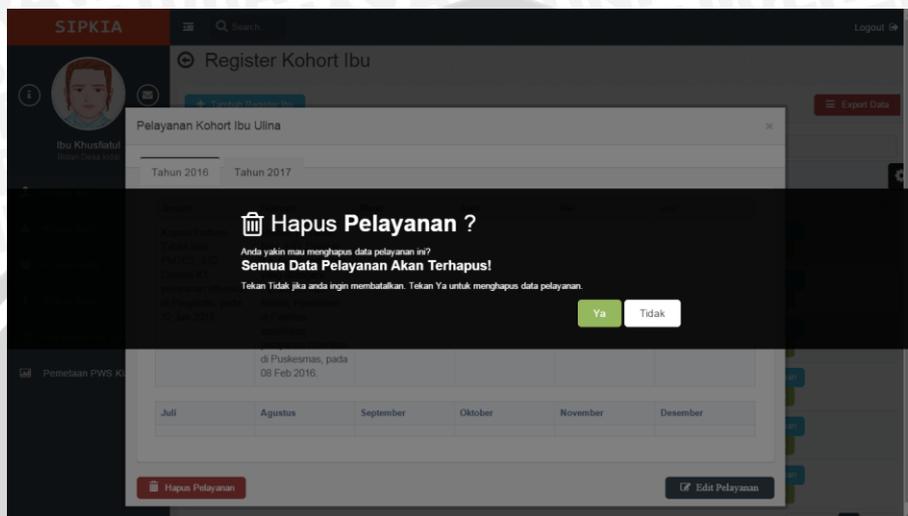
Antarmuka ubah pelayanan merupakan halaman untuk mengubah data pelayanan pada salah satu register kohort ibu, bayi, balita dan anak. Halaman ini berupa pop up atau modals dimana user disuruh untuk mengisi form perubahan data dari data terakhir. Implementasi antarmuka ubah pelayanan dapat dilihat pada Gambar 5.28 berikut,



Gambar 5.28 Antarmuka halaman ubah pelayanan

5.2.4.8 Implementasi Antarmuka Hapus Pelayanan

Antarmuka hapus pelayanan merupakan halaman untuk menghapus data pelayanan pada register kohort ibu, bayi, balita dan anak. Halaman ini berupa pop up atau modals dimana user disuruh mengkonfirmasi hapus pelayanan. Implementasi antarmuka hapus pelayanan dapat dilihat pada Gambar 5.29 berikut,



Gambar 5.29 Antarmuka halaman hapus pelayanan

5.2.4.9 Implementasi Antarmuka Lihat Rekap Desa

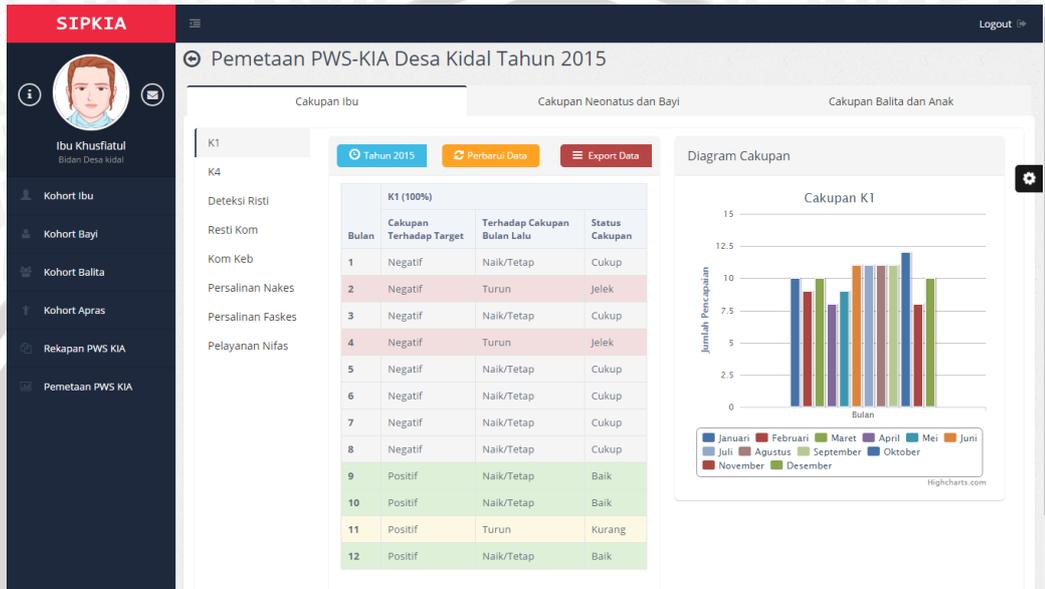
Antarmuka lihat rekap desa merupakan halaman yang menampilkan data rekap pada sebuah desa. Rekap dari 4 macam kohort diimplementasikan pada 3 tab yang terdiri dari cakupan ibu, cakupan bayi dan cakupan balita apras. Rekap yang ditampilkan berdasarkan tahun yang dipilih oleh bidan desa atau pegawai puskesmas. Implementasi antarmuka lihat rekap desa dapat dilihat pada Gambar 5.30 berikut,



Gambar 5.30 Antarmuka lihat rekap desa

5.2.4.10 Implementasi Antarmuka Lihat Pemetaan Desa

Antarmuka lihat pemetaan desa merupakan halaman yang menampilkan pemetaan dari cakupan-cakupan pada sebuah desa. Pemetaan dari 4 macam kohort diimplementasikan pada 3 tab yang terdiri dari cakupan ibu, cakupan bayi dan cakupan balita apras. Pemetaan yang ditampilkan berdasarkan tahun yang dipilih oleh bidan desa atau pegawai puskesmas. Implementasi antarmuka lihat pemetaan desa dapat dilihat pada Gambar 5.31 berikut,



Gambar 5.31 Antarmuka lihat pemetaan desa

5.2.4.11 Implementasi Antarmuka Lihat Rekapitulasi Kecamatan

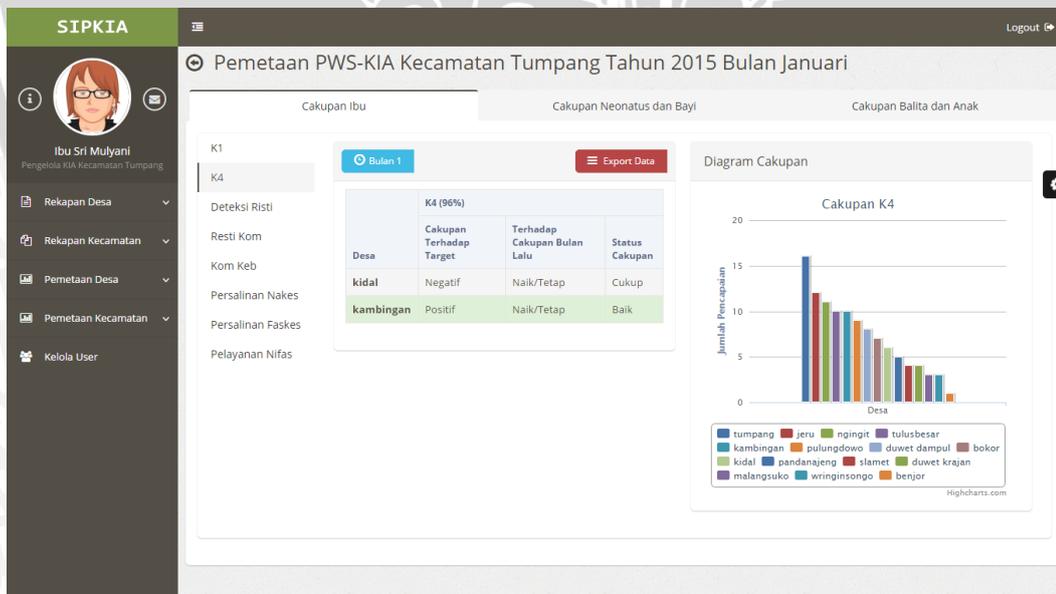
Antarmuka lihat rekapitulasi kecamatan merupakan halaman yang menampilkan data rekapitulasi sekecamatan. Rekapitulasi dari 4 macam kohort diimplementasikan pada 3 tab yang terdiri dari cakupan ibu, cakupan bayi dan cakupan balita apras. Rekapitulasi yang ditampilkan berdasarkan tahun dan bulan yang dipilih oleh bidan desa atau pegawai puskesmas. Pada halaman ini menampilkan data cakupan yang dicapai pada masing-masing desa. Implementasi antarmuka lihat rekapitulasi desa dapat dilihat pada Gambar 5.32 berikut,

Desa	K1				K4				Deteksi Resti				Bumil Resti			
	Lalu	Sekarang	Kom	%	Lalu	Sekarang	Kom	%	Lalu	Sekarang	Kom	%	Lalu	Sekarang	Kom	%
Tumpang	0	22	22	9%	0	16	16	6%	0	16	16	32%	0	5	5	2%
Malangsono	0	5	5	9%	0	3	3	5%	0	3	3	27%	0	1	1	2%
Jeru	0	9	9	8%	0	12	12	11%	0	12	12	53%	0	3	3	3%
Wringinsongo	0	4	4	8%	0	3	3	6%	0	3	3	29%	0	0	0	0%
Bokor	0	6	6	11%	0	7	7	12%	0	7	7	61%	0	0	0	0%
Slamet	0	7	7	9%	0	4	4	5%	0	4	4	26%	0	1	1	1%
Kidal	0	10	10	8%	0	6	6	5%	0	6	6	25%	0	3	3	3%
Kambangan	0	5	5	7%	0	10	10	15%	0	10	10	74%	0	2	2	3%
Ngingit	0	8	8	10%	0	11	11	13%	0	11	11	66%	0	1	1	1%
Pandanajeng	0	5	5	7%	0	5	5	7%	0	5	5	35%	0	1	1	1%
Pulungdowo	0	9	9	6%	0	9	9	6%	0	9	9	30%	0	4	4	3%
Tulusbesar	0	10	10	11%	0	10	10	11%	0	5	5	26%	0	1	1	1%
Benjor	0	1	1	3%	0	1	1	3%	0	2	2	26%	0	1	1	3%
Duwet Dampul	0	0	0	13%	0	0	0	13%	0	1	1	8%	0	0	0	2%

Gambar 5.32 Antarmuka lihat rekapan kecamatan

5.2.4.12 Implementasi Antarmuka Lihat Pemetaan Kecamatan

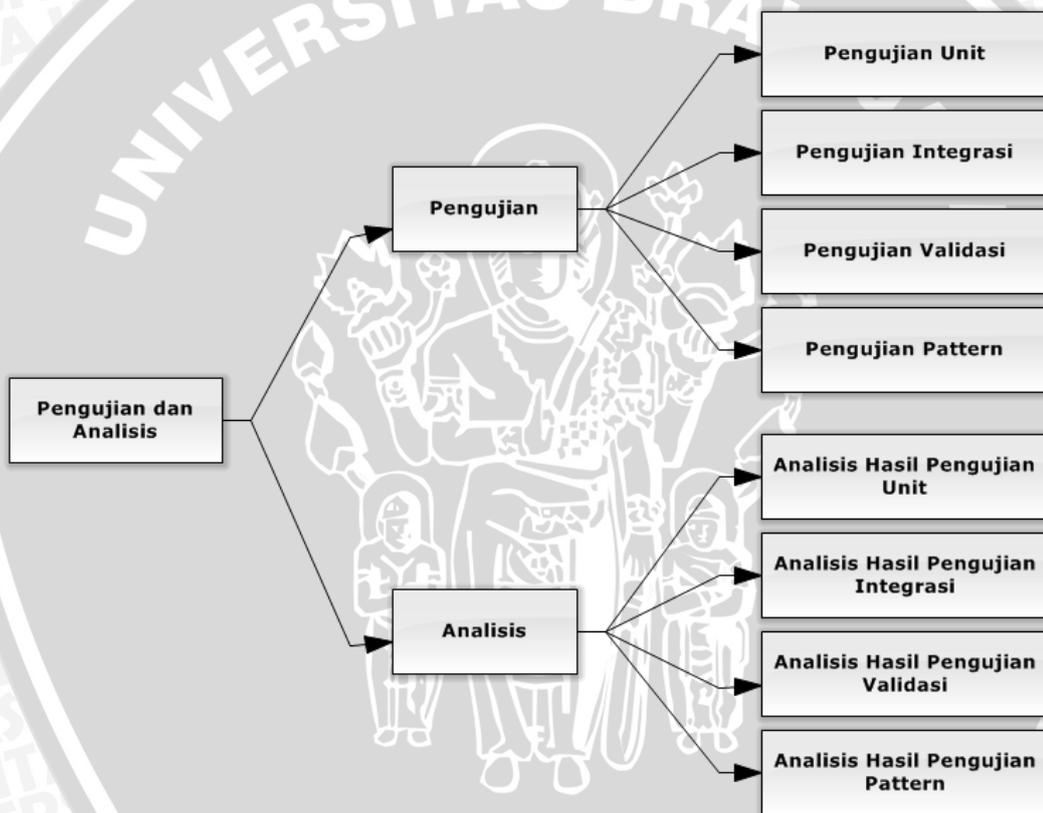
Antarmuka lihat pemetaan kecamatan merupakan halaman yang menampilkan pemetaan dari cakupan-cakupan KIA. Pemetaan dari 4 macam kohort diimplementasikan pada 3 tab yang terdiri dari cakupan ibu, cakupan bayi dan cakupan balita apras. Pemetaan yang ditampilkan berdasarkan tahun dan bulan yang dipilih pegawai puskesmas. Pada halaman ini menampilkan pemetaan cakupan yang dicapai pada masing-masing desa. Implementasi antarmuka lihat pemetaan kecamatan dapat dilihat pada Gambar 5.33 berikut,



Gambar 5.33 Antarmuka lihat pemetaan desa

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini menjelaskan tentang pengujian dan analisis dari hasil implementasi yang telah diperoleh dari bab sebelumnya. Proses pengujian dimulai dari pengujian code program yang akan dilakukan pada pengujian unit, kemudian dilanjutkan pengujian arsitektur atau desain yang akan dilakukan pada pengujian integrasi, kemudian dilanjutkan dengan pengujian sistem atau fungsionalitas yang akan dilakukan pada pengujian validasi dan yang terakhir adalah pengujian pada pola perancangan yang digunakan yaitu meguji pattern *factory method*. Sedangkan proses analisis bertujuan untuk mendapatkan kesimpulan dari semua hasil pengujian yang telah dilakukan. Tahapan pengujian dan analisis SIPKIA dapat dilihat pada Gambar 6.1 berikut,



Gambar 6.1 Tahap-tahap pengujian dan analisis

6.1 Pengujian

Pengujian yang dilakukan terdiri dari pengujian unit, pengujian integrasi, pengujian validasi dan pengujian pattern. Berikut proses pengujian dari masing-masing pengujian yang dilakukan,



6.1.1 Pengujian Unit

Tujuan dari pengujian unit adalah untuk memastikan beberapa algoritma yang memiliki prioritas tinggi telah diimplementasikan sesuai dengan yang diharapkan. Pengujian unit dilakukan menggunakan pengujian basis path. Algoritma yang diuji dalam pengujian ini adalah algoritma lihat register pada kelas C_Bidan dan algoritma tambah register pada *concrete creator* dalam hal ini kelas BidanDesaKidal.

1. Pengujian Algoritma Lihat Register

Algoritma lihat register pada kelas C_Bidan merupakan algoritma dengan prioritas tinggi dengan tujuan mengambil data register pada fungsi factory. Gambar 6.2 memaparkan algoritma lihat register beserta node flowgraph.

```
public function lihatRegister($bidan,$jenis)
{
  (1)if ($bidan=="kidal") {
    (2)$factoryKidal = new BidanDesaKidal();
    $hasil['data']=$factoryKidal->lihatRegister($jenis);
  (3)}else if ($bidan=="kambangan") {
    (4)$factoryKambangan = new BidanDesaKambangan();
    $hasil['data']=$factoryKambangan->lihatRegister($jenis);
  (5)}else if ($bidan=="ngingit") {
    (6)$factoryNgingit = new BidanDesaNgingit();
    $hasil['data']=$factoryNgingit->lihatRegister($jenis);
  (7)}else if ($bidan=="pandanajeng") {
    (8)$factoryPandanajeng = new BidanDesaPandanajeng();
    $hasil['data']=$factoryPandanajeng->lihatRegister($jenis);
  (9)}else if ($bidan=="pulungdowo") {
    (10)$factoryPulungdowo = new BidanDesaPulungdowo();
    $hasil['data']=$factoryPulungdowo->lihatRegister($jenis);
  (11)}else if ($bidan=="bokor") {
    (12)$factoryBokor = new BidanDesaBokor();
    $hasil['data']=$factoryBokor->lihatRegister($jenis);
  (13)}else if ($bidan=="slamet") {
    (14)$factorySlamet = new BidanDesaSlamet();
    $hasil['data']=$factorySlamet->lihatRegister($jenis);
  (15)}else if ($bidan=="tumpang") {
    (16)$factoryTumpang = new BidanDesaTumpang();
    $hasil['data']=$factoryTumpang->lihatRegister($jenis);
  (17)}else if ($bidan=="dampul") {
    (18)$factoryDampul = new BidanDesaDampul();
    $hasil['data']=$factoryDampul->lihatRegister($jenis);
  (19)}else if ($bidan=="duwet") {
    (20)$factoryDuwet = new BidanDesaDuwet();
    $hasil['data']=$factoryDuwet->lihatRegister($jenis);
  (21)}else if ($bidan=="benjor") {
    (22)$factoryBenjor = new BidanDesaBenjor();
    $hasil['data']=$factoryBenjor->lihatRegister($jenis);
  (23)}else if ($bidan=="malangsuko") {
    (24)$factoryMalangsuko = new BidanDesaMalangsuko();
    $hasil['data']=$factoryMalangsuko->lihatRegister($jenis);
  }
}
```

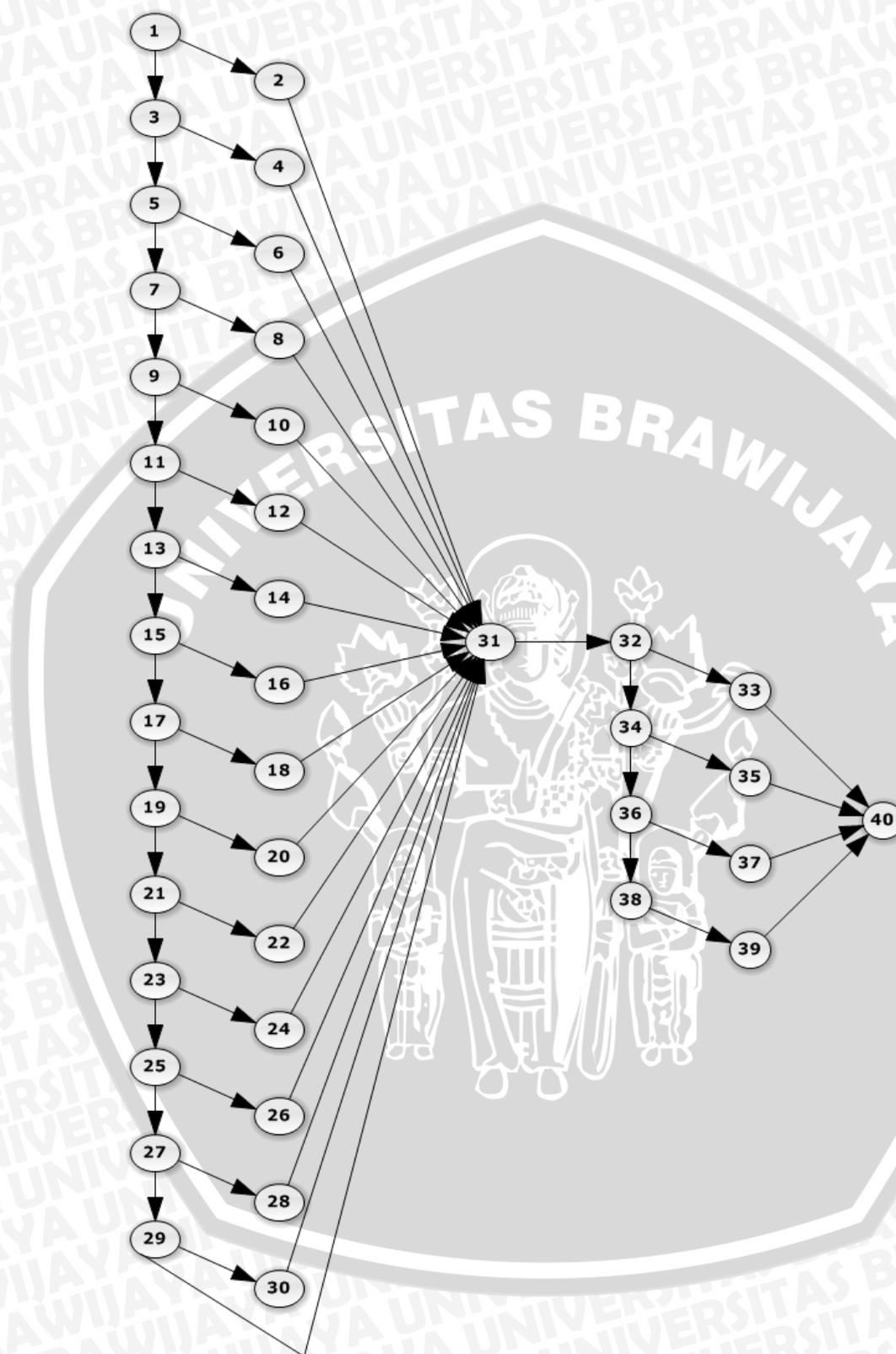
```

(25)}else if ($bidan=="wringinsongo") {
(26)$factoryWringinsongo = new BidanDesaWringinsongo();
    $hasil['data']=$factoryWringinsongo->lihatRegister($jenis);
(27)}else if ($bidan=="jeru") {
(28)$factoryJeru = new BidanDesaJeru();
    $hasil['data']=$factoryJeru->lihatRegister($jenis);
(29)}else if ($bidan=="tulusbesar") {
(30)$factoryTulusbesar = new BidanDesaTulusBesar();
    $hasil['data']=$factoryTulusbesar->lihatRegister($jenis);
}
(31)$view="";
(32)if ($jenis=="ibu") {
(33)$view="B_KohortIbu";
(34)}else if ($jenis=="bayi") {
(35)$view="B_KohortBayi";
(36)}else if ($jenis=="balita") {
(37)$view="B_KohortBalita";
(38)}else if ($jenis=="apras") {
(39)$view="B_KohortApras";
}
(40)$this->load->view($view, $hasil);
}

```

Gambar 6.2 Pembentukan node algoritma lihat register

Pada pembentukan node algoritma lihat register, node yang terbentuk sebanyak empat puluh. Dari node yang sudah terbentuk langkah selanjutnya adalah membuat flowgraph dari algoritma lihat register berdasarkan node-node yang ditentukan. Flowgraph algoritma lihat register dapat dilihat pada Gambar 6.3 berikut,



Gambar 6.3 Flowgraph algoritma lihat register

Berdasarkan flowgraph algoritma lihat register yang dapat dilihat pada Gambar 6.3 maka dapat dihitung nilai *cyclometric complexity* $V(G)$ sebagai berikut,

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 58 - 40 + 2 \\ &= 20 \end{aligned}$$

$$\begin{aligned} V(G) &= P + 1 \\ &= 19 + 1 \\ &= 20 \end{aligned}$$

Sehingga dari hasil persamaan *cyclometric complexity* maka didapatkan jalur independen yaitu,

1. 1-2-31-32-33-40
2. 1-3-4-31-32-33-40
3. 1-3-5-6-31-32-33-40
4. 1-3-5-7-8-31-32-33-40
5. 1-3-5-7-9-10-31-32-33-40
6. 1-3-5-7-9-11-12-31-32-33-40
7. 1-3-5-7-9-11-13-14-31-32-33-40
8. 1-3-5-7-9-11-13-15-16-31-32-33-40
9. 1-3-5-7-9-11-13-15-17-18-31-32-33-40
10. 1-3-5-7-9-11-13-15-17-19-20-31-32-33-40
11. 1-3-5-7-9-11-13-15-17-19-21-22-31-32-33-40
12. 1-3-5-7-9-11-13-15-17-19-21-23-24-31-32-33-40
13. 1-3-5-7-9-11-13-15-17-19-21-23-25-26-31-32-33-40
14. 1-3-5-7-9-11-13-15-17-19-21-23-25-27-28-31-32-33-40
15. 1-3-5-7-9-11-13-15-17-19-21-23-25-27-29-30-31-32-33-40
16. 1-3-5-7-9-11-13-15-17-19-21-23-25-27-29-31-32-33-40
17. 1-3-5-7-9-11-13-15-17-19-21-23-25-27-29-30-31-32-34-35-40
18. 1-3-5-7-9-11-13-15-17-19-21-23-25-27-29-30-31-32-34-36-37-40
19. 1-3-5-7-9-11-13-15-17-19-21-23-25-27-29-30-31-32-34-36-38-39-40
20. 1-3-5-7-9-11-13-15-17-19-21-23-25-27-29-30-31-32-34-36-38-40

Berdasarkan 20 jalur independen yang telah didefinisikan tersebut maka dapat dibentuk kasus ujinya. Tabel 6.1 memaparkan kasus uji algoritma lihat register.



Tabel 6.1 Kasus uji algoritma lihat register

No	Jalur	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	1-2-31-32-33-40	Bidan="kidal" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa kidal	Sistem menampilkan data register kohort ibu pada desa kidal	Valid
2	1-3-4-31-32-33-40	Bidan="kambingan" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa kambingan	Sistem menampilkan data register kohort ibu pada desa kambingan	Valid
3	1-3-5-6-31-32-33-40	Bidan="ngingit" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa ngingit	Sistem menampilkan data register kohort ibu pada desa ngingit	
4	1-3-5-7-8-31-32-33-40	Bidan="pandanajeng" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa pandanajeng	Sistem menampilkan data register kohort ibu pada desa pandanajeng	Valid
5	1-3-5-7-9-10-31-32-33-40	Bidan="pulungdowo" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa pulungdowo	Sistem menampilkan data register kohort ibu pada desa pulungdowo	Valid
6	1-3-5-7-9-11-12-31-32-33-40	Bidan="bokor" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa bokor	Sistem menampilkan data register kohort ibu pada desa bokor	Valid
7	1-3-5-7-9-11-13-14-31-32-33-40	Bidan="slamet" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa slamet	Sistem menampilkan data register kohort ibu pada desa slamet	Valid
8	1-3-5-7-9-11-13-15-16-31-32-33-40	Bidan="tumpang" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa tumpang	Sistem menampilkan data register kohort ibu pada desa tumpang	Valid
9	1-3-5-7-9-11-13-15-17-18-31-32-33-40	Bidan="dampul" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa dampul	Sistem menampilkan data register kohort ibu pada desa dampul	Valid
10	1-3-5-7-9-11-13-15-17-19-20-31-32-33-40	Bidan="duwet" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa duwet	Sistem menampilkan data register kohort ibu pada desa duwet	Valid

Tabel 6.1 Kasus uji algoritma lihat register (lanjutan)

No	Jalur	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
11	1-3-5-7-9-11-13-15-17-19-21-22-31-32-33-40	Bidan= "benjor" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa benjor	Sistem menampilkan data register kohort ibu pada desa benjor	Valid
12	1-3-5-7-9-11-13-15-17-19-21-23-24-31-32-33-40	Bidan= "malangsuko" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa malangsuko	Sistem menampilkan data register kohort ibu pada desa malangsuko	Valid
13	1-3-5-7-9-11-13-15-17-19-21-23-25-26-31-32-33-40	Bidan= "wringinsongo" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa wringinsongo	Sistem menampilkan data register kohort ibu pada desa wringinsongo	Valid
14	1-3-5-7-9-11-13-15-17-19-21-23-25-27-28-31-32-33-40	Bidan= "jeru" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa jeru	Sistem menampilkan data register kohort ibu pada desa jeru	Valid
15	1-3-5-7-9-11-13-15-17-19-21-23-25-27-29-30-31-32-33-40	Bidan= "tulusbesar" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa tulusbesar	Sistem menampilkan data register kohort ibu pada desa tulusbesar	Valid
16	1-3-5-7-9-11-13-15-17-19-21-23-25-27-29-31-32-33-40	Bidan= "noname" dan jenis = "ibu"	Sistem menampilkan halaman error	Sistem menampilkan halaman error	Valid
17	1-3-5-7-9-11-13-15-17-19-21-23-25-27-29-30-31-32-34-35-40	Bidan= "tulusbesar" dan jenis = "bayi"	Sistem menampilkan data register kohort bayi pada desa tulusbesar	Sistem menampilkan data register kohort bayi pada desa tulusbesar	Valid
18	1-3-5-7-9-11-13-15-17-19-21-23-25-27-29-30-31-32-34-36-37-40	Bidan= "tulusbesar" dan jenis = "balita"	Sistem menampilkan data register kohort balita pada desa tulusbesar	Sistem menampilkan data register kohort balita pada desa tulusbesar	Valid

Tabel 6.1 Kasus uji algoritma lihat register (lanjutan)

No	Jalur	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
19	1-3-5-7-9-11-13-15-17-19-21-23-25-27-29-30-31-32-34-36-38-39-40	Bidan="tulusbesar" dan jenis = "apras"	Sistem menampilkan data register kohort apras pada desa tulusbesar	Sistem menampilkan data register kohort apras pada desa tulusbesar	Valid
20	1-3-5-7-9-11-13-15-17-19-21-23-25-27-29-31-32-34-36-38-40	Bidan="noname" dan jenis = "noname"	Sistem menampilkan halaman error	Sistem menampilkan halaman error	Valid

2. Pengujian Algoritma Tambah Register

Algoritma tambah register pada kelas *BidanDesaKidal* merupakan algoritma dengan prioritas tinggi dengan tujuan menambahkan data register baru pada komponen *concrete creator*. Gambar 6.4 memaparkan algoritma tambah register beserta node flowgraph.

```
function tambahRegister($jenis){
  (1)switch ($jenis) {
  (2)case "ibu":
  (3)$register = new RegisterKohortIbu;
    $hasil=$register->addRegister("kidal");
    break;
  (4)case "bayi":
  (5)$register = new RegisterKohortBayi;
    $register->addRegister("kidal");
    break;
  (6)case "balita":
  (7)$register = new RegisterKohortBalita;
    $register->addRegister("kidal");
    break;
  (8)case "apras":
  (9)$register = new RegisterKohortApras;
    $register->addRegister("kidal");
    break;
  (10)default:
  (11)echo "Salah Parameter";
    break;
  (12)}
}
```

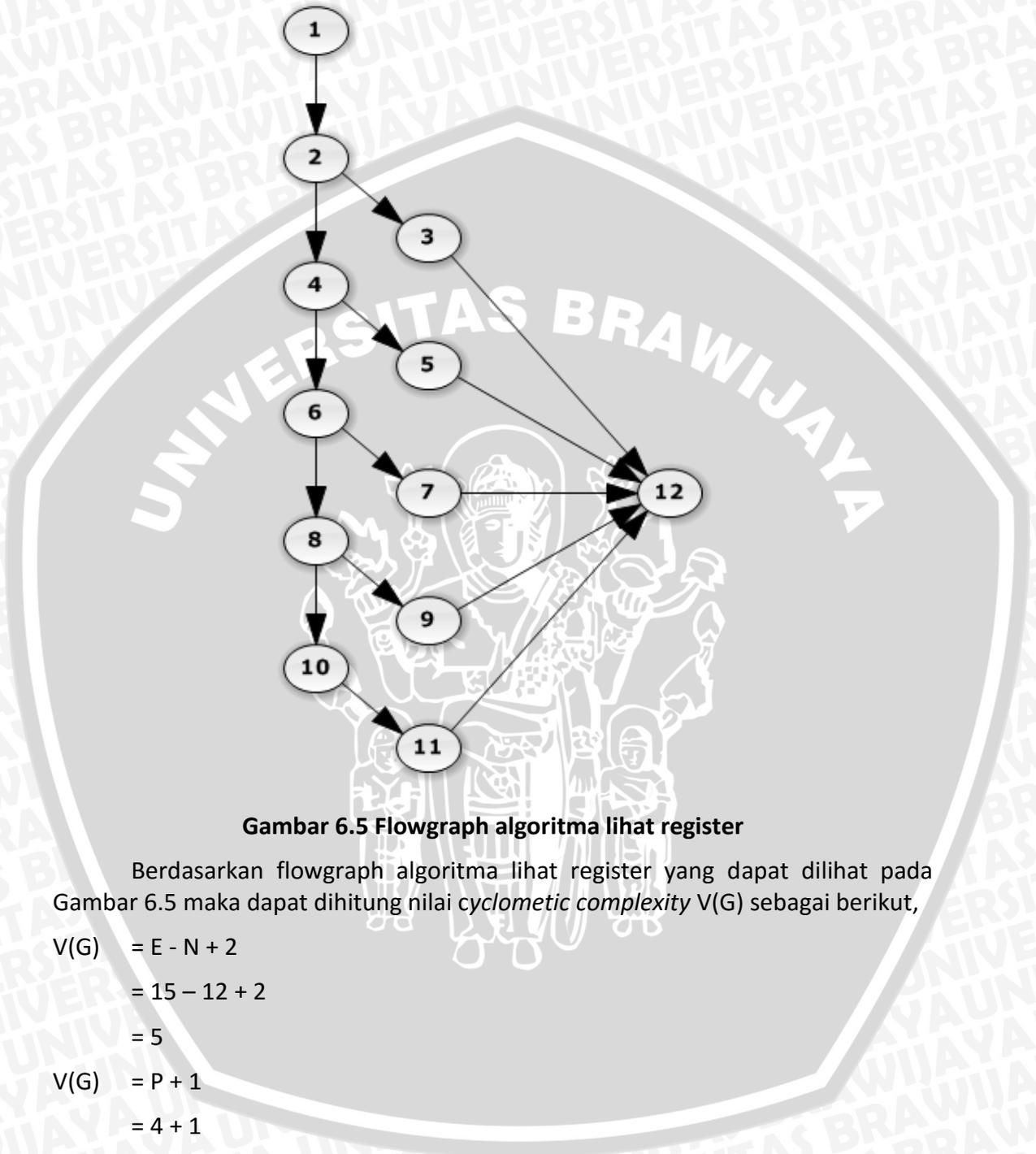
Gambar 6.4 Node algoritma tambah register

Pada pembentukan node algoritma tambah register, node yang terbentuk sebanyak duabelas. Dari node yang sudah terbentuk langkah selanjutnya adalah membuat flowgraph dari algoritma tambah register berdasarkan node-node yang



repository.ub.ac.id

ditentukan. Flowgraph algoritma tambah register dapat dilihat pada gambar 6.5 berikut,



Gambar 6.5 Flowgraph algoritma lihat register

Berdasarkan flowgraph algoritma lihat register yang dapat dilihat pada Gambar 6.5 maka dapat dihitung nilai *cyclometric complexity* $V(G)$ sebagai berikut,

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 15 - 12 + 2 \\ &= 5 \end{aligned}$$

$$\begin{aligned} V(G) &= P + 1 \\ &= 4 + 1 \\ &= 5 \end{aligned}$$

Sehingga dari hasil persamaan *cyclometric complexity* maka didapatkan jalur independen yaitu,



1. 1-2-3-12
2. 1-2-4-5-12
3. 1-2-4-6-7-12
4. 1-2-4-6-8-9-12
5. 1-2-4-6-8-10-11-12

Berdasarkan 5 jalur independen yang telah didefinisikan tersebut maka dapat dibentuk kasus ujinya. Tabel 6.2 memaparkan kasusu uji algoritma lihat register.

Tabel 6.2 Kasus uji algoritma lihat register

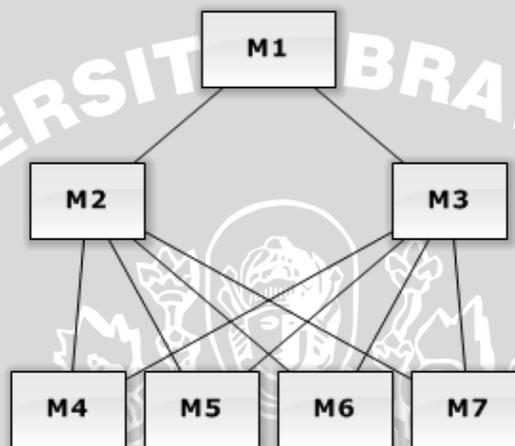
No	Jalur	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	1-2-3-12	Jenis = "ibu"	Sistem menampilkan halaman lihat register ibu	Sistem menampilkan halaman lihat register ibu	Valid
2	1-2-4-5-12	Jenis = "bayi"	Sistem menampilkan halaman lihat register bayi	Sistem menampilkan halaman lihat register bayi	Valid
3	1-2-4-6-7-12	Jenis = "balita"	Sistem menampilkan halaman lihat register balita	Sistem menampilkan halaman lihat register balita	Valid
4	1-2-4-6-8-9-12	Jenis = "apras"	Sistem menampilkan halaman lihat register apras	Sistem menampilkan halaman lihat register apras	Valid
5	1-2-4-6-8-10-11-12	Jenis = "noname"	Sistem menampilkan halaman error	Sistem menampilkan halaman error	Valid

6.1.2 Pengujian Integrasi

Pengujian integrasi adalah teknik yang sistematis untuk membangun arsitektur perangkat lunak dengan melakukan tes untuk mengungkap kesalahan yang terkait dengan tampilan program. Tujuannya adalah untuk mengambil komponen unit diuji dan membangun program terstruktur yang telah ditentukan pada tahap desain (Pressman,2010). Pada kasus ini pengujian integrasi menggunakan strategi bottom-up yaitu pengujian dimulai dari unit terkecil sampai ke dalam sebuah modul atau komponen yang lebih besar. Modul yang akan di uji dalam pengujian integrasi yaitu pengujian modul lihat register dan tambah register.

1. Pengujian Integrasi Modul Lihat Register

Pengujian modul lihat register terdiri dari unit `viewRegister()` pada komponen produk yaitu kelas `RegisterKohortIbu`, `RegisteKohortBayi`, `RegisteKohortBalita` dan `RegisteKohortApras`. Kemudian unit `lihatRegister()` pada komponen Creator yaitu pada kelas `BidanDesaKidal`, `BidanDesaKambingan` dan lainnya. Selanjutnya unit `lihatRegister()` pada komponen client yaitu pada kelas `C_Bidan`. Dari keseluruhan unit akan diintegrasikan untuk menghasilkan pengujian dalam satu modul lihat register. Masing-masing unit akan disimbolkan dalam sebuah modul atau bagian. Bagan dari siklus bottom-up integration dapat dilihat pada Gambar 6.6 berikut,



Gambar 6.6 Hirarki bottom-up integration lihat register

Pada Gambar 6.6 terdiri dari 7 modul yang mewakili masing-masing unit atau kumpulan unit. M1 merupakan modul utama yaitu fungsi `lihatRegister()` pada kelas `C_Bidan`, M2 merupakan modul yang akan dipanggil oleh modul M1 yaitu fungsi `lihatRegister()` pada kelas `BidanDesaKidal`, M3 juga merupakan modul yang akan dipanggil oleh modul M1 yaitu fungsi `lihatRegister()` pada kelas `BidanDesaKambingan()`, M4 merupakan modul yang akan dipanggil oleh M2 dan M3 yaitu fungsi `viewRegister()` pada kelas `RegisterKohortIbu`, M5 merupakan modul yang akan dipanggil oleh M2 dan M3 yaitu fungsi `viewRegister()` pada kelas `RegisterKohortBayi`, M6 merupakan modul yang akan dipanggil oleh M2 dan M3 yaitu fungsi `viewRegister()` pada kelas `RegisterKohortBalita`, dan M7 merupakan modul yang akan dipanggil oleh M2 dan M3 yaitu fungsi `viewRegister()` pada kelas `RegisterKohortApras`. Pengujian yang dilakukan adalah menggunakan strategi bottom-up sehingga modul yang diuji dimulai dari unit terkecil atau terbawah yaitu dengan urutan pengujian M7-M6-M5-M4-M3-M2-M1. Hasil integrasi dari masing-masing modul tersebut dapat dilihat pada Tabel 6.3 berikut,

Tabel 6.3 Pengujian integrasi tiap modul lihat register

Modul	Fungsi	Kelas	Modul yang dipanggil	Hasil yang diharapkan	Hasil Integrasi
M7	viewRegister()	RegisterKohortApras	-	Sesuai	Valid
M6	viewRegister()	RegisterKohortBalita	-	Sesuai	Valid
M5	viewRegister()	RegisterKohortBayi	-	Sesuai	Valid
M4	viewRegister()	RegisterKohortIbu	-	Sesuai	Valid
M3	lihatRegister()	BidanDesaKambingan	M4,M5,M6, M7	Sesuai	Valid
M2	lihatRegister()	BidanDesaKidal	M4,M5,M6, M7	Sesuai	Valid
M1	lihatRegister()	C_Bidan	M2,M3	Sesuai	Valid

Dari hasil pengujian pada Tabel 6.3 merupakan pengujian secara keseluruhan dari tiap modul yang dimulai dari modul terkecil sampai ke modul utama. Pengujian tersebut sudah mewakili dari masing-masing kasus uji dari tiap modul. Untuk detail pengujian pada masing-masing kasus uji dari modul utama dapat dilihat pada Tabel 6.4 berikut,

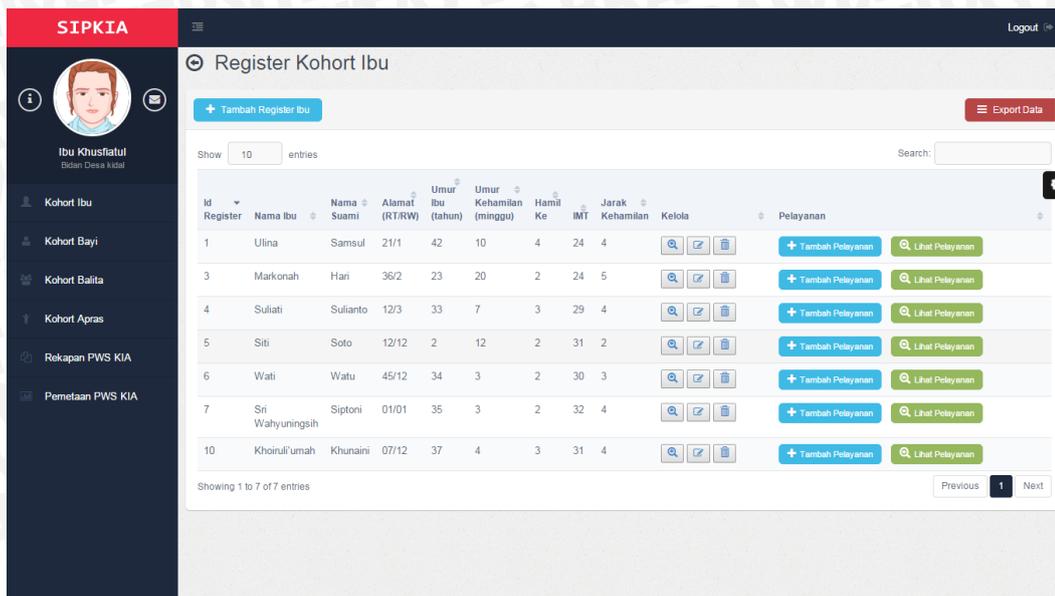
Tabel 6.4 Pengujian integrasi modul utama lihat register

No	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Bidan= "kidal" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa kidal	Sistem menampilkan data register kohort ibu pada desa kidal	Valid
2	Bidan= "kambingan" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa kambingan	Sistem menampilkan data register kohort ibu pada desa kambingan	Valid
3	Bidan= "ngingit" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa ngingit	Sistem menampilkan data register kohort ibu pada desa ngingit	
4	Bidan= "pandanajeng" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa pandanajeng	Sistem menampilkan data register kohort ibu pada desa pandanajeng	Valid
5	Bidan= "pungdowo" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa pungdowo	Sistem menampilkan data register kohort ibu pada desa pungdowo	Valid
6	Bidan= "bokor" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa bokor	Sistem menampilkan data register kohort ibu pada desa bokor	Valid
7	Bidan= "slamet" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa slamet	Sistem menampilkan data register kohort ibu pada desa slamet	Valid
8	Bidan= "tumpang" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa tumpang	Sistem menampilkan data register kohort ibu pada desa tumpang	Valid

Tabel 6.4 Pengujian integrasi modul utama lihat register (lanjutan)

No	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
9	Bidan= "dampul" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa dampul	Sistem menampilkan data register kohort ibu pada desa dampul	Valid
10	Bidan= "duwet" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa duwet	Sistem menampilkan data register kohort ibu pada desa duwet	Valid
11	Bidan= "benjor" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa benjor	Sistem menampilkan data register kohort ibu pada desa benjor	Valid
12	Bidan= "malangsuko" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa malangsuko	Sistem menampilkan data register kohort ibu pada desa malangsuko	Valid
13	Bidan= "wringinsongo" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa wringinsongo	Sistem menampilkan data register kohort ibu pada desa wringinsongo	Valid
14	Bidan= "jeru" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa jeru	Sistem menampilkan data register kohort ibu pada desa jeru	Valid
15	Bidan= "tulusbesar" dan jenis = "ibu"	Sistem menampilkan data register kohort ibu pada desa tulusbesar	Sistem menampilkan data register kohort ibu pada desa tulusbesar	Valid
16	Bidan= "noname" dan jenis = "ibu"	Sistem menampilkan halaman error	Sistem menampilkan halaman error	Valid
17	Bidan= "tulusbesar" dan jenis = "bayi"	Sistem menampilkan data register kohort bayi pada desa tulusbesar	Sistem menampilkan data register kohort bayi pada desa tulusbesar	Valid
18	Bidan= "tulusbesar" dan jenis = "balita"	Sistem menampilkan data register kohort balita pada desa tulusbesar	Sistem menampilkan data register kohort balita pada desa tulusbesar	Valid
19	Bidan= "tulusbesar" dan jenis = "apras"	Sistem menampilkan data register kohort apras pada desa tulusbesar	Sistem menampilkan data register kohort apras pada desa tulusbesar	Valid
20	Bidan= "noname" dan jenis = "noname"	Sistem menampilkan halaman error	Sistem menampilkan halaman error	Valid

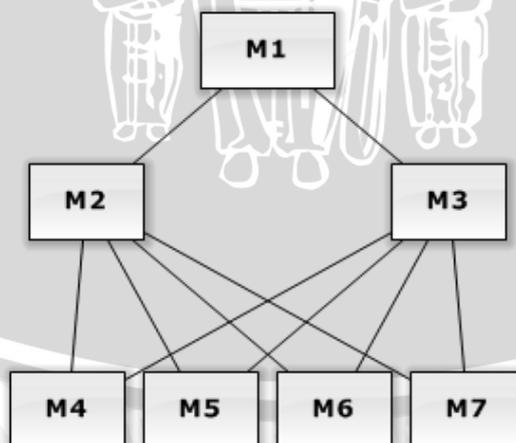
Dari hasil pengujian pada Tabel 6.4 integrasi pada modul utama dapat tervalidasi. Untuk tampilan interface dari hasil uji modul lihat register ibu pada dapat dilihat pada Gambar 6.7 berikut,



Gambar 6.7 Tampilan hasil pengujian integrasi modul lihat register

2. Pengujian Integrasi Modul Tambah Register

Pengujian modul tambah register terdiri dari unit `addRegister()` pada komponen produk yaitu kelas `RegisterKohortIbu`, `RegisteKohortBayi`, `RegisteKohortBalita` dan `RegisteKohortApras`. Kemudian unit `tambahRegister()` pada komponen *creator* yaitu pada kelas `BidanDesaKidal`, `BidanDesaKambangan` dan lainnya. Selanjutnya unit `tambahRegister()` pada komponen client yaitu pada kelas `C_Bidan`. Dari keseluruhan unit akan diintegrasikan untuk menghasilkan pengujian dalam satu modul lihat register. Masing-masing unit akan disimbolkan dalam sebuah modul atau bagian. Bagan dari siklus bottom-up integration dapat dilihat pada Gambar 6.8 berikut,



Gambar 6.8 Hirarki bottom-up integration tambah register

Pada Gambar 6.8 terdiri dari 7 modul yang mewakili masing-masing unit atau kumpulan unit. M1 merupakan modul utama yaitu fungsi tambahRegister() pada kelas C_Bidan, M2 merupakan modul yang akan dipanggil oleh modul M1 yaitu fungsi tambahRegister() pada kelas BidanDesaKidal, M3 juga merupakan modul yang akan dipanggil oleh modul M1 yaitu fungsi tambahRegister() pada kelas BidanDesaKambingan(), M4 merupakan modul yang akan dipanggil oleh M2 dan M3 yaitu fungsi addRegister() pada kelas RegisterKohortIbu, M5 merupakan modul yang akan dipanggil oleh M2 dan M3 yaitu fungsi addRegister() pada kelas RegisterKohortBayi, M6 merupakan modul yang akan dipanggil oleh M2 dan M3 yaitu fungsi addRegister() pada kelas RegisterKohortBalita, dan M7 merupakan modul yang akan dipanggil oleh M2 dan M3 yaitu fungsi addRegister() pada kelas RegisterKohortApras. Pengujian yang dilakukan adalah menggunakan strategi bottom-up sehingga modul yang diuji dimulai dari unit terkecil atau terbawah yaitu dengan urutan pengujian M7-M6-M5-M4-M3-M2-M1. Hasil integrasi dari masing-masing modul tersebut dapat dilihat pada Tabel 6.5 berikut,

Tabel 6.5 Pengujian integrasi tiap modul tambah register

Modul	Fungsi	Kelas	Modul yang dipanggil	Hasil yang diharapkan	Hasil Integrasi
M7	addRegister()	RegisterKohortApras	-	Sesuai	Valid
M6	addRegister()	RegisterKohortBalita	-	Sesuai	Valid
M5	addRegister()	RegisterKohortBayi	-	Sesuai	Valid
M4	addRegister()	RegisterKohortIbu	-	Sesuai	Valid
M3	tambahRegister()	BidanDesaKambingan	M4,M5,M6,M7	Sesuai	Valid
M2	tambahRegister ()	BidanDesaKidal	M4,M5,M6,M7	Sesuai	Valid
M1	tambahRegister ()	C_Bidan	M2,M3	Sesuai	Valid

Dari hasil pengujian pada Tabel 6.5 merupakan pengujian secara keseluruhan dari tiap modul yang dimulai dari modul terkecil sampai ke modul utama. Pengujian tersebut sudah mewakili dari masing-masing kasus uji dari tiap modul. Untuk detail pengujian pada masing-masing kasus uji dari modul utama dapat dilihat pada Tabel 6.6 berikut,

Tabel 6.6 Pengujian integrasi modul utama tambah register

No	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Bidan= "kidal" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa kidal	Sistem dapat menambahkan data register kohort ibu pada desa kidal	Valid
2	Bidan= "kambingan" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa kambingan	Sistem dapat menambahkan data register kohort ibu pada desa kambingan	Valid

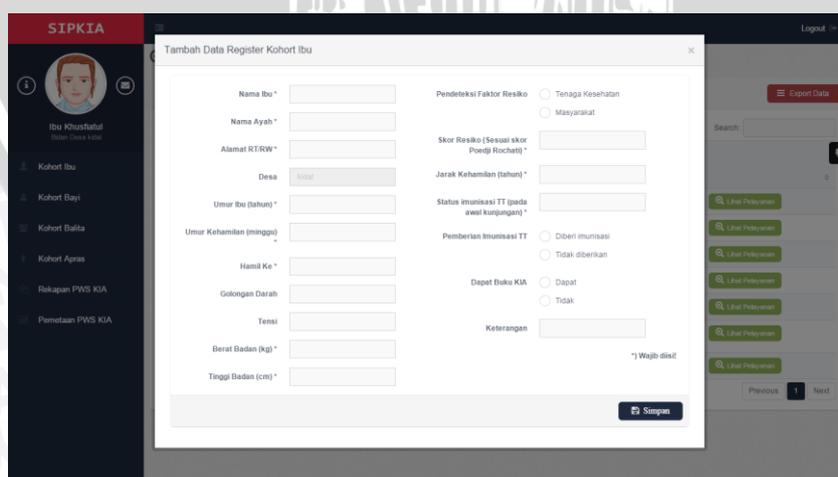
Tabel 6.6 Pengujian integrasi modul utama tambah register (lanjutan)

No	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
3	Bidan= "ngingit" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa ngingit	Sistem dapat menambahkan data register kohort ibu pada desa ngingit	
4	Bidan= "pandanajeng" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa pandanajeng	Sistem dapat menambahkan data register kohort ibu pada desa pandanajeng	Valid
5	Bidan= "pulungdowo" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa pulungdowo	Sistem dapat menambahkan data register kohort ibu pada desa pulungdowo	Valid
6	Bidan= "bokor" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa bokor	Sistem dapat menambahkan data register kohort ibu pada desa bokor	Valid
7	Bidan= "slamet" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa slamet	Sistem dapat menambahkan data register kohort ibu pada desa slamet	Valid
8	Bidan= "tumpang" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa tumpang	Sistem dapat menambahkan data register kohort ibu pada desa tumpang	Valid
9	Bidan= "dampul" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa dampul	Sistem dapat menambahkan data register kohort ibu pada desa dampul	Valid
10	Bidan= "duwet" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa duwet	Sistem dapat menambahkan data register kohort ibu pada desa duwet	Valid
11	Bidan= "benjor" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa benjor	Sistem dapat menambahkan data register kohort ibu pada desa benjor	Valid
12	Bidan= "malangsuko" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa malangsuko	Sistem dapat menambahkan data register kohort ibu pada desa malangsuko	Valid
13	Bidan= "wringinsongo" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa wringinsongo	Sistem dapat menambahkan data register kohort ibu pada desa wringinsongo	Valid

Tabel 6.6 Pengujian integrasi modul utama tambah register (lanjutan)

No	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
14	Bidan= "jeru" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa jeru	Sistem dapat menambahkan data register kohort ibu pada desa jeru	Valid
15	Bidan= "tulusbesar" dan jenis = "ibu"	Sistem dapat menambahkan data register kohort ibu pada desa tulusbesar	Sistem dapat menambahkan data register kohort ibu pada desa tulusbesar	Valid
16	Bidan= "noname" dan jenis = "ibu"	Sistem dapat menambahkan halaman error	Sistem dapat menambahkan halaman error	Valid
17	Bidan= "tulusbesar" dan jenis = "bayi"	Sistem dapat menambahkan data register kohort bayi pada desa tulusbesar	Sistem dapat menambahkan data register kohort bayi pada desa tulusbesar	Valid
18	Bidan= "tulusbesar" dan jenis = "balita"	Sistem dapat menambahkan data register kohort balita pada desa tulusbesar	Sistem dapat menambahkan data register kohort balita pada desa tulusbesar	Valid
19	Bidan= "tulusbesar" dan jenis = "apras"	Sistem dapat menambahkan data register kohort apras pada desa tulusbesar	Sistem dapat menambahkan data register kohort apras pada desa tulusbesar	Valid
20	Bidan= "noname" dan jenis = "noname"	Sistem menampilkan halaman error	Sistem menampilkan halaman error	Valid

Dari hasil pengujian pada Tabel 6.6 integrasi pada modul utama dapat tervalidasi. Untuk tampilan interface dari hasil uji modul tambah register ibu dapat dilihat pada Gambar 6.9 berikut,



Gambar 6.9 Tampilan hasil pengujian integrasi modul tambah register

6.1.3 Pengujian Validasi

Pengujian validasi bertujuan untuk mengetahui apakah sistem yang telah dibangun telah menyediakan fungsi-fungsi yang dibutuhkan. Item-item yang telah dibuat dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan yang akan digunakan sebagai acuan untuk melakukan pengujian fungsional. Pengujian validasi menggunakan pengujian Black box, karena pengujian ini tidak memfokuskan alur jalannya algoritma program dan lebih ditekankan untuk menemukan kesesuaian antara kinerja sistem dengan daftar kebutuhan sistem. Berikut ini akan dijelaskan bagaimana proses pengujian fungsional untuk tiap-tiap kebutuhan sistem. Dalam melakukan pengujian validasi terdapat kasus-kasus yang digunakan untuk mengetahui kemungkinan-kemungkinan tindakan yang akan dilakukan oleh pengguna aplikasi. Berikut ini merupakan kasus-kasus yang sebagian besar akan dilakukan oleh pengguna aplikasi.

1. Kasus Uji Lihat Register

Tabel 6.7 Pengujian lihat register

Nama Kasus Uji	Uji Lihat Register
Objek Uji	Uji Halaman Lihat Register
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan data register yang tersimpan dari database yang terdiri dari register ibu, bayi, balita dan apras
Prosedur Uji	Menekan salah satu menu lihat register
Hasil yang Diharapkan	Sistem dapat menampilkan halaman lihat register beserta data register yang diminta yaitu data ibu, bayi, balita atau apras
Hasil Uji Coba	Valid

Sumber : Pengujian

2. Kasus Uji Tambah Register

Tabel 6.8 Pengujian tambah register

Nama Kasus Uji	Uji Tambah Register
Objek Uji	Uji Halaman Tambah Register
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan form tambah register dan menyimpan data register baru ke database
Prosedur Uji	Menekan tombol tambah register (ibu, bayi, balita atau apras), , menekan tombol simpan.
Hasil yang Diharapkan	Sistem dapat menampilkan halaman form tambah register dan dapat menyimpan data baru (data ibu, bayi, balita atau apras) ke database
Hasil Uji Coba	Valid

Sumber : Pengujian

3. Kasus Uji Ubah Register

Tabel 6.9 Pengujian ubah register

Nama Kasus Uji	Uji Ubah Register
Objek Uji	Uji Halaman Ubah Register
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan form ubah register dan menyimpan data register yang telah dirubah ke database
Prosedur Uji	Menekan tombol ubah register (ibu, bayi, balita atau apras), menekan tombol simpan.
Hasil yang Diharapkan	Sistem dapat menampilkan halaman form ubah register dan dapat menyimpan perubahan data (ibu, bayi, balita atau apras) ke database
Hasil Uji Coba	Valid

Sumber : Pengujian

4. Kasus Uji Hapus Register

Tabel 6.10 Pengujian hapus register

Nama Kasus Uji	Uji Hapus Register
Objek Uji	Uji Halaman Hapus Register
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan konfirmasi hapus register dan menghapus data register yang dipilih dari database
Prosedur Uji	Menekan tombol hapus register (ibu, bayi, balita atau apras) dan konfirmasi ya
Hasil yang Diharapkan	Sistem dapat menampilkan pop up konfirmasi hapus register dan dapat menghapus data register (ibu, bayi, balita atau apras) yang dipilih dari database
Hasil Uji Coba	Valid

Sumber : Pengujian

5. Kasus Uji Lihat Pelayanan

Tabel 6.11 Pengujian lihat pelayanan

Nama Kasus Uji	Uji Lihat Pelayanan
Objek Uji	Uji Halaman Lihat Pelayanan
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan data pelayanan yang tersimpan dari database yang terdiri dari pelayanan ibu, bayi, balita dan apras
Prosedur Uji	Menekan tombol lihat pelayanan
Hasil yang Diharapkan	Sistem dapat menampilkan halaman lihat pelayanan beserta data pelayanan yang diminta yaitu data ibu, bayi, balita atau apras
Hasil Uji Coba	Valid

6. Kasus Uji Tambah Pelayanan

Tabel 6.12 Pengujian tambah pelayanan

Nama Kasus Uji	Uji Tambah Pelayanan
Objek Uji	Uji Halaman Tambah Pelayanan
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan form tambah pelayanan dan menyimpan data pelayanan baru ke database
Prosedur Uji	Menekan tombol tambah pelayanan (ibu, bayi, balita atau apras), menekan tombol simpan.
Hasil yang Diharapkan	Sistem dapat menampilkan halaman form tambah pelayanan dan dapat menyimpan data baru (data ibu, bayi, balita atau apras) ke database
Hasil Uji Coba	Valid

Sumber : Pengujian

7. Kasus Uji Ubah Pelayanan

Tabel 6.13 Pengujian ubah pelayanan

Nama Kasus Uji	Uji Ubah Pelayanan
Objek Uji	Uji Halaman Ubah Pelayanan
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan form ubah pelayanan dan menyimpan data pelayanan yang telah dirubah ke database
Prosedur Uji	Menekan tombol ubah pelayanan (ibu, bayi, balita atau apras), menekan tombol simpan.
Hasil yang Diharapkan	Sistem dapat menampilkan halaman form ubah pelayanan dan dapat menyimpan perubahan data (ibu, bayi, balita atau apras) ke database
Hasil Uji Coba	Valid

Sumber : Pengujian

8. Kasus Uji Hapus Pelayanan

Tabel 6.14 Pengujian hapus pelayanan

Nama Kasus Uji	Uji Hapus Pelayanan
Objek Uji	Uji Halaman Hapus Pelayanan
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan konfirmasi hapus pelayanan dan menghapus data pelayanan yang dipilih dari database
Prosedur Uji	Menekan tombol hapus pelayanan (ibu, bayi, balita atau apras) dan konfirmasi ya
Hasil yang Diharapkan	Sistem dapat menampilkan pop up konfirmasi hapus pelayanan dan sistem dapat menghapus data pelayanan (ibu, bayi, balita atau apras) dari database
Hasil Uji Coba	Valid

Sumber : Pengujian

9. Kasus Uji Lihat Rekap Desa

Tabel 6.15 Pengujian lihat rekap desa

Nama Kasus Uji	Uji Lihat Rekap Desa
Objek Uji	Uji Halaman Lihat Rekap Desa
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan data rekap pada masing-masing desa pada tahun tertentu
Prosedur Uji	Menekan tombol rekap desa
Hasil yang Diharapkan	Sistem dapat menampilkan halaman rekap desa beserta data cakupan pada tahun tertentu
Hasil Uji Coba	Valid

Sumber : Pengujian

10. Kasus Uji Cetak Rekap Desa

Tabel 6.16 Pengujian cetak rekap desa

Nama Kasus Uji	Uji Cetak Rekap Desa
Objek Uji	Uji Cetak Rekap Desa
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk mencetak (export) data rekap desa ke dalam sebuah file yang dipilih.
Prosedur Uji	Menekan tombol export. Memilih salah satu jenis file.
Hasil yang Diharapkan	Sistem dapat mencetak data rekap desa ke dalam file yang dipilih oleh pengguna
Hasil Uji Coba	Valid

Sumber : Pengujian

11. Kasus Uji Lihat Rekap Kecamatan

Tabel 6.17 Pengujian lihat rekap kecamatan

Nama Kasus Uji	Uji Lihat Rekap Kecamatan
Objek Uji	Uji Halaman Lihat Rekap Kecamatan
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan data rekap sekecamatan pada bulan dan tahun tertentu
Prosedur Uji	Menekan tombol rekap kecamatan
Hasil yang Diharapkan	Sistem dapat menampilkan halaman rekap kecamatan beserta data cakupan pada bulan dan tahun tertentu
Hasil Uji Coba	Valid

Sumber : Pengujian

12. Kasus Uji Cetak Rekapam Kecamatan

Tabel 6.18 Pengujian cetak rekapam kecamatan

Nama Kasus Uji	Uji Cetak Rekapam Kecamatan
Objek Uji	Uji Cetak Rekapam Kecamatan
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk mencetak (export) data rekapam kecamatan ke dalam sebuah file yang dipilih.
Prosedur Uji	Menekan tombol export. Memilih salah satu jenis file.
Hasil yang Diharapkan	Sistem dapat mencetak data rekapam kecamatan ke dalam file yang dipilih oleh pengguna
Hasil Uji Coba	Valid

Sumber : Pengujian

13. Kasus Uji Lihat Pemetaan Desa

Tabel 6.19 Pengujian pemetaan desa

Nama Kasus Uji	Uji Lihat Pemetaan Desa
Objek Uji	Uji Halaman Lihat Pemetaan Desa
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan data pemetaan dan diagram cakupan pada masing-masing desa pada tahun tertentu
Prosedur Uji	Menekan tombol pemetaan desa
Hasil yang Diharapkan	Sistem dapat menampilkan halaman pemetaan desa beserta data status cakupan dan diagram cakupan pada tahun tertentu
Hasil Uji Coba	Valid

Sumber : Pengujian

14. Kasus Uji Cetak Pemetaan Desa

Tabel 6.20 Pengujian cetak rekapam desa

Nama Kasus Uji	Uji Cetak Pemetaan Desa
Objek Uji	Uji Cetak Pemetaan Desa
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk mencetak (export) data pemetaan desa ke dalam sebuah file yang dipilih.
Prosedur Uji	Menekan tombol export. Memilih salah satu jenis file.
Hasil yang Diharapkan	Sistem dapat mencetak data pemetaan desa ke dalam file yang dipilih oleh pengguna
Hasil Uji Coba	Valid

Sumber : Pengujian

15. Kasus Uji Lihat Pemetaan Kecamatan

Tabel 6.21 Pengujian lihat pemetaan kecamatan

Nama Kasus Uji	Uji Lihat Pemetaan Kecamatan
Objek Uji	Uji Halaman Lihat Pemetaan Kecamatan
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan data pemetaan sekecamatan dan diagram cakupan pada bulan dan tahun tertentu
Prosedur Uji	Menekan tombol pemetaan kecamatan
Hasil yang Diharapkan	Sistem dapat menampilkan halaman pemetaan kecamatan beserta data status cakupan dan diagram cakupan tahun tertentu
Hasil Uji Coba	Valid

Sumber : Pengujian

16. Kasus Uji Cetak Pemetaan Kecamatan

Tabel 6.22 Pengujian cetak pemetaan kecamatan

Nama Kasus Uji	Uji Cetak Pemetaan Kecamatan
Objek Uji	Uji Cetak Pemetaan Kecamatan
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk mencetak (export) data pemetaan kecamatan ke dalam sebuah file yang dipilih.
Prosedur Uji	Menekan tombol export. Memilih salah satu jenis file.
Hasil yang Diharapkan	Sistem dapat mencetak data pemetaan kecamatan ke dalam file yang dipilih oleh pengguna
Hasil Uji Coba	Valid

Sumber : Pengujian

17. Kasus Uji Perbarui Rekap

Tabel 6.23 Pengujian perbarui rekap

Nama Kasus Uji	Uji Lihat Rekap Desa
Objek Uji	Uji Halaman Lihat Rekap Desa
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk memperbarui data rekap masing-masing desa pada tahun tertentu
Prosedur Uji	Menekan tombol perbarui rekap
Hasil yang Diharapkan	Sistem dapat memberikan alert bahwa pembaharuan rekap telah dilakukan dan data yang akan ditampilkan berupa data yang telah diperbarui
Hasil Uji Coba	Valid

Sumber : Pengujian

18. Kasus Uji Lihat User

Tabel 6.24 Pengujian lihat user

Nama Kasus Uji	Uji Lihat User
Objek Uji	Uji Halaman Lihat User
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan data user yang tersimpan dari database
Prosedur Uji	Menekan menu kelola user
Hasil yang Diharapkan	Sistem dapat menampilkan halaman lihat user beserta data user yang terdaftar
Hasil Uji Coba	Valid

Sumber : Pengujian

19. Kasus Uji Tambah User

Tabel 6.25 Pengujian tambah user

Nama Kasus Uji	Uji Tambah User
Objek Uji	Uji Halaman Tambah User
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan form tambah user dan menyimpan data user baru ke database
Prosedur Uji	Menekan tombol tambah user, menekan tombol simpan.
Hasil yang Diharapkan	Sistem dapat menampilkan halaman form tambah user dan dapat menyimpan data baru ke database
Hasil Uji Coba	Valid

Sumber : Pengujian

20. Kasus Uji Ubah User

Tabel 6.26 Pengujian ubah user

Nama Kasus Uji	Uji Ubah User
Objek Uji	Uji Halaman Ubah User
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan form ubah user dan menyimpan data user yang telah dirubah ke database
Prosedur Uji	Menekan tombol ubah user, menekan tombol simpan.
Hasil yang Diharapkan	Sistem dapat menampilkan halaman form ubah user dan dapat menyimpan perubahan data user ke database
Hasil Uji Coba	Valid

Sumber : Pengujian

21. Kasus Uji Hapus User

Tabel 6.27 Pengujian hapus user

Nama Kasus Uji	Uji Hapus User
Objek Uji	Uji Halaman Hapus User
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan konfirmasi hapus user dan menghapus data user yang dipilih dari database
Prosedur Uji	Menekan tombol hapus user dan konfirmasi ya
Hasil yang Diharapkan	Sistem dapat menampilkan pop up konfirmasi hapus user dan dapat menghapus data user yang dipilih dari database
Hasil Uji Coba	Valid

Sumber : Pengujian

22. Kasus Uji Login

Tabel 6.28 Pengujian login

Nama Kasus Uji	Uji Login
Objek Uji	Uji Halaman Login
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan form login dan dapat memvalidasi proses login. Pengujian ini dilakukan oleh semua pengguna
Prosedur Uji	Mengakses halaman localhost/sipkia, memasukkan username dan password yang benar
Hasil yang Diharapkan	Sistem dapat menampilkan halaman login dan dapat memvalidasi proses login
Hasil Uji Coba	Valid

Sumber : Pengujian

23. Kasus Uji Logout

Tabel 6.29 Pengujian logout

Nama Kasus Uji	Uji Logout
Objek Uji	Uji Halaman Logout
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk keluar dari sistem.
Prosedur Uji	Menekan tombol logout pada bagian pojok kanan atas di semua halaman dan konfirmasi ya
Hasil yang Diharapkan	Sistem dapat memberikan pop up konfirmasi logout dan mengeluarkan user dari sistem dengan menghapus session user.
Hasil Uji Coba	Valid.

6.1.4 Pengujian Pattern

Pengujian pola yang dipakai pada pembangunan SIPKIA ini adalah dengan membuat studi kasus jika sistem akan dikembangkan. Fungsi dari pola sendiri merupakan proses penggunaan ulang code-code yang memiliki kemiripan. Pada pola *factory method* kemungkinan adanya pengembangan adalah bertambahnya *concrete creator* atau *concrete product*. Dalam kasus SIPKIA *concrete creator* dapat berkembang ketika adanya desa yang bertambah dalam sebuah kecamatan. Bertambahnya *concrete creator* mungkin akan terjadi jika dalam kecamatan Tumpang ada penambahan desa atau ada sebuah desa yang pecah menjadi beberapa desa. Hal ini sudah pernah terjadi pada desa Duwet dimana saat ini desa Duwet telah terpecah menjadi desa Duwet Dampul dan Duwet Krajan. Sedangkan pengembangan dari sisi *concrete product* juga sangat memungkinkan karena faktanya register kohort tidak hanya terdiri dari register kohort ibu (antenatal sebelum kelahiran), bayi, balita dan apras melainkan ada sebuah register lagi yaitu register ibu nifas. Perbedaan dari register kohort ibu (antenatal) yang ada dalam SIPKIA dengan register ibu nifas adalah dimana pada register nifas memfokuskan pencatatan medis terkait dengan perkembangan ibu nifas. Sedangkan pada register ibu (antenatal) dalam SIPKIA merupakan catatan register dari seorang ibu yang sedang hamil (antenatal).

Dari paparan diatas kemungkinan terjadinya pengembangan komponen dalam *factory method* tersebut perlu adanya ilustrasi pengujian jika pengembangan tersebut benar-benar terjadi. Studi kasus atau ilustrasi yang dapat dilakukan adalah semisal adanya pengembangan desa yaitu adanya penambahan desa X dalam lingkup kecamatan Tumpang serta penambahan register ibu nifas pada sistem. Dengan adanya pola *factory method* pengembangan sistem tidak merubah banyak code program melainkan hanya menambah sedikit pada code program dengan memanfaatkan design pattern yang telah terbentuk. Penambahan-penambahan tersebut untuk lebih jelasnya dapat dilihat pada Tabel 6.30 berikut,

Tabel 6.30 Ilustrasi penambahan komponen pada sistem

Penambahan Komponen	Alur penambahan code program	Tambahan Code Program
Desa X (Concrete creator)	1. Menambahkan code program pada masing masing fungsi pada kelas C_Bidan untuk mendirect jika desa X dibutuhkan oleh client.	<pre>else if (\$bidan=="desaX") { \$factorydesaX=new BidanDesaX(); \$hasil['data']=\$factorydesaX ->lihatRegister(\$jenis); }</pre>
	2. Membuat kelas BidanDesaX yang extends kelas BidanDesa dan membuat fungsi-fungsinya.	<pre>class BidanDesaX extends BidanDesa{ function lihatRegister(){ // isi fungsi } . . . function lihatPemetaan(){ // isi fungsi } }</pre>
Register Kohort Ibu Nifas (concrete product)	1. Menambahkan code program pada masing masing fungsi pada semua kelas <i>concrete creator</i> seperti kelas BidanDesaKidal dan BidanDesaKambingan untuk mendirect jika Kohort ibu nifas dibutuhkan oleh client.	<pre>function lihatRegister(\$jenis){ switch (\$jenis) { case "nifas": \$register = new RegisterKohortIbuNifas; \$hasil=\$register-> viewRegister("kidal"); return \$hasil; break; } }</pre>
	2. Membuat kelas RegisterKohortIbuNifas yang extends RegisterKohort dan membuat fungsi-fungsinya.	<pre>class RegisterKohortIbuNifas extends RegisterKohort { function viewRegister(){ // isi fungsi } . . . function deletePelayanan(){ // isi fungsi } }</pre>

6.2 Analisis

Analisis dilakukan untuk mendapatkan kesimpulan dari hasil uji yang dilakukan oleh sistem, yakni pengujian unit, pengujian integrasi, pengujian validasi dan pengujian pattern. Proses analisis dilakukan sesuai dengan pengujian yang dilakukan pada sistem yakni analisis hasil uji pengujian unit, analisis hasil uji pengujian integrasi, analisis hasil uji pengujian validasi dan analisis hasil uji pengujian pattern.

6.2.1 Analisis Hasil Uji Pengujian Unit

Dari kasus uji yang telah dilaksanakan sesuai dengan prosedur pengujian basis path yang telah disebutkan dalam sub pokok bahasan 6.1.1 pada algoritma lihat register di kelas C_Bidan dan algoritma tambah register di kelas BidanDesaKidal bahwa seluruh kasus uji memiliki hasil yang telah sesuai dengan yang diharapkan atau bernilai valid.

6.2.2 Analisis Hasil Uji Pengujian Integrasi

Dari dua modul yang diuji dalam pengujian integrasi menunjukkan bahwa dari modul terendah sampai modul utama pada modul lihat register dan tambah register memiliki hasil yang valid. Hal ini dapat dikatakan bahwa pengujian arsitektur yang dilakukan telah berhasil dan dapat dilanjutkan ke pengujian sistem atau pengujian validasi.

6.2.3 Analisis Hasil Uji Pengujian Validasi

Dari kasus uji yang telah dilaksanakan sesuai dengan prosedur pengujian fungsional yang telah disebutkan dalam sub pokok bahasan 6.1.3 maka didapatkan hasil seperti yang ditunjukkan dalam tabel 6.31 berikut.

Tabel 6.31 Hasil pengujian fungsional

No.	Kasus Uji	Hasil yang Didapatkan	Status
1.	Uji Lihat Register	Sistem dapat menampilkan halaman lihat register beserta data register yang diminta yaitu data ibu, bayi, balita atau apras	Valid
2.	Uji Tambah Register	Sistem dapat menampilkan halaman form tambah register dan dapat menyimpan data baru (data ibu, bayi, balita atau apras) ke database	Valid
3.	Uji Ubah Register	Sistem dapat menampilkan halaman form ubah register dan dapat menyimpan perubahan data (ibu, bayi, balita atau apras) ke database	Valid
4.	Uji Hapus Register	Sistem dapat menampilkan pop up konfirmasi hapus register dan dapat menghapus data register (ibu, bayi, balita atau apras) yang dipilih dari database	Valid

Tabel 6.31 Hasil pengujian fungsional (lanjutan)

No.	Kasus Uji	Hasil yang Didapatkan	Status
5.	Uji Lihat Pelayanan	Sistem dapat menampilkan halaman lihat pelayanan beserta data pelayanan yang diminta yaitu data ibu, bayi, balita atau apras	Valid
6.	Uji Tambah Pelayanan	Sistem dapat menampilkan halaman form tambah pelayanan dan dapat menyimpan data baru (data ibu, bayi, balita atau apras) ke database	Valid
7.	Uji Ubah Pelayanan	Sistem dapat menampilkan halaman form ubah pelayanan dan dapat menyimpan perubahan data (ibu, bayi, balita atau apras) ke database	Valid
8.	Uji Hapus Pelayanan	Sistem dapat menampilkan pop up konfirmasi hapus pelayanan dan sistem dapat menghapus data pelayanan (ibu, bayi, balita atau apras) dari database	Valid
9.	Uji Lihat Rekap Desa	Sistem dapat menampilkan halaman rekap desa beserta data cakupan pada tahun tertentu	Valid
10.	Uji Cetak Rekap Desa	Sistem dapat mencetak data rekap desa ke dalam file yang dipilih oleh pengguna	Valid
11.	Uji Lihat Rekap Kecamatan	Sistem dapat menampilkan halaman rekap kecamatan beserta data cakupan pada bulan dan tahun tertentu	Valid
12.	Uji Cetak Rekap Kecamatan	Sistem dapat mencetak data rekap kecamatan ke dalam file yang dipilih oleh pengguna	Valid
13.	Uji Lihat Pemetaan Desa	Sistem dapat menampilkan halaman pemetaan desa beserta data status cakupan dan diagram cakupan pada tahun tertentu	Valid
14.	Uji Cetak Pemetaan Desa	Sistem dapat mencetak data pemetaan desa ke dalam file yang dipilih oleh pengguna	Valid
15.	Uji Lihat Pemetaan Kecamatan	Sistem dapat menampilkan halaman pemetaan kecamatan beserta data status cakupan dan diagram cakupan tahun tertentu	Valid
16.	Uji Cetak Pemetaan Kecamatan	Sistem dapat mencetak data pemetaan kecamatan ke dalam file yang dipilih oleh pengguna	Valid
17.	Uji Lihat Rekap Desa	Sistem dapat memberikan alert bahwa pembaharuan rekap telah dilakukan dan data yang akan ditampilkan berupa data yang telah diperbarui	Valid

Tabel 6.31 Hasil pengujian fungsional (lanjutan)

No.	Kasus Uji	Hasil yang Didapatkan	Status
18.	Uji Lihat User	Sistem dapat menampilkan halaman lihat user beserta data user yang terdaftar	Valid
19.	Uji Tambah User	Sistem dapat menampilkan halaman form tambah user dan dapat menyimpan data baru ke database	Valid
20.	Uji Ubah User	Sistem dapat menampilkan halaman form ubah user dan dapat menyimpan perubahan data user ke database	Valid
21.	Uji Hapus User	Sistem dapat menampilkan pop up konfirmasi hapus user dan dapat menghapus data user yang dipilih dari database	Valid
22.	Uji Login	Sistem dapat menampilkan halaman login dan dapat memvalidasi proses login	Valid
23.	Uji Logout	Sistem dapat memberikan pop up konfirmasi logout dan mengeluarkan user dari sistem dengan menghapus session user.	Valid

Sumber: Pengujian

Proses analisis terhadap hasil pengujian fungsional dilakukan dengan melihat kesesuaian antara fungsi hasil kerja sistem dengan daftar kebutuhan sistem. Pada pengujian satu sampai delapan merupakan gabungan dari pengujian register kohort ibu, bayi, balita dan apras. Oleh karena itu semua fungsional yang diuji 48 fungsional. Berikut merupakan perhitungan nilai hasil pengujian fungsionalitas:

$$\begin{aligned}
 \text{Validasi} &= \frac{\text{Jumlah tindakan yang dilakukan}}{\text{Jumlah tindakan dalam daftar kebutuhan}} \times 100\% \\
 &= \frac{48}{48} \times 100\% \\
 &= 100\%
 \end{aligned}$$

Berdasarkan dari hasil pengujian fungsional di atas maka dapat dianalisis bahwa implementasi dan fungsionalitas sistem telah sesuai dengan daftar kebutuhan sistem yang dijelaskan dalam tahapan analisis kebutuhan sistem.

6.2.4 Analisis Hasil Uji Pengujian Pattern

Dari hasil pengujian pola perancangan *factory method* dapat dilakukan analisis. Dengan adanya design pattern dalam sistem proses pengembangan tidak mengubah code program sistem tetapi menambahkan code program ke sebagian kelas yang telah dibuat. Berdasarkan hal tersebut maka bisa dikatakan bahwa penggunaan pola *factory method* dalam pembangunan SIPKIA memudahkan proses pengembangan sistem bahkan kemudahan tersebut juga dapat dirasakan ketika proses pembangunan dilakukan. Ketika *concrete creator* yang satu selesai maka untuk mengerjakan *concrete creator* yang lain tidak membutuhkan waktu

yang lama karena tinggal mengambil pola dari *concrete creator* yang sudah jadi. Begitu juga ketika *concrete product* satu telah selesai maka untuk membangun *concrete product* yang lainnya tidak membutuhkan waktu yang lama karena polanya telah dikerjakan. Secara keseluruhan pembangunan Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory method* dapat memudahkan dan mempercepat proses pembangunan SIPKIA sehingga target yang direncanakan telah tercapai.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil rekayasa kebutuhan, perancangan, implementasi dan pengujian sistem yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut :

1. Pembangunan Sistem Informasi Register Kohort Bidan Desa dan Pemetaan Kesehatan Ibu dan Anak Menggunakan Pola Perancangan *Factory Method* diawali dengan melakukan wawancara ke bidan desa dan pegawai puskesmas serta observasi langsung ke puskesmas pembantu desa dan puskesmas kecamatan Tumpang. Proses rekayasa kebutuhan diawali dari tahap elisitasi kebutuhan, spesifikasi kebutuhan, validasi dan verifikasi, manajemen kebutuhan dan terakhir pemodelan kebutuhan.
2. Proses perancangan Sistem Informasi Kesehatan Ibu dan Anak Bidan Desa menggunakan konsep pola perancangan (*design pattern*) dan pola yang digunakan adalah pola *factory method* . Selain itu proses perancangan terdiri dari perancangan perancangan arsitektur, pemodelan kelas, pemodelan data ,perancangan komponen, dan perancangan antarmuka pengguna.
3. Sistem diimplementasikan berbasis web dengan menggunakan framework codeigniter dan template bootstrap serta database MySQL. Dengan adanya sistem ini akan mendukung atau bahkan menggantikan buku kohort secara manual dengan administrasi digital guna meningkatkan pelayanan mutu kesehatan ibu dan anak. Selain itu sistem rekap dan pemetaan otomatis dapat mengurangi waktu kerja bidan desa dan pegawai puskesmas yang mana waktu tersebut dapat dialihkan ke kegiatan pelayanan kepada ibu dan anak. Pemetaan sendiri juga dapat digunakan oleh pegawai puskesmas untuk mengetahui desa/kelurahan mana yang paling memerlukan perhatian dan tindak lanjut yang harus dilakukan.
4. Pengujian Sistem Informasi Kesehatan Ibu dan Anak Bidan Desa dilakukan menggunakan pengujian unit, pengujian integrasi, pengujian validasi dan pengujian pola *factory method*. Hasil pengujian validasi menyatakan bahwa implementasi dan fungsionalitas sistem telah sesuai dengan daftar kebutuhan sistem dengan akurasi 100%. Pada pengujian pola *factory method* menyimpulkan bahwa pembangunan sistem menggunakan pola *factory method* dapat memudahkan dan mempercepat proses pembangunan SIPKIA.

7.2 Saran

1. Pengguna dalam sistem untuk mengelola data kohort sebatas bidan desa, hal ini dapat ditingkatkan ke pengguna bidan praktek swasta (BPS), bidan puskesmas dan bidan pada rumah sakit.

2. Perekapan data dari register kohort pada sistem hanya menyediakan rekapan PWS-KIA sehingga perlu pengembangan untuk perekapan imunisasi, perekapan gizi dan ibu nifas.
3. Dapat dilakukan pengembangan combine pattern jika suatu saat memang diperlukan.



DAFTAR PUSTAKA

- Aulia, R.S., 2014. Rancang Bangun Sistem Informasi Pemetaan Sebaran Guru Kabupaten Lumajang. Skripsi sarjana komputer PTIIK Universitas Brawijaya.
- Booch, G., 1991. Object-Oriented Analysis and Design with Application, Benjamin/Cummings.
- Booch, G., Rumbaugh, J., Jacobson, I., 1999. The Unified Modeling Language Reference Manual. Reading Massachusetts: Addison Wesley.
- Dalbey, Dr. John. 2003. Pseudocode Standard. Tersedia di : http://users.csc.calpoly.edu/~jdalbey/SWE/pdl_std.html diakses pada 20 Februari 2016
- Depkes. 2010. Pedoman Pemantauan Wilayah Setempat Kesehatan Ibu dan Anak (PWS-KIA). Jakarta : Kementerian Kesehatan RI Direktorat Jenderal Bina Kesehatan Masyarakat Direktorat Bina Kesehatan Ibu
- Ellis, B., Stylos, J., Myers, B., 2007. The factory pattern in API design: A usability evaluation, in: Proceedings of the 29th International Conference on Software Engineering. IEEE Computer Society, pp. 302–312.
- Freeman, E., Robson, E., Sierra, K., & Bates, B. 2004. Head First design patterns. Sebastopol, CA: O'Reilly.
- Gamma, E., Helm, R., Johnson, R., dan Vlissides, J., 1995. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. ISBN 0-201-63361-2
- Henney, K., 2003. Factory and Disposal Methods: A Complementary and Symmetric Pair of Patterns. Conference: VikingPLOP, Second Nordic Conference on Pattern Languages of Programs, At Bergen, Norway.
- Hill, M.G., 2005. McGraw-Hill Concise Encyclopedia of Science and Technology. New York : McGraw-Hill Digital, pp.1149
- Hovenga, E. S., dan Grain, H., 2013. Health information systems. Studies In Health Technology And Informatics, 193120-140.
- Imbar, R. V., dan Kurniawan, Y., 2012. Perancangan Sistem Informasi Pelayanan Medis Rawat Jalan Poliklinik Kebidanan dan Kandungan pada RSUD Kota Batam. Jurnal Sistem Informasi, 7(1).
- Irawan, Y., 2010. Sistem Informasi Pengolahan Data Pasien Kohort Ibu Pada Bagian Kebidanan di Puskesmas Garuda Bandung. Tugas akhir Diploma Tiga Program Studi Manajemen Informatika. Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia

- Khoiri, A. 2008. Pengembangan Sistem Informasi Posyandu Guna Mendukung Surveilans Kesehatan Ibu & Anak Berbasis Masyarakat Pada Desa Siaga (Studi Kasus Di Kelurahan Manisrejo Kecamatan Taman Kota Madiun Provinsi Jawa Timur). PhD Thesis. program Pascasarjana Universitas Diponegoro.
- Kusumaningrum, D., Kusumo, D. S., Widowati, S., 2008. Implementasi Refactoring untuk Mendapatkan Abstract Factory Design Pattern pada Kode Berbasis Object Oriented. Tugas Akhir Teknik Informatika Universitas Telkom.
- Mellor, S. J., Balcer, M., dan Foreword By-Jacoboson, I., 2013 Executable UML: A foundation for model-driven architectures. Addison-Wesley Longman Publishing Co., Inc..
- Pizam, Abraham. 2005. International Encyclopedia of Hospitality Management. Amsterdam: Elsevier, , pp. 343
- Pressman, R.S., 2010. Software Engineering A Practitioner's Approach. Seventh Edition. New York: The McGraw-Hill Companies, Inc.,
- Rosa,A.S., Shalaahuddin,M., 2014 .Rekayasa Perangkat Lunak. Bandung:Informaika Bandung.
- S. Kaliski, B., 2006. Encyclopedia of Business and Finance, ed. 2nd, Detroit: Gale, pp. 385-388.
- Sommerville, Ian. 2011. Software Engineering. Ninth Edition. Boston: Pearson Education, Inc.,
- Supeno., 2014. Sistem Informasi Rekam Medik Unit Kebidanan dan Kandungan Pada Rumah Sakit Ibu dan Anak Widiyanti Palembang.
- Surya, I. P. A. H., Wibowo, J., dan Amelia, T., 2012. Rancang Bangun Sistem Informasi Rumah Bersalin (Studi Kasus Rumah Bersalin Bidan Ni Wayan Suriati). Jurnal JSIKA, 1(2).

LAMPIRAN A FORM VALIDASI DAN VERIFIKASI

FORM VALIDASI DAN VERIFIKASI			
KEBUTUHAN SIPKIA			
Kode	Kebutuhan Fungsional	Validasi	Verifikasi
FR-01	Melihat data register kohort ibu	✓	✓
FR-02	Menambah data register kohort Ibu	✓	✓
FR-03	Megubah data register kohort ibu	✓	✓
FR-04	Menghapus data register kohort ibu	✓	✓
FR-05	Melihat detail pelayanan kohort ibu	✓	✓
FR-06	Menambah pelayanan kohort ibu	✓	✓
FR-07	Mengubah pelayanan kohort ibu	✓	✓
FR-08	Menghapus pelayanan kohort ibu	✓	✓
FR-09	Melihat data register kohort bayi	✓	✓
FR-10	Menambah data register kohort Bayi	✓	✓
FR-11	Megubah data register kohort bayi	✓	✓
FR-12	Menghapus data register kohort bayi	✓	✓
FR-13	Melihat detail pelayanan kohort bayi	✓	✓
FR-14	Menambah pelayanan kohort bayi	✓	✓
FR-15	Mengubah pelayanan kohort bayi	✓	✓
FR-16	Menghapus pelayanan kohort bayi	✓	✓
FR-17	Melihat data register kohort balita	✓	✓
FR-18	Menambah data register kohort Balita	✓	✓
FR-19	Megubah data register kohort balita	✓	✓
FR-20	Menghapus data register kohort balita	✓	✓
FR-21	Melihat detail pelayanan kohort balita	✓	✓
FR-22	Menambah pelayanan kohort balita	✓	✓
FR-23	Mengubah pelayanan kohort balita	✓	✓
FR-24	Menghapus pelayanan kohort balita	✓	✓
FR-25	Melihat data register kohort apras	✓	✓
FR-26	Menambah data register kohort Apras	✓	✓
FR-27	Megubah data register kohort apras	✓	✓
FR-28	Menghapus data register kohort apras	✓	✓

FR-29	Melihat detail pelayanan kohort apras	✓	✓
FR-30	Menambah pelayanan kohort apras	✓	✓
FR-31	Mengubah pelayanan kohort apras	✓	✓
FR-32	Menghapus pelayanan kohort apras	✓	✓
FR-33	Melihat rekapan PWS-KIA desa	✓	✓
FR-34	Mencetak rekapan PWS-KIA desa	✓	✓
FR-35	Melihat rekapan PWS-KIA kecamatan	✓	✓
FR-36	Mencetak rekapan PWS-KIA kecamatan	✓	✓
FR-37	Melihat pemetaan PWS-KIA desa	✓	✓
FR-38	Mencetak pemetaan PWS-KIA desa	✓	✓
FR-39	Melihat pemetaan PWS-KIA kecamatan	✓	✓
FR-40	Mencetak pemetaan PWS-KIA kecamatan	✓	✓
FR-41	Memperbarui rekapan PWS-KIA	✓	✓
FR-42	Memperbarui rekapan PWS-KIA	✓	✓
FR-43	Melihat user sistem	✓	✓
FR-44	Menambah user sistem	✓	✓
FR-45	Mengedit user sistem	✓	✓
FR-46	Menghapus user sistem	✓	✓
FR-47	Login	✓	✓
FR-48	Logout	✓	✓

Mengetahui Validator dan Verifikator

Bidan Desa Kidal


 Khufahel P. A.

Pengelola KIA Kecamatan


 Sri Suliyani