

BAB II

LANDASAN TEORI

2.1. Health Level 7 version 2.5.1

Health Level 7 (HL7) adalah salah satu dari beberapa standard ANSI (American National Standards Institute), yang telah diakreditasi oleh SDO (Standards Developing Organizations). Standarisasi ini dipakai khususnya untuk bidang atau area *healthcare system*. HL7 menciptakan standard untuk pertukaran, manajemen, dan integrasi informasi kesehatan elektronik untuk tujuan klinis dan administratif. HL7 tidak mengembangkan perangkat lunak, tetapi hanya menyediakan organisasi kesehatan dengan spesifikasi untuk membuat sistem dapat saling bertukar informasi. HL7 merupakan standard pertukaran data medis yang berbentuk teks (Benson, 2010:91).

HL7 version 2.5.1 menggunakan metodologi yang didefinisikan dengan baik berdasarkan informasi data model. HL7 version 2.5.1 menggunakan analisis yang teliti, teknik penulisan, menggabungkan *message* dan format *message* dengan kerumitan lebih sedikit.

Level 7 dari HL7 berasal dari level ketujuh dari *Open System Interconnection* (OSI). Dimana level ketujuh mempunyai fungsi mengkomunikasikan data antara aplikasi satu dengan aplikasi lainnya. Dengan demikian *Health Level 7* memiliki arti komunikasi data kesehatan antar satu aplikasi dengan aplikasi lainnya. (ANSI, 2007)

2.1.1 Sejarah HL7

HL7 *version 2* adalah standard pengiriman data yang digunakan paling besar di dunia. Lebih dari 90% rumah sakit yang ada di USA menggunakan HL7 sebagai standard pengiriman data. HL7 pertama kali dikenalkan pada tahun 1987, dengan nama HL7 *version 1.0*. HL7 *version 1.0* hanya fokus pada pertukaran informasi diantaranya *admissions*, *discharges* dan *transfers* (ADT). Pada tahun 1988 HL7 *version 2.0* mulai dikenalkan, dengan ada penambahan perlakuan pada permintaan pertukaran dan pembuatan laporan, yang berbasis pada standard ASTM (*American Society of Testing and Materials*) E.1238.88. Dan pada tahun 1991, *version 2.1* mulai dikenalkan.

HL7 *version 2* dikembangkan selama lebih dari 20 tahun. Waktu penulisan, versi yang terakhir adalah versi 2.6, dan standard ANSI telah menyetujuinya. Selama pengembangan beberapa periode lingkup HL7 *version 2* mengalami peningkatan besar sekali, tetapi dengan prinsip dasar yang susah untuk dirubah. Standard HL7 *version 2.6* sekarang mempunyai 1.965 lembar dan 717.000 kata. Ini adalah alasan mengapa HL7 *version 2* mengalami perubahan yang besar sekali dalam informasi kesehatan.

Salah satu prinsip dasar HL7 adalah tetap kompatibility dengan versi yang lama, walaupun standard telah dikembangkan terus menerus. Ide itu digunakan dalam pembuatan sistem, sistem yang menggunakan versi baru harus dapat mengerti *message* dari versi lama (Benson T, 2010:93).

2.1.2 Struktur HL7 *Message version 2*

Struktur HL7 *version 2.5.1* terdiri dari *namespace*, *datatype namespace*, *group namespace*, *message namespace* dan *segment namespace*.

A. Namespace

Namespace merupakan kelas-kelas yang digunakan oleh datatype.

Tabel 2.1 Tabel contoh *namespace*

Kelas	Deskripsi
DT	Date
ID	Identifier
IS	ID Set
ST	Short String
TM	Time

B. Datatype Namespace

Datatype merupakan bagian dasar penyusun format HL7 *message version 2.5*. *Datatype* digunakan untuk membangun atau membatasi setiap elemen. *Datatype* memiliki 89 macam data, tapi kebanyakan aplikasi HL7 hanya memakai beberapa macam *Datatype*. (Benson T, 2010:101)

Tabel 2.2 Tabel contoh *datatype*

Kelas	Komponen	Datatype
AD (Address)	Street	ST
	Other Designation	ST
	City	ST
	State or Province	ST
	Zip or Postal Code	ST
	Country	ID
	Address Type	ID
	Other Geographic	ST

Kelas	Komponen	Datatype
	Designation	

C. Segment Namespace

Segment didefinisikan sebagai sebuah tabel yang ada di bawah *segment* MSH (*Message Header*). Semua HL7 *message version 2* dimulai dengan MSH dan ini merupakan contoh bagaimana sebuah pesan dapat didefinisikan. (Benson, 2010:96)

Segment memiliki *field-field*, dan setiap *field* memiliki arti yang berbeda-beda. Pada pembentukan HL7 *message version 2.5.1*, setiap *field* pada *segment* akan menggantikan setiap data pasien yang ada.

Tabel 2.3 Tabel contoh *segment*

Segment	Field	Datatype
ABS	ABS-1: Discharge Care Provider	XCN
	ABS-2: Transfer Medical Service Code	CE
	ABS-3: Severity of Illness Code	CE
	ABS-4: Date/Time of Attestation	TS
	ABS-5: Attested By	XCN
	ABS-6: Triage Code	CE
	ABS-7: Abstract Completion Date/Time	TS
	ABS-8: Abstracted By	XCN
	ABS-9: Case Category Code	CE
	ABS-10: Caesarian Section Indicator	ID
	ABS-11: Gestation Category Code	CE
	ABS-12: Gestation Period - Weeks	NM
	ABS-13: Newborn Code	CE
	ABS-14: Stillborn Indicator	ID
ACC	ACC-1: Accident Date/Time	TS

Segment	Field	Datatype
	ACC-2: Accident Code	CE
	ACC-3: Accident Location	ST
	ACC-4: Auto Accident State	CE
	ACC-5: Accident Job Related Indicator	ID
	ACC-6: Accident Death Indicator	ID
	ACC-7: Entered By	XCN
	ACC-8: Accident Description	ST
	ACC-9: Brought In By	ST
	ACC-10: Police Notified Indicator	ID
	ACC-11: Accident Address	XAD

D. Message Namespace

Message Namespace merupakan struktur *message*. *Message namespace* menggabungkan setiap segmen dari *message*. Setiap kelas dalam *message namespace* memiliki kegunaan masing-masing.

Tabel 2.4 Tabel contoh *message*

Message	Segment	Deskripsi
ADT 01	0 : MSH	Message Header
	1 : SFT	Software Segment (optional repeating)
	2 : EVN	Event Type
	3 : PID	Patient Identification
	4 : PD1	Patient Additional Demographic (optional)
	5 : ROL	Role (optional repeating)
	6 : NK1	Next of Kin / Associated Parties (optional repeating)
	7 : PV1	Patient Visit
	8 : PV2	Patient Visit - Additional Information (optional)
	9 : ROL	Role (optional repeating)
	10 : DB1	Disability (optional)

Message	Segment	Deskripsi
		repeating)
	11 : OBX	Observation/Result (optional repeating)
	12 : AL1	Patient Allergy Information (optional repeating)
	13 : DG1	Diagnosis (optional repeating)
	14 : DRG	Diagnosis Related Group (optional)
	15 : ADT_A01_PROCEDURE	a Group object
	16 : GT1	Guarantor (optional repeating)
	17 : ADT_A01_INSURANCE	a Group object
	18 : ACC	(Accident) optional
	19 : UB1	UB82 (optional)
	20 : UB2	UB92 Data (optional)
	21 : PDA	Patient Death and Autopsy (optional)
ADT 02	0: MSH	Message Header
	1: SFT	Software Segment (optional repeating)
	2: EVN	Event Type
	3: PID	Patient Identification
	4: PD1	Patient Additional Demographic (optional)
	5: ROL	Role (optional repeating)
	6: PV1	Patient Visit
	7: PV2	Patient Visit - Additional Information (optional)
	8: ROL	Role (optional repeating)
	9: DB1	Disability (optional repeating)
	10: OBX	Observation/Result (optional repeating)
	11: PDA	Patient Death and Autopsy (optional)

E. Group Namespace

Group namespace merupakan sebuah grup dari sekumpulan segmen *message* yang dapat diulangi secara bersama-sama atau dipilih untuk dimasukkan atau dikeluarkan secara bersama-sama. (Benson T, 2010:100)

Tabel 2.5 *Group namespace*

Message	Group	Datatype
ABS	ABS-1: Discharge Care Provider	XCN
	ABS-2: Transfer Medical Service Code	CE
	ABS-3: Severity of Illness Code	CE
	ABS-4: Date/Time of Attestation	TS
	ABS-5: Attested By	XCN
	ABS-6: Triage Code	CE
	ABS-7: Abstract Completion Date/Time	TS
	ABS-8: Abstracted By	XCN
	ABS-9: Case Category Code	CE
	ABS-10: Caesarian Section Indicator	ID
	ABS-11: Gestation Category Code	CE
	ABS-12: Gestation Period - Weeks	NM
	ABS-13: Newborn Code	CE
	ABS-14: Stillborn Indicator	ID
ACC	ACC-1: Accident Date/Time	TS
	ACC-2: Accident Code	CE
	ACC-3: Accident Location	ST
	ACC-4: Auto Accident State	CE
	ACC-5: Accident Job Related Indicator	ID
	ACC-6: Accident Death Indicator	ID
	ACC-7: Entered By	XCN
	ACC-8: Accident Description	ST
	ACC-9: Brought In By	ST
	ACC-10: Police Notified Indicator	ID

Message	Group	Datatype
	ACC-11: Accident Address	XAD

F. Delimiters

Untuk menyusun pesan HL7 *version 2.5* dibutuhkan *Delimiters*, yang digunakan untuk memisahkan setiap elemen yang ada.

Tabel 2.6 Simbol *delimiters*

Simbol	Kegunaan
	<i>field separator</i>
^	<i>component separator</i>
~	<i>repetition separator</i>
\	<i>escape character</i>
&	<i>subcomponent separator</i>
<CR>	<i>segment terminator</i>

Field Separator merupakan pembatas antara *field* satu dengan yang lainnya. Namun bila ada *field* yang tidak ada data maka ditulis dengan || . Contohnya pada segmen MSH ada sembilan *field*, maka disana juga harus ada sembilan *field separator* untuk membatasi satu dengan yang lainnya.

Component separator merupakan pemisah komponen dalam satu *field*. Contohnya ADT^01.

Repetition separator merupakan pemisah bila terjadi pengulangan. *Escape character* digunakan untuk menandai di sebuah text bahwa ada perlakuan khusus. *Escape character* dapat digunakan mengirim *delimiters* dengan sebuah

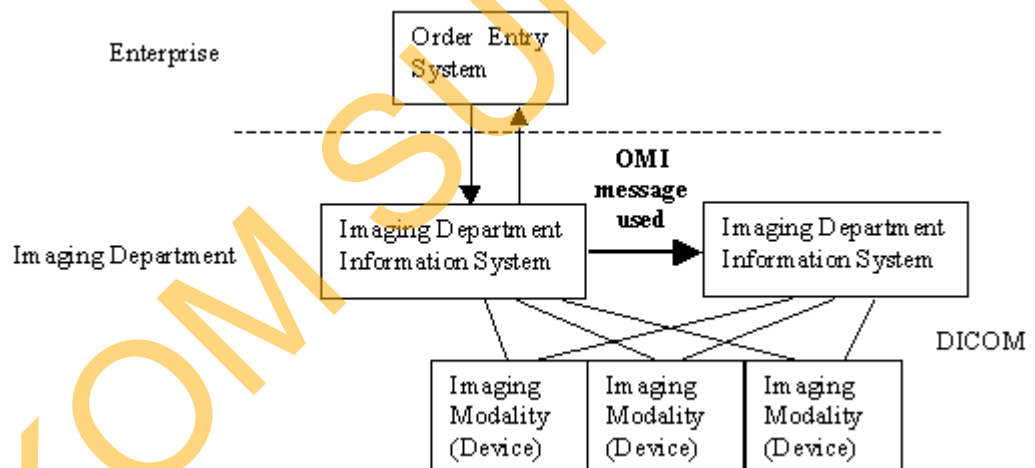
pesan. Dan *escape character* juga dapat digunakan untuk mengindikasikan adanya format. Contohnya seperti `\.br\` mengindikasikan baris berakhir, `\.sp3\` mengindikasikan akan dilompati 3 ruang di format *datatype Formatted Text* (FT).

Subcomponent separator digunakan untuk memisahkan antara sub-komponen dan komponennya.

Segment terminator digunakan untuk mengakhiri segmen yang menggunakan ASCII. (Benson T, 2010:95)

2.2. HL7 Message

2.2.1 OMI^O23 (Imaging Order Message)



Gambar 2.1 Penggunaan OMI Message

HL7 Message ini digunakan dalam komunikasi antar sistem informasi yang terlibat dalam pemenuhan permintaan yang diarahkan ke departemen pencitraan, seperti *Radiologi Information System* (RIS) dan *Picture Archiving and Communication System* (PACS).

Tabel 2.7 Struktur OMI *Message*

Segmen	Description
MSH	<i>Message Header</i> , berisi informasi bagaimana <i>message</i> diproses dan diuraikan. Ini termasuk identifikasi <i>message</i> , pengirim, penerima, tipe <i>message</i> , dll.
PID	<i>Patient Identification</i> , digunakan untuk menyediakan dasar demografi mengenai subyek pengujian
PV1	<i>Patient Visit</i> , berisi informasi kunjungan pasien di setiap pemeriksaan
ORC	<i>Common Order</i> , berisi informasi dasar pengujian setiap <i>order</i> yang diterima. Segmen ini mencakup pengidentifikasi urutan, siapa yang melakukan <i>order</i> , kapan dilakukan <i>order</i> , tindakan apa yang perlu diambil, dll.
OBR	<i>Observation Request</i> , berisi informasi permintaan pemeriksaan. OBR mengidentifikasi tipe pemeriksaan yang dilakukan, dan mengunci informasi <i>order</i> pemeriksaan.
NTE	<i>Notes and Comments</i>

2.2.2 ORU^R01

ORU^R01 dibuat untuk mengembalikan hasil pemeriksaan dari RIS ke HIS.

Tabel 2.8 Struktur ORU *Message*

Segmen	Description
MSH	Message Header, berisi informasi bagaimana <i>message</i> diproses dan diuraikan. Ini termasuk identifikasi <i>message</i> , pengirim, penerima, tipe <i>message</i> , dll.
OBR	Observation Request, berisi informasi permintaan pemeriksaan. OBR mengidentifikasi tipe pemeriksaan yang dilakukan, dan mengunci informasi <i>order</i> pemeriksaan.
OBX	Observation/Result, berisi informasi hasil pemeriksaan yang telah dilakukan. Ini termasuk identifikasi pemeriksaan, hasil pemeriksaan, kapan pemeriksaan dilakukan, dll.

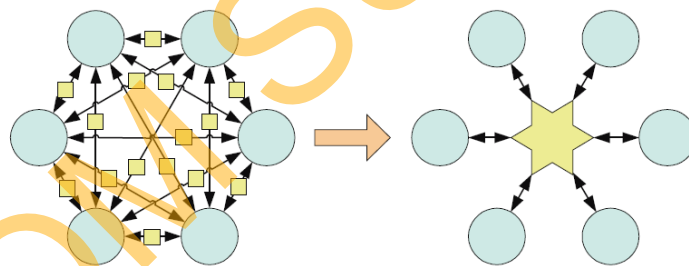
2.3. *Broker* HL7

Broker HL7 adalah sebuah aplikasi yang dirancang sebagai antarmuka antar aplikasi-aplikasi kesehatan yang ada di rumah sakit. *Broker* menyimpan istilah-istilah yang di pakai dalam standard HL7. *Broker* bertugas menerjemahkan setiap data pasien menjadi data HL7. Dan dikirim ke setiap aplikasi rumah sakit yang telah memakai standard HL7. Apabila aplikasi rumah sakit tidak memakai standard HL7, maka *broker* tidak dapat mengirim data ke aplikasi tersebut.

Broker bertugas untuk melakukan seleksi dari setiap data pasien yang diterimanya. *Broker* secara otomatis akan melakukan *disable* pada *message namespace* yang tidak ada datanya. Dan *broker* akan menunjukkan data apa saja

yang dikirim, agar memudahkan dokter dalam memeriksa riwayat pasien. *Broker* juga akan mencatat setiap pesan yang telah dikirim atau diterima. Apabila terjadi *error* pada saat pengiriman atau penerimaan data, maka akan muncul informasi pada *broker*. Dengan adanya informasi ini, admin akan lebih mudah melakukan *tracking* pada *broker*.

Keberhasilan pengiriman data menggunakan standard HL7 sangat bergantung pada *broker*. Oleh sebab itu desain *broker* yang sederhana, akan memudahkan *user* dalam menjalankan aplikasi ini. Dengan adanya *broker*, para dokter tidak perlu ragu apakah data pasien berhasil dikirim atau tidak. Karena pengiriman data akan dipantau terus historinya, dan admin dapat dengan mudah mencari di bagian apa tidak tercapainya pengiriman data. Gagalnya pengiriman dan penerimaan data akan dapat dengan mudah teratasi.



Gambar 2.2 Interface antar aplikasi di rumah sakit

Gambar 2.2 merupakan perbandingan antara rumah sakit yang memakai standard HL7 maupun tidak. Pada rumah sakit yang tidak memakai standard HL7 membutuhkan banyak antar muka agar bisa berkomunikasi satu sama lain. Sedangkan pada rumah sakit yang memakai standard HL7 hanya membutuhkan satu antar muka saja agar bisa berkomunikasi satu dengan lainnya yang disebut dengan *broker* HL7 (Benson, 2010:26).

2.4. Radiology Information System

Radiology Information System (RIS) merupakan sebuah sistem yang dirancang untuk mendukung alur kerja operasional dan analisis dalam suatu departemen radiologi. RIS juga merupakan tempat penyimpanan data pasien dan pelaporan data pasien, dan memberikan kontribusi terhadap catatan data pasien secara elektronik, baik sebagai pendiagnosa suatu penyakit maupun sebagai acuan pemberian arah pengobatan bagi para petugas radiologi dalam sebuah rumah sakit (*The Royal College of Radiologists*, 2008).

RIS adalah penggerak alur kerja departemen radiologi. RIS bertanggung jawab untuk pemesanan penjadwalan, membagikan informasi klinis yang dibutuhkan dalam membuat pemesanan penjadwalan dan menyediakan informasi klinis untuk departemen lain yang memerlukannya, mempersiapkan worklist atau daftar kerja modality, dan menyediakan informasi data yang dibutuhkan *Picture Archiving and Communication System* (PACS) untuk menjalankan perannya.

2.5. PACS

Picture Archiving and Communication System (PACS) adalah filmless dan metode komputerisasi komunikasi dan menyimpan data gambar medis seperti computed radiographic, digital radiographic, computed tomographic, ultrasound, fluoroscopic, magnetic resonance dan foto X-ray (Tong dkk, 2009). Selama lebih dari 100 tahun, efisiensi praktek radiologi telah dibatasi oleh film dan kegiatan penanganan film, dengan adanya PACS memungkinkan gambar radiologi dapat dilihat secara virtual atau elektronik dimanapun pada computer server ataupun computer personal biasa (Dreyer dkk, 2006).

Akusisi citra adalah titik awal data citra masuk ke PACS dari hasil pemeriksaan citra yang dilakukan oleh berbagai modalitas citra digital (seperti CT - Computed Tomography, MR - Magnetic Resonance, PET - Positron Emission Tomography, US - Ultrasound, XA - XRay Angiography, dll).

Saat citra telah diakusisi, PACS akan mengelolanya dengan tepat untuk memastikan penyimpanan, pengambilan, dan pengiriman seluruh citra dapat dilakukan tanpa kesalahan. Selain itu PACS akan menjamin penyimpanan data citra jangka panjang, dan dapat digunakan kapan saja saat dibutuhkan, secara real time, terutama untuk interpretasi citra. Inti PACS terdiri dari: sistem manajemen database relasional (seperti Oracle, MS-SQL, Sybase), media penyimpan (seperti RAID, Jukebox), software pengendali (image manager), dan antarmuka RIS.

Sistem manajemen database adalah jantung dari PACS. Relasi antara citra dan lokasi penyimpanan disimpan dan dikelola di dalam database, berikut dengan semua data terkait yang dibutuhkan untuk pemanfaatan citra. Sistem manajemen database harus dapat menyediakan data citra berdasarkan pada pencarian pasien atau pemeriksaan tertentu saat diminta (*to be queried*) oleh RIS atau sistem lainnya.

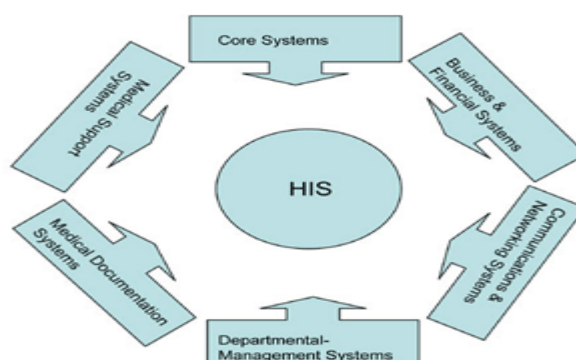
2.6. Hospital Information System

Hospital Information System (HIS) pada dasarnya adalah sebuah sistem informasi yang membantu para penyedia layanan medis dapat mengelola semua informasi secara efektif. HIS ada pertama kali pada tahun 1960 dan dengan seiring waktu mengalami perkembangan. Saat itu HIS masih mengelola penagihan dan persediaan rumah sakit. Sekarang HIS di desain untuk mengatur administrasi,

aspek keuangan dan semua aspek klinis yang ada di rumah sakit (EMR Consultant, 2013).

Rumah sakit yang telah beralih ke HIS memiliki akses informasi yang cepat dan dapat diandalkan termasuk catatan pasien. Catatan pasien yang disimpan dalam HIS lebih detail seperti rekam medis pasien. Keuntungan rumah sakit yang menggunakan HIS adalah (1) memudahkan dokter dalam memproses catatan medis pasien yang bervariasi, (2) membantu pihak yang berwenang di rumah sakit dalam mengembangkan kebijakan, (3) efisien dan akurat dalam mengolah rekam medis pasien, serta (4) meningkatkan pemantauan akan pasien dalam menggunakan obat dan mengurangi redundansi data pasien (EMR Consultant, 2013). Fungsi dari HIS antara lain:

1. Menjadi sistem inti di rumah sakit
2. Menjadi sistem pendukung medis
3. Menjadi sistem dokumentasi medis
4. Menjadi sistem manajemen rumah sakit
5. Menjadi sistem komunikasi dan jaringan di rumah sakit
6. Menjadi sistem bisnis dan finansial di rumah sakit



Gambar 2.3 Fungsi dari HIS

2.7. Post Method

Metode pengiriman data dari *client* ke server yang paling dikenal ada 2 yaitu GET dan POST. GET dan POST dapat mengirim permintaan dan melakukan respon tapi kedua metode ini terdapat perbedaan. Perbedaannya adalah metode GET menggunakan parameter URL, pengiriman data dengan menggunakan metode GET hanya bisa terbatas sedangkan dengan metode POST bisa mengirimkan data ke server dalam jumlah yang besar, metode GET biasanya digunakan untuk menampilkan data (contoh SQL select) dan metode POST digunakan untuk melakukan pembaharuan (contoh SQL insert, SQL update).

Metode POST mengirimkan datanya ke server tidak sebagai bagian dari URL string, tetapi menggunakan bagian dari badan pesan. Metode POST mengirimkan data lebih aman daripada metode GET, karena data yang dikirimkannya tidak terlihat di URL string. Informasi yang sensitif dan bersifat rahasia harus dikirimkan ke server dengan metode POST.

2.8. HAPI V2.5.1

HAPI adalah komponen yang didesain khusus untuk pembuatan HL7 *Message*. HAPI membantu para developer dalam menerjemahkan data-data pasien ke dalam pesan HL7 *version 2* atau ke format XML. HAPI mengembangkan HL7 *Message version 2.1, 2.2, 2.3, 2.3.1, 2.4, 2.5, 2.5.1 dan 2.6*. Komponen HAPI digunakan oleh developer dengan bahasa pemrograman JAVA. HAPI bersifat *open-source*.

2.9. Use Case Diagram

Use case diagram adalah diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *Use case* diagram juga disebut gambaran graphical dari beberapa atau semua aktor, use case dan interaksi diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun. Dalam menggambarkan proses sistem, *use case* diagram menggambarannya dari sudut pandang *user* dan memfokuskan pada proses komputerisasi.

2.10. Fitur-Fitur Broker HL7

A. Communication protocol

Communication protocol yang digunakan *broker* HL7 adalah *Simple Object Access Protocol* (SOAP). SOAP merupakan standar untuk bertukar pesan-pesan berbasis XML melalui jaringan komputer atau sebuah jalan untuk program yang berjalan pada suatu sistem operasi (OS) untuk berkomunikasi dengan program pada OS yang sama maupun berbeda dengan menggunakan HTTP dan XML sebagai mekanisme untuk pertukaran data. SOAP menspesifikasikan secara jelas bagaimana cara untuk meng-*encode header* HTTP dan *file* XML sehingga program pada suatu komputer dapat memanggil program pada komputer lain dan mengirimkan informasi, dan bagaimana program yang dipanggil memberikan tanggapan.

B. Message Tree

Message Tree merupakan salah satu fitur yang ditujukan untuk menghubungkan antara sistem dengan *user*. *Sintaks* HL7 merupakan sederetan baris data pasien yang dirubah menjadi kode yang tidak bisa terbaca oleh *user*.

Oleh sebab itu, fitur pembacaan *sintaks* HL7 sangat dibutuhkan. Hasil *generate sintaks* HL7 disajikan dengan diagram *tree*. Dengan tujuan agar memudahkan user membaca dengan cepat data pasien apa saja yang dikirimkan.

Broker akan membaca *sintaks* HL7 tiap *segment* yang ada di *message*. Satu *segment* akan dirincikan satu per satu *field* nya. Setiap *field* pasti memiliki arti yang berbeda-beda, oleh sebab itu *broker* akan menguraikan HL7 *Message* agar *user* tidak mengalami kesulitan dalam membaca arti *sintaks* HL7.

C. Control Activity

Control Activity merupakan fitur yang berguna bagi seorang admin dalam mengatur jalannya pengiriman dan penerimaan HL7 *Message*. *Error* sering terjadi pada saat pengiriman maupun penerimaan data, dengan adanya fitur ini admin dapat melihat *error* apa yang terjadi. *Error* bisa terjadi di bagian jaringan internet atau *message* yang dikirimkan tidak sesuai dengan aturan yang ada. Hal ini akan bisa di telusuri oleh *control activity* dan melaporkannya ke admin.

D. Log Console

Log Console merupakan fitur yang berfungsi sebagai *log history*. Semua kegiatan pada *broker* akan di *monitoring* dengan fitur ini. Fitur ini akan mencatat semua proses dari seorang admin *login*, mengaktifkan *communication protocol*, melakukan pengiriman atau penerimaan data, berhasil atau gagalanya dalam melakukan pengiriman *message*, dan *log off* dari *broker*.

Broker harus terus dipantau kegiatannya, agar tidak terjadi kesalahan dalam penggunaanya. Fitur ini sangat membantu dalam menganalisa siapa yang mengirim *message* ke sistem lain maupun menerima *message* dari sistem lain.

Dengan tujuan data pasien dapat terjaga dengan baik, dan tidak disalah gunakan oleh pihak-pihak yang tidak bertanggung jawab.

E. Integrasi dengan Sistem

Broker terintegrasi dengan beberapa aplikasi yang ada di rumah sakit, diantaranya PACS, RIS dan sistem informasi lainnya. Setiap perubahan data pasien dan rekam medis pasien yang ada di *broker* harus merubah data yang ada di setiap sistem yang ada di rumah sakit.

Broker merupakan antar muka setiap sistem yang ada di rumah sakit. Data pasien dan rekam medis pasien akan selalu berubah, oleh sebab itu *broker* bertanggung jawab penuh atas perubahan data tersebut. Apabila sebuah sistem terlambat mengetahui perubahan data tersebut, maka akan terjadi kesalahan komunikasi. Sehingga sistem yang lain tidak akan berjalan dengan baik.

F. Read and Write Message

Dengan adanya fitur ini, *broker* mampu membaca dan menulis HL7 *message* yang dibutuhkan sistem informasi di rumah sakit. Dalam fitur *read message*, peran *broker* menjadi pembaca HL7 *Message* yang dikirimkan oleh RIS, PACS, HIS, dll. Sedangkan pada fitur *write message*, peran *broker* sebagai penulis HL7 *message* yang dibutuhkan oleh RIS, PACS, HIS, dll.

G. Setting MSH

Setting MSH merupakan fitur tambahan yang tidak dimiliki oleh semua *broker* HL7 di dunia seperti 7edit, iGuana, HL7Connect, dll. *Setting MSH* digunakan untuk merubah data yang ada di MSH seperti merubah *sender* atau *receiver* HL7 *Message*.

Keuntungan dari fitur ini adalah developer tidak perlu merubah koding setiap melakukan penginstalan *broker* di beberapa rumah sakit. Contoh kasus ketika para developer melakukan penginstalan di Rumah Sakit Adi Husada, MSH HL7 telah ditentukan siapa *sender* dan *receiver*. Ketika aplikasi yang sama diinstal di National Hospital, para developer harus merubah MSH nya. Para developer bisa merubah MSH nya melalui fitur setting MSH. Jadi para developer tidak perlu membongkar kodingnya untuk merubah MSH nya.

STIKOM SURABAYA