**BAB II**

 **KAJIAN PUSTAKA**

**2.1 Penelitian Sebelumnya**

**Penelitian Pertama** dari Hadiansyah, Chandra (2017) yang berjudul “Pembangunan *Server Security Information Management* untuk monitoring keamanan Di *Server* Diskominfo Provinsi Jawa Barat” dijelaskan bahwa untuk menjaga keamanan data elektronik dan informasi, Diskominfo melakukan pemantauan dan langkah antisipasi pencurian data pada *Server* milik instansi terkait agar keamanan data informasi yang bersifat *Confidential* pada seluruh *Server* yang ada di Diskominfo tetap terjaga dari berbagai gangguan yang berasal dari serangan *Attacker* baik dari sisi internal maupun sisi eksternal.

Karena belum tersedianya alat untuk melakukan pemantauan *log*, maka langkah pembangunan sistem **SIEM** sangat dibutuhkan untuk melakukan monitoring server yang masih rentan terhadap serangan. Keuntungan yang diperoleh oleh Diskominfo provinsi Jawa Barat dalam pembangunan **OSSIM** adalah proses monitoring menjadi lebih mudah saat melakukan pengolahan *log* data, serta dapat melakukan pemantauan aktivitas pada komputer atau sistem yang sedang berjalan.

Penggunaan **SIEM** diharapkan dapat membantu tingkat keamanan pada Diskominfo Provinsi Jawa Barat, membantu pemantauan dan monitoring *log*, atau *event* yang terkirim ke *server* **OSSIM**, sehingga pengolahan *log* data, dan pemantauan aktivitas pada komputer atau sistem yang sedang berjalan dapat lebih mudah dilakukan.

**Penelitian Kedua** oleh Syani, Ropi (2018) yang berjudul “Analisis dan

Perancangan *Network Security System* Menggunakan Teknik *Host-Based Intrusion Detection System* (**HIDS**) Berbasis *Cloud Computing*” dijelaskan bahwa *Cloud Computing* merepresentasikan teknologi untuk menggunakan infrastruktur komputasi dengan cara yang lebih efisien, Di sisi lain, arsitektur yang rumit dan terdistribusi semacam itu menjadi target yang menarik bagi para penyusup *Cyber Attacks*.

Metode yang digunakan adalah **HIDS** karena metode tersebut mampu melakukan pemeriksaan sistem yang hanya dapat dilakukan bila aplikasi **IDS** dipasang pada *host*, seperti *file integrity checking, registry monitoring, log analysis, rootkit detection* dan *active response*. Berdasarkan uraian tersebut maka digunakanlah *Cloud Computing* sebagai perantara untuk menganalisis dan mengimplementasikan sistem keamanan jaringan menggunakan teknik *Host-Based Intrusion Detection System* dari **IDS** dengan *software* **OSSEC** sebagai *tools* keamanan jaringannya.

**Penelitian Ketiga** oleh Arass (2019) dengan judul “*Smart* **SIEM**: *From Big Data Logs and Events To Smart Data Alerts*” diperoleh jawaban bahwa pertahanan *Cyber* telah menjadi kebutuhan yang semakin berulang dan bahkan wajib untuk semua jenis organisasi yang memegang sistem informasi, dikarenakan sebuah perusahaan tidak lagi menganggap *cyberattacks* sebagai sebuah kemungkinan, tetapi berubah menjadi ancaman yang pasti akan terjadi, sehingga memerlukan banyak waktu dan berbagai macam cara agar supaya serangan

tersebut dapat diatasi.

Penyedia jasa layanan keamanan jaringan terkemuka, yaitu *Conero* telah mengkonfirmasi bahwa serangan *Distributed Denial Of Service* (**DDOS**) terus meningkat tajam pada setiap tahunnya sebanyak 40%, dengan lebih dari 400 ribu serangan terjadi tiap bulan di tahun 2018 dan terus berlanjut hingga saat ini. Oleh karena itu dibutuhkan sebuah sistem terpusat yang dapat melakukan monitoring aktivitas sistem informasi, dan kemudian dapat digunakan untuk memberikan kesimpulan pada sebuah *event*, apakah aktivitas yang sedang terjadi masuk ke

dalam insiden serangan atau tidak.

**2.2 Penjelasan SIEM**

*Security Information Event Management* (**SIEM**) merupakan sebuah sistem yang memiliki fungsi untuk monitoring atau memantau laju lalu lintas jaringan serta sekaligus memberikan informasi dari sebuah log secara *real-time*. **SIEM** juga merupakan sebuah sistem yang dapat mengumpulkan log-log dari beberapa

tempat seperti aplikasi, server, *database*, *network*, *firewall*, dan beberapa jenisnya.

**2.3 *Security Information Management* (SIM)**

*Security Information Management* (**SIM**) adalah perangkat yang berguna sebagai sarana pengumpulan *File Log*, dan penyimpanan yang dilakukan pada repositori pusat. Perangkat **SIM** digunakan sebagai analisa setelah pemeriksaan mencapai tahap selanjutnya. Oleh karena itu, **SIM** juga disebut sebagai manajemen *Log*. **SIM** sendiri berasal dari sebuah agen dan *software*, yang berjalan pada suatu *server* dan komputer, dan telah di monitor sebelumnya. Pada **SIM**, seorang *Administrator* sistem dapat memasuki konsol dan menjalankan laporan, grafik, dan bagan keamanan secara *Real Time*.

Beberapa sistem **SIM** memiliki *Filter* bawaan yang dapat digunakan untuk melakukan pengawasan, serta membersihkan *Log* sebelum mengirim informasi ke *Server* pusat. Langkah ini dimaksudkan untuk mengurangi jumlah data yang akan dikirim di seluruh jaringan yang dapat menyebabkan kemacetan *Bandwidth* pada jaringan, untuk kemudian disimpan pada *Server* perangkat **SIM**.

**2.4 *Security Event Management* (SEM)**

*Security Event Management* (**SEM**) adalah usaha yang dilakukan untuk mengidentifikasi, mengumpulkan, mengevaluasi, melakukan pemantauan, serta peringatan kepada sistem. Pada makna khusus, **SEM** merupakan versi yang lebih baik dari **SIM*.*** Meskipun keduanya dipandang sebagai bidang manajemen keamanan yang berbeda. Sama seperti **SIM**, data biasanya diteruskan dari komputer *Host* ke repositori pusat menggunakan **SNMP**, *Syslog,* dan protokol komunikasi lainnya. Repository terpusat dapat memastikan bahwa sebuah *Event* dan peringatan akan disimpan dalam penyimpanan yang aman dan ter forensik.

Informasi tersebut kemudian dianalisis, dengan menggunakan algoritma keamanan dan perhitungan statistik untuk mengidentifikasi ancaman, kerentanan, dan risiko. **SEM** dapat mengurai entri secara lebih signifikan, dengan begitu **SEM** dimaksudkan agar dapat segera memberitahu bagian yang bertanggung jawab

setiap kali entri mendapat perhatian.

**2.4.1 *Security Information and Event Management***



**Gambar 2.1 Arsitektur SIEM**

Menurut Maulana Irvansyah (2019) *Security Information and Event Management* atau disingkat sebagai **SIEM** adalah sistem yang membantu pengguna untuk melakukan monitoring trafik pada jaringan dan memberikan informasi secara purna waktu. Selain monitoring trafik pada jaringan, **SIEM** juga dapat melakukan kegiatan monitoring pada setiap perangkat yang ada di dalam jaringan seperti *Server*, perangkat *Network Switch*, *Router*, bahkan sampai perangkat *Security* seperti *Firewall*.

Cara kerja yang dilakukan oleh **SIEM** merupakan tahapan pada aplikasi dan perangkat keamanan yang menghasilkan *Log* untuk setiap aktivitas yang terjadi. Jika sistem pengguna melakukan *Maintenance* karena instalasi *software*, maka hasil tersebut ditampilkan pada **SYSLOG**. Jika perangkat keamanan seperti

*Firewall* pengguna Sebuah log dihasilkan ketika peringatan keamanan terjadi. Demikian pula, semua aplikasi menghasilkan log untuk setiap peristiwa yang

terjadi.

*Log* mentah (*Raw Logs*) yang dikirim sangat sulit untuk dibaca dan dianalisis. **SIEM** adalah alat yang dapat menganalisis *log* mentah dan menampilkan informasi yang dibutuhkan. **SIEM** mengumpulkan *log* dari berbagai aplikasi dan perangkat keamanan dan mengelolanya secara terpusat atau yang biasa dikenal sebagai penyimpanan *log*.

Ukuran setiap *log* tergantung pada tingkat lalu lintas jaringan yang ada. Oleh karena itu, analisis *Big Data* juga memainkan peran penting dalam **SIEM**. Artinya, **SIEM** mengumpulkan semua *log* dari berbagai aplikasi atau perangkat keamanan (*Log Sources*), untuk kemudian dilakukan pengolahan dan analisa pada

*Log* sesuai dengan yang dibutuhkan.

**2.4.2 Zeek**



**Gambar 2.2 Arsitektur Zeek**

*Zeek* adalah sebuah *tools* yang digunakan untuk menganalisa lalu lintas jaringan *open source* yang berbentuk pasif. Fungsi dasar dari *Zeek* sendiri adalah untuk monitoring keamanan, memeriksa semua lalu lintas pada tautan secara mendalam, dan digunakan untuk mengawasi tanda-tanda dari aktivitas yang mencurigakan. Namun, secara lebih umum, Zeek juga mendukung berbagai tugas analisa lalu lintas, bahkan yang berada di luar domain keamanan, termasuk pengukuran suatu kinerja, dan juga membantu untuk menyelesaikan suatu masalah yang terjadi pada lalu lintas jaringan keamanan yang sedang diawasi oleh *Zeek*.

*Zeek* merupakan bagian dari IDS atau Intrusion Detection System yang memiliki tujuan untuk mengawasi, serta mencegah arus lalu lintas mencurigakan yang terjadi pada jaringan pengguna. Adapun secara garis besar, IDS adalah sebuah metode yang biasa digunakan untuk mendeteksi pola serangan di dalam suatu jaringan, untuk kemudian memberitahukan aktivitas tersebut kepada

pengguna melalui pesan secara *real time*.

 IDS dibagi menjadi beberapa bagian seperti Network-Based Intrusion

Detection System (NIDS), dan juga Host-Based Intrusion Detection System (HIDS) yang memiliki fungsi berbeda antar jenis. NIDS lebih mengacu kepada pengawasan terhadap jaringan, sehingga letak dari jenis IDS ini seringkali berada pada *gate*. Sedangkan untuk HIDS, jenis ini lebih sering dipasang pada bagian

server, agar fungsi IDS menjadi maksimal.

Pada *Zeek*, jenis yang digunakan sebagai dasar adalah Network-Based Intrusion Detection System (NIDS). Hal tersebut bisa terjadi karena *Zeek* merupakan sebuah alat yang mampu menganalisa jaringan pada sumber terbuka, maupun pada sumber tertutup yang berjalan pada satu jaringan. *Zeek* mempunyai banyak faktor positif pada sisi analisa, sehingga menjadikan *Zeek* pilihan utama para pengguna, ketika akan mengawasi pergerakan di dalam jaringan.

**2.5 *Software* Pada Elastic Stack**

Penggunaan SIEM sebagai basis penelitian tentu tak terlepas dari perangkat lunak yang dipakai untuk mendukung perangkat SIEM. Beberapa *software* yang

digunakan dijelaskan sebagai berikut:

**2.5.1 Elasticsearch**

*Elasticsearch* adalah mesin pencari dan analisis *Open Source* yang didistribusikan untuk semua jenis data. Termasuk diantaranya adalah teks, numerik, geospasial, data terstruktur, dan data yang tidak terstruktur. *Elasticsearch* dikembangkan di atas *Apache Lucene* dan dirilis pertama kali pada tahun 2010 oleh *Elasticsearch N.V.*, yang sekarang lebih dikenal sebagai *Elastic*. Dikenal dengan **API REST** yang sederhana, sifat terdistribusi, kecepatan, dan skalabilitas, *Elasticsearch* adalah komponen utama dari *Elastic Stack*, satu set alat *Open Source* yang digunakan untuk konsumsi data, pengayaan, penyimpanan, analisis, dan visualisasi. Biasa disebut sebagai Tumpukan *Elasticsearch, Logstash,* dan *Kibana* (**ELK**), *Elastic Stack* saat ini mencakup lebih banyak koleksi agen pengiriman ringan yang dikenal sebagai *Beats*, yang gunanya adalah untuk mengirim data ke *Elasticsearch*. *Elasticsearch* digunakan untuk memantau kecepatan dan skalabilitas untuk mengindeks berbagai jenis konten membuatnya tepat digunakan sebagai:

* Pencarian aplikasi
* Pencarian situs web
* Pencarian perusahaan
* *Logging* dan *log analytics*
* Metrik infrastruktur dan pemantauan wadah
* Pemantauan kinerja aplikasi
* Analisis dan visualisasi data geospasial
* Analisis keamanan
* Analisis bisnis

**2.5.2 Logstash**

*Logstash* adalah mesin yang digunakan untuk pengumpulan data *open source* serta memiliki kemampuan untuk melakukan sejumlah kerja secara *multitasking*. *Logstash* secara dinamis dapat menyatukan data dari sumber yang berbeda dan

menormalkan data ke tujuan yang pengguna pilih.

**2.5.3 Filebeat**

*Filebeat* adalah suatu perangkat pengiriman ringan yang digunakan untuk meneruskan dan memusatkan data log. Diinstal sebagai agen di server pengguna, Filebeat memonitor file log atau lokasi yang telah ditentukan oleh pengguna, mengumpulkan peristiwa *log*, dan meneruskannya ke Elasticsearch atau Logstash untuk dilakukan pengindeksan.

**2.5.4 Kibana**

*Kibana* adalah aplikasi *Front End Open Source* yang kedudukannya berada di atas *Elastic Stack*, *Kibana* menyediakan kemampuan pencarian dan visualisasi data, untuk data yang di index pada *Elasticsearch*. Biasanya *Kibana* lebih dikenal sebagai alat pembuatan bagan untuk *Elastic Stack*, yang sebelumnya disebut sebagai *ELK Stack* setelah *Elasticsearch*, *Logstash*, dan *Kibana*. *Kibana* juga bertugas sebagai *User Interface* untuk memantau, mengelola, dan mengamankan *Cluster Elastic Stack*, serta *Hub* terpusat untuk solusi yang dibuat oleh *Elastic Stack*.

Integrasi ketat pada *Kibana, Elasticsearch,* dan *Elastic Stack* membuatnya

ideal untuk mendukung hal-hal berikut ini:

1. Mencari, melihat, dan memvisualisasikan data yang diindeks dalam *Elasticsearch* dan menganalisis data melalui pembuatan diagram batang, diagram lingkaran, tabel, histogram, dan peta. Tampilan dasbor menggabungkan elemen-elemen visual untuk kemudian dibagikan melalui browser untuk memberikan tampilan analitis *Real Time* ke *Volume* data yang besar untuk

mendukung kasus penggunaan seperti:

1. *Logging* dan *Log Analytics*
2. Metrik infrastruktur dan pemantauan wadah
3. Pemantauan kinerja aplikasi (**APM**)
4. Analisis dan visualisasi data geospasial
5. Analisis keamanan
6. Analisis bisnis
7. Memantau, mengelola, dan mengamankan *Instance Elastic Stack* melalui antarmuka web.
8. Akses terpusat untuk solusi yang dikembangkan pada *Elastic Stack* untuk aplikasi yang dapat diamati, lebih aman, dan kebutuhan untuk pencarian pada perusahaan.
	1. **Signature Based Detection**



**Gambar 2.3 Cara kerja signature based**

Gambar 2.3 menunjukan cara kerja dari metode signature based dalam mendeteksi setiap serangan yang terjadi. Pada pendeteksian setiap serangan, signature based menyimpan pola – pola yang sudah terdapat didalam database dari aplikasi IDS yang sudah dipasang. Rule tersebut digunakan sebagai pembanding jika ada trafik atau paket data yang sama dengan salah satu rule yang telah didefinisikan dalam sistem IDS. Setelah rule dipasang, setiap paket yang melalui jaringan akan diperiksa oleh IDS signature based dengan membandingkan setiap paket data dengan rule yang ada dalam database. Jika paket yang diperiksa ada yang sama dengan pola atau rule yang telah didefinisikan, maka akan muncul pesan deteksi yang ditampilkan sebagai sebuah serangan. Jika tidak ada rule yang sesuai dengan pola yang dibandingkan antara paket data dan rule yang ada, maka

paket data tersebut akan dilewatkan dan tidak dianggap sebagai sebuah serangan.

* 1. **Kibana Query Language**

*Kibana Query Language* (**KQL**) adalah sebuah sintaks sederhana yang digunakan untuk melakukan filter data pada *Elasticsearch* menggunakan metode pencarian teks atau field-based search. **KQL** hanya bisa digunakan untuk melakukan penyaringan data, namun tidak memiliki peran dalam melakukan

penyortiran serta penggabungan data.

**KQL** memiliki berbagai fitur yang membedakan metode Query ini dari *Lucene Query Syntax*. Hal itu bisa terjadi karena **KQL** dapat memberikan pertanyaan pada bidang yang masuk ke dalam sarang, juga pada bidang yang mengandung *Script*. KQL juga tidak mendukung ekspresi reguler atau pencarian dengan istilah *Fuzzy*.

* 1. **Jenis-Jenis Serangan Terhadap Situs Web (Owasp Top 10)**

*OWASP Top 10* merupakan sebuah panduan bagi para *developers* dan *security team* mengenai kelemahan-kelemahan pada web yang rentan diserang dan harus segera disiasati. Berbagai kelemahan ini memudahkan penyusup untuk menanamkan *malware*, mencuri data, atau mengambil alih sepenuhnya situs atau komputer milik *user*. Berikut ini penjelasan singkat dari kesepuluh ancaman keamanan situs yang ada pada *OWASP Top 10*:

1. Injection

[Maurice](https://kemptechnologies.com/blog/author/maurice/) [McMullin](https://kemptechnologies.com/blog/author/maurice/) (2018) *Injection* adalah Serangan injeksi terjadi

ketika data yang tidak dipercaya disuntikkan atau dimasukkan secara manual ke input yang dikirim ke aplikasi, OS atau *database*. Data yang tidak dipercaya berisi kode berbahaya atau parameter input yang dijalankan oleh aplikasi target. Masukan berbahaya ini sering kali memungkinkan akses ke data yang tidak boleh ditampilkan, atau dapat memanggil kode eksternal yang dapat digunakan

penyerang untuk menyusupi aplikasi web dengan cara tambahan.

1. Broken Authentication

Menurut Situmeang, Prahadi, dan Christianto (2018) *Broken Authentication* dan *Session Management* merupakan salah satu dari beberapa serangan yang dapat dilakukan oleh seseorang yang berniat jahat, biasa disebut dengan istilah *hijackers*, dengan memanfaatkan beberapa kelemahan pada fungsi *authentication* dan *session management*.

1. Sensitive Data Exposure

Menurut Chandra, Simanjuntak, dan Fadillah (2018) *sensitive data* merupakan informasi yang harus dilindungi dari penyerang, seperti *password*, alamat, *social security*, *credit card*, informasi bank, rekam kesehatan, dan lain sebagainya. Sedangkan pengertian *data exposure* adalah keadaan dimana data atau informasi tidak dilindungi dengan baik atau terekspose sehingga penyerang berkesempatan untuk mengeksploitasi dan mencuri data.

1. XML External Entities

[Ian](https://www.acunetix.com/blog/author/ianmuscat/) [Muscat](https://www.acunetix.com/blog/author/ianmuscat/) (2018) *XML External Entities* adalah serangan ke situs

web dan aplikasi yang menganalisa input **XML**. *Input* ini bisa mereferensikan *entity external* untuk mengetahui kelemahan yang ada pada input **XML** nya. *Entiti external* yang dimaksud disini biasanya berupa unit penyimpanan, seperti misalnya *hard drive*. Analisa input **XML** bisa dibuat seakan-akan mereka mengirim data ke *entity external* yang tidak dipercaya, dimana mereka bisa mengirim data-data sensitif ke *hacker* nya langsung. Cara terbaik untuk mengatasi **XEE** ini adalah dengan memiliki *web app* yang mengandung jenis data yang tidak

terlalu kompleks.

1. Broken Access Control

Menurut Nugraha (2017*) broken access control* atau lebih sering disebut sebagai otorisasi, adalah suatu proses bagaimana web aplikasi memberikan akses fungsi ataupun konten kepada beberapa user, dan tidak kepada user yang lain. Proses ini terjadi setelah autentifikasi atau lebih umum dikenal dengan proses

*login*.

1. Cross Site Scripting

Menurut Syaikhoni, Ariyadi, dan Sunarna (2018) *Cross Site Scripting* menggunakan aplikasi web untuk mengantarkan kode/skrip berbahaya ke *browser* dari korban karena penyerang tidak bisa langsung menjalankan kode/*script* berbahaya di *browser* korban.

1. Insecure Deserialization

Borislav Kiprin (2019) Sistem modern yang kompleks sudah

terdistribusi. Karena komponen komunikasi antara satu dengan yang lain dan pembagian informasi, seperti memindahkan data antar layanan, menyimpan informasi, format biner asli sudah menjadi tidak ideal. Singkatnya, serialisasi adalah proses mengubah data biner ini menjadi *string* atau biasa disebut dengan karakter **ASCII**, sehingga dapat dipindahkan menggunakan protokol standar.

1. Using Components With Known Vulnerabilities

Menurut Febrianto, Ahnaf, dan Simon (2018) Kerentanan suatu komponen yang masuk ke dalam celah “*Vulnerability*” ini disebabkan oleh pihak ketiga dari komponen tersebut yang menyerang *library* suatu aplikasi baik dari *web base* ataupun *platform* lainnya. Sehingga komponen pihak ketiga ini dengan

kerentanan/celah yang diketahui bisa menyerang sistem operasi itu sendiri, seperti *framework* yang telah disediakan (**CMS**), *web server* dan beberapa *plugin* atau komponen yang diketahui oleh *developer* yang telah terinstal atau telah masuk

dalam *library* tersebut di dalam sistem aplikasi.

1. Insufficient Logging and Monitoring

Menurut Borislav Kiprin (2018) Kebanyakan *web app* tidak mengambil langkah yang cukup untuk mendeteksi penembusan data. Rata-rata orang baru sadar kalau terjadi penembusan di situs web mereka setelah 200 hari. Ini tentunya memberikan penyerang banyak waktu untuk melakukan penyerangan. **OWASP** merekomendasikan *developer* untuk merancang *logging and monitoring* serta

rencana *response insiden* agar mereka tahu jika ada penyerangan yang terjadi pada aplikasi mereka.

1. Security Misconfiguration

Menurut Admir Dizdar (2019) *Security Misconfiguration* dapat diartikan sebagai kegagalan menerapkan semua kontrol keamanan untuk *server* atau aplikasi web, sistem mengimplementasikan sebuah kontrol keamanan, tetapi berjalan dengan suatu kesalahan. *Security Misconfiguration* adalah kelemahan yang paling sering terjadi di antara kelemahan lain. Biasanya kesalahan terjadi jika *user* hanya menggunakan *default* konfigurasi tanpa melihat kebutuhan situs web.

* 1. **Tingkat Ancaman Pada Owasp Top 10 2017**

Pada Tabel 2.1 terlihat ringkasan dari 10 risiko keamanan aplikasi teratas pada tahun 2017 dan faktor risiko yang telah ditetapkan untuk setiap risikonya.

Faktor-faktor ini ditentukan berdasarkan statistik yang tersedia pada ***OWASP Top*** 10.

Tingkat ancaman pada Owasp Top 10 2017 terlihat pada Tabel 2.1 berikut

**Tabel 2.1 Tingkat Ancaman**



* 1. **Flowchart**

*Flowchart* menurut Indrajani (2011) merupakan gambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. *Flowchart* digunakan sebagai alat bantu komunikasi dan untuk menggambar alur kerja dari sebuah sistem. Maka dari itu ketika akan membuat sebuah sistem jelas seorang desainer harus merancang *Flowchart* terlebih dahulu agar apa yang akan dikerjakan bisa tertata dengan baik, selain itu dengan adanya *Flowchart* ini dapat membantu orang awam untuk memahami bagaimana cara kerja sebuah sistem yang akan dibuat.

Berikut ini adalah beberapa penjelasan mengenai simbol-simbol *Flowchart* yang sudah sering digunakan dan sesuai dengan standar internasional.

: *Flow* - Digunakan untuk menggambarkan arah atau alur



dari sistem.

: *Terminator* - Digunakan untuk memulai dan mengakhiri  *Flowchart.*

: *Process* - Digunakan untuk menjelaskan proses atau cara

  kerja sistem.

: *Decision* - Digunakan untuk menjelaskan sebuah situasi

  pada sistem.

 : *On-Page Reference/Connector* - Digunakan untuk

  menghubung alur yang masih berada di halaman yang sama.

: *Off-Page Reference/Off-Page Connector* - Digunakan  untuk menghubungkan alur yang sudah berada di halaman

yang berbeda.

=: *Data* - Digunakan untuk menampilkan *input* dan *output*.

: *Document* - Digunakan untuk menulis dokumen atau

laporan

**2.11 Topologi Jaringan**

Topologi jaringan komputer merupakan susunan geometric dari node dan link kabel dalam *Local Area Network* (LAN).Topologi jaringan komputer mengacu pada bentuk atau tata letak jaringan. Topologi jaringan komputer menggambarkan bagaimana komunikasi ditentukan oleh topologi jaringan komputer. Jaringan komputer terdiri dari beberapa komputer yang terhubung menggunakan beberapa jenis *interface*, masing-masing memiliki satu atau lebih perangkat *interface* seperti *Network Interface Card* (NIC) atau perangkat serial untuk jaringan point to point (PPP). Setiap komputer didukung oleh perangkat jaringan yang menyediakan fungsionalitas server atau klien. Perangkat keras yang digunakan untuk mengirim data melalui jaringan disebut media. Ini termasuk kabel tembaga, serat optik, atau transmisi nirkabel. Pengkabelan standar yang digunakan untuk tujuan ini adalah 10 base-T kategori 5 kabel ethernet.

Ada 2 jenis utama dari kategori jaringan yaitu berbasis peer to peer dan server. Dalam jaringan berbasis server komputer yang menjadi penyedia layanan utama seperti layanan file atau layanan email. Komputer yang menyediakan layanan disebut server dan komputer yang meminta dan menggunakan layanan disebut klien sedangkan, jaringan peer to peer, berbagai komputer pada jaringan dapat bertindak sebagai klien dan server. Sistem operasi jaringan menggunakan protokol data jaringan untuk berkomunikasi di jaringan komputer lain.

**2.12 Slack Messenger**

Slack adalah nama dari sebuah perusahaan *enterprise* yang memberikan pelayanan jasa kepada pengguna berupa suatu aplikasi berbentuk *workplace* atau *cloud working* yang dapat memungkinkan banyak orang untuk bekerja secara bersamaan dalam satu alur. Slack mempunyai banyak fitur yang berguna untuk mengirimkan pesan, file, dan juga data kepada sesama pengguna yang berada di dalam satu *workspace*.

Slack juga memiliki 2 tipe pesan utama yang bisa digunakan, ada *Channels* yang berguna untuk mengirimkan pesan kepada beberapa pengguna dalam satu waktu, dan ada *Direct Message* yang digunakan untuk mengirimkan pesan secara

lebih personal dari pengguna ke pengguna lain.

*Channels* yang ada di dalam Slack bisa diatur menjadi *public* atau *private used*. Ketika menggunakan *Channels* sebagai *public used*, maka siapapun dapat melihat dan masuk ke dalam *workspace*. Sedangkan *channels* yang diatur menjadi *private used*, hanya dapat digunakan oleh seseorang yang memiliki kode undangan dan bersifat terbatas yaitu *Channels* hanya bisa menampung paling

banyak 8 orang saja.

Keunggulan dari Slack adalah penggunaan *User Interface* pada Slack lebih profesional jika dibandingkan dengan para kompetitornya. Slack memiliki banyak fitur yang dapat digunakan pada versi gratis maupun berbayar, dan juga memiliki kontrol penuh terhadap siapa yang bisa masuk ke dalam workspace maupun siapa

yang tidak.