Muhammad Adha Ridwan - 13523098

1 Jaringan Komputer

https://youtube.com/@samekosaba

1.1 Digital Encoding Methods (2.5 points)

Terdapat beberapa metode untuk melakukan digital encoding, yaitu perubahan informasi dalam bentuk bit menjadi bentuk sinyal yang dapat disalurkan dalam jaringan, masing-masing dengan kelebihan dan kekurangannya.

Untuk masing-masing metode berikut, jelaskan bagaimana skema encoding bekerja, apa kelebihannya dibanding skema sebelumnya, dan apa kekurangan baru yang diperkenalkan skema tersebut.

1.1.1 NRZ (Non-return to zero) (0.5 points)

Answer:

Non-return to zero merupakan metode encoding tersimple, dimana:

- Bit '1' direpresentasikan dengan tegangan positif (high voltage).
- Bit '0' direpresentasikan dengan tegangan negatif (low voltage).
- Sinyal tidak kembali ke nol selama durasi bit.

Kelebihan:

- Implementasi sangat sederhana dan murah
- Efisiensi bandwidth tinggi (1 bit per symbol)
- Konsumsi daya rendah
- Mudah untuk di-generate dan di-decode

Kekurangan:

- Tidak ada sinkronisasi clock built-in
- Baseline wander (drift DC) pada sequence panjang bit yang sama
- Tidak dapat mendeteksi error
- Sulit untuk clock recovery pada receiver
- Rentan terhadap noise dan interferensi

Contoh: 1 0 1 0 1 1 0

Figure 1: NRZ Visualization & Example (Sumber Pribadi)

1.1.2 NRZI (Non-return to zero inverted) (0.5 points)

Answer:

Non return to zero inverted menggunakan transisi sinyal untuk merepresentasikan data:

- Bit '1' direpresentasikan dengan perubahan level sinyal (transisi)
- Bit '0' direpresentasikan dengan tidak ada perubahan level sinyal
- Level awal bisa high atau low

Kelebihan dibanding NRZ:

- Lebih toleran terhadap polarity inversion
- Mengurangi masalah baseline wander untuk data dengan banyak bit '1'
- Differential encoding memberikan noise immunity yang lebih baik

Kekurangan:

- Masih tidak ada clock recovery untuk sequence panjang bit '0'
- Masalah sinkronisasi tetap ada pada long runs of zeros
- Kompleksitas encoding/decoding sedikit meningkat
- Tetap rentan terhadap DC drift pada sequence bit '0' yang panjang

Contoh: 1 0 1 0 1 1 0

Figure 2: NRZI Visualization & Example (Sumber Pribadi)

1.1.3 Manchester (0.5 points)

Answer:

Manchester encoding menggunakan transisi di tengah setiap bit period:

- Bit '1' direpresentasikan dengan transisi low-to-high di tengah bit
- Bit '0' direpresentasikan dengan transisi high-to-low di tengah bit
- Ada transisi di setiap bit period, memberikan clock signal

Kelebihan:

- Lebih immune terhadap polarity inversion

- Tetap memiliki self-synchronizing capability
- Better noise immunity karena differential nature
- Dapat bekerja dengan kabel yang terpasang terbalik

Kekurangan:

- Kompleksitas encoding/decoding paling tinggi di antara metode sebelumnya
- Masih membutuhkan bandwidth 2x
- Implementasi hardware/software lebih rumit
- Delay processing lebih tinggi

Contoh: 1 0 1 0 1 1 0

Figure 3: Manchester Visualization & Example (Koleksi)

1.1.4 Differential Manchester (0.5 points)

Answer:

Merupakan gabungan dari differential dan manchester encoding:

- Selalu ada transisi di tengah setiap bit (untuk clock)
- Bit '1' direpresentasikan dengan NO transisi di awal bit period
- Bit '0' direpresentasikan dengan transisi di awal bit period

Kelebihan:

- Lebih immune terhadap polarity inversion
- Tetap memiliki self-synchronizing capability
- Better noise immunity karena differential nature
- Dapat bekerja dengan kabel yang terpasang terbalik

Kekurangan:

- Kompleksitas encoding/decoding paling tinggi di antara metode sebelumnya
- Masih membutuhkan bandwidth 2x
- Implementasi hardware/software lebih rumit
- Delay processing lebih tinggi

Contoh: 1 0 1 0 1 1 0

Figure 4: Differential Manchester Visualization % Example (Sumber Pribadi)

1.1.5 4B/5B (0.5 points)

Answer:

4B/54 adalah block coding scheme:

- Setiap 4 bit data di-map ke 5 bit code
- Code dipilih sedemikian rupa sehingga tidak ada lebih dari 3 consecutive zeros
- 5-bit codes kemudian di-transmit menggunakan NRZI
- Hanya 16 dari 32 possible 5-bit patterns yang digunakan untuk data

Kelebihan:

- Efisiensi bandwidth lebih baik
- Guaranteed clock transitions (max 3 consecutive bits sama)
- Error detection capability melalui invalid code detection
- DC balance yang baik
- Compatible dengan existing NRZI infrastructure

Kekurangan:

- Membutuhkan lookup table untuk encoding/decoding
- Overhead 25% (4 bit menjadi 5 bit)
- Kompleksitas implementasi karena butuh buffering dan table lookup
- Delay karena harus menunggu 4 bit sebelum encoding
- Membutuhkan memory untuk code table

Contoh: 1 0 1 0 1 1 0

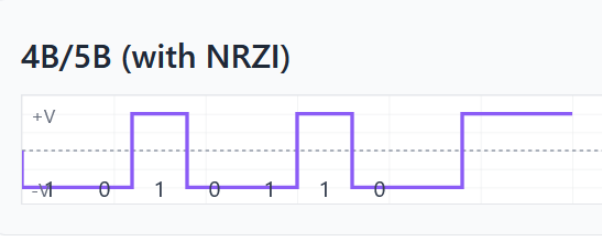

Figure 5: 4B/5B Visualization & Example (Sumber Pribadi)

Bonus: (0.1 poin/subsoal) Berikan contoh untuk masing-masing metode.

Bonus 2: (0.4 poin/subsoal) Berikan visualisasi untuk masing-masing contoh. Visualisasi dibuat sendiri.

1.2 Data Link Layer Transmission (2.5 points)

Jawablah pertanyaan-pertanyaan terkait transmisi pada data link layer berikut.

1.2.1 Ethernet Collision Problem (1 point)

Pada protokol Ethernet, terdapat sebuah masalah yang sangat umum terjadi, yaitu ketika terjadi collision pada transmisi data. Mengapa fenomena ini menjadi masalah? Lalu jelaskan cara kerja algoritma yang digunakan protokol Ethernet untuk mengatasi masalah tersebut!

Answer:

Collision pada transmisi data menjadi masalah karena:

- Kerusakan Data - Ketika dua atau lebih perangkat mengirim data secara bersamaan pada medium yang sama (shared medium), sinyal-sinyal tersebut akan saling bertabrakan dan dapat menghasilkan sinyal yang rusak. Sinyal yang rusak ini tidak dapat diinterpretasi dengan benar oleh penerima.
- Pemborosan Bandwith - Collision menyebabkan waktu transmisi terbuang sia-sia karena data yang dikirim harus diulang kembali. Semakin sering terjadi collision, semakin banyak bandwidth yang terbuang karena terus mengirim ulang.
- Penurunan Throughput Jaringan - Dengan adanya collision yang berulang, throughput jaringan secara keseluruhan akan menurun drastis, terutama ketika jumlah perangkat dalam jaringan bertambah.
- Ketidakpastian Waktu Pengiriman - Collision membuat waktu pengiriman data menjadi tidak dapat diprediksi karena adanya delay akibat retransmission.

Protokol yang digunakan adalah CSMA/CD (Carrier Sense Muktiple Access with Collision Detection

Cara Kerja:

1. Carrier Sense
    - Sebelum mengirim data, perangkat akan "mendengarkan" apakah ada transmisi lain yang sedang berlangsung di medium
    - Jika medium sedang sibuk, perangkat akan menunggu sampai medium kosong
2. Multiple Access
    - Ketika medium terdeteksi kosong, perangkat dapat mulai mengirim data
    - Namun, karena ada delay propagasi, dua perangkat bisa saja mulai mengirim secara bersamaan

3. Collision Detection

   - Selama proses pengiriman, perangkat terus memantau medium untuk mendeteksi collision
   - Collision terdeteksi ketika sinyal yang diterima berbeda dengan sinyal yang dikirim

4. Jam Signal and Stopping

   - Ketika collision terdeteksi, perangkat akan mengirim "jam signal" (32-bit pola khusus) untuk memastikan semua perangkat mengetahui adanya collision
   - Semua perangkat yang terlibat akan menghentikan transmisi

5. Binary Exponential Backoff

   - Setelah collision, perangkat akan menunggu selama periode random sebelum mencoba mengirim ulang
   - Waktu tunggu dihitung dengan rumus:

$$\text{Waktu Tunggu} = \text{random}(0, 2^n - 1) \times \text{Slot Time}$$

   - n adalah jumlah tabrakan (collision) yang telah terjadi untuk frame yang sama, dengan nilai maksimum $n = 10$.
   - Jika collision terjadi sebanyak 16 kali berturut-turut, maka frame akan dianggap gagal dan akan di-drop (gagal dikirim).

1.2.2 Wireless Network Collision (0.5 points)

Mengapa protokol jaringan wireless tidak bisa menggunakan metode yang sama dengan Ethernet untuk mengatasi masalah collision tersebut?

Answer:

Protokol jaringan wireless tidak bisa menggunakan CSMA/CD (seperti Ethernet) karena beberapa keterbatasan:

1. Ketidakmampuan CD yang Reliable

   Problem:

   - Pada wireless, perangkat tidak dapat secara reliable mendeteksi collision karena sinyal yang diterima jauh lebih lemah dibanding sinyal yang ditransmisikan
   - Radio transceiver umumnya tidak dapat mengirim dan menerima secara bersamaan pada frekuensi yang sama (half-duplex nature)
   - Power level sinyal transmit jauh lebih besar daripada sinyal receive, sehingga collision sulit dideteksi

2. Hidden Note Problem

   Skenario Hidden Note: A ←-¿ B ←-¿ C

   - Node A dan C berada di luar jangkauan satu sama lain
   - Keduanya dapat berkomunikasi dengan B

- A tidak dapat mendeteksi transmisi dari C, begitu juga sebaliknya
- Ketika A dan C mengirim data ke B secara bersamaan, collision terjadi di B
- Namun A dan C tidak mendeteksi adanya collision karena mereka tidak dapat "mendengar" satu sama lain

3. Exposed Note Problem: A ←-¿ B C ←-¿ D
   - B ingin mengirim ke A, dan C ingin mengirim ke D
   - C dapat mendengar transmisi B, sehingga C akan menunda transmisinya
   - Padahal transmisi C ke D tidak akan mengganggu transmisi B ke A
   - Menyebabkan penggunaan medium yang tidak efisien
4. Karakteristik Medium Wireless
   - Sinyal wireless mengalami fading, multipath, dan interferensi
   - Kekuatan sinyal bervariasi berdasarkan jarak, obstacles, dan kondisi lingkungan
   - Carrier sensing menjadi tidak reliable karena sinyal lemah bisa jadi bukan karena medium kosong, tapi karena jarak atau obstacle

1.2.3 Wireless Collision Avoidance (1 point)

Jelaskan bagaimana skema alternatif yang digunakan jaringan wireless bekerja untuk mengatasi terjadinya collision.

Answer:

1. Algoritma Dasar CSMA/CA
   - Physical Carrier Sensing

```
if (medium == BUSY) {
    wait until medium becomes IDLE;
}
wait for DIFS period;
```

   - Virtual Carreir Sensing

```
if (NAV > current_time) {
defer transmission;
wait until NAV expires;
}
```

   - Random Backoff

```
if (first_transmission) {
    backoff_time = random(0, CW_min)    slot_time;
} else {
    CW = min(CW    2, CW_max);  // Binary exponential
       backoff
    backoff_time = random(0, CW)    slot_time;
}
```

- Transmission

```
while (backoff_time > 0) {
    if (medium == BUSY || NAV > current_time) {
        freeze backoff counter;
        goto Step 1;
    }
    decrement backoff_time;
}
transmit_frame();
```

2. RTS/CTS (4- Way Handshake)

   Algoritma RTS/CTS:

   - RTS Transmission

```
Sender:
- Wait for DIFS
- Send RTS(duration, receiver_address)
- Set timeout timer for CTS_timeout
```

   - CTS Response

```
Receiver:
- Receive RTS
- Wait for SIFS
- Send CTS(duration - RTS_time - SIFS)
- Set NAV for data reception
```

   - Data Transmission

```
Sender:
- Receive CTS within timeout
- Wait for SIFS
- Send DATA frame
- Set timeout for ACK_timeout
```

   - ACK Response

```
Receiver:
- Receive DATA correctly
- Wait for SIFS
- Send ACK
```

   - NAV (Network Allocation Vector) Update:

```
Nodes overhearing RTS: NAV = CTS_timeout + DATA_time +
   SIFS + ACK_time
Nodes overhearing CTS: NAV = DATA_time + SIFS + ACK_time
```

3. Binary Exponential Backoff

```
class BackoffAlgorithm:
    def __init__(self):
        self.CW_min = 15       # Minimum contention window
        self.CW_max = 1023     # Maximum contention window
        self.retry_limit = 7   # Maximum retry attempts
        self.slot_time = 9     # microseconds (802.11g)

    def calculate_backoff(self, retry_count):
        if retry_count == 0:
            CW = self.CW_min
        else:
            CW = min(self.CW_min * (2  retry_count), self.
               CW_max)

        backoff_slots = random.randint(0, CW)
        return backoff_slots * self.slot_time

    def transmission_attempt(self, retry_count):
        backoff_time = self.calculate_backoff(retry_count)

        # Wait for backoff period
        while backoff_time > 0:
            if self.medium_busy() or self.nav_active():
                # Freeze backoff counter
                return False  # Defer transmission
            backoff_time -= self.slot_time

        return True  # Proceed with transmission
```

4. Frame Fragmentation dan Burst Transmission

- Fragmentation Algorithm:

```
if (frame_size > fragmentation_threshold) {
    fragments = split_frame(frame, fragment_size);
    for each fragment in fragments:
        send_fragment_with_ack();
        if (ack_timeout) {
            retransmit_fragment();
        }
}
```

- Frame Bursting

```
after_successful_transmission() {
    if (more_frames_to_send && remaining_TXOP >
       frame_time) {
        wait_SIFS();
        send_next_frame();
    }
}
```

Integrasi Semua Mekanisme:
Semua mekanisme ini bekerja secara bersamaan dan saling melengkapi:

- CSMA/CA + Backoff -¿ Basic channel access
- RTS/CTS -¿ Hidden node solution untuk frame besar
- Fragmentation -¿ Error recovery optimization

Bonus: Jawaban dengan lebih detail terkait algoritma yang digunakan berpotensi diberikan poin tambahan.

1.3 Transport Layer Transmission (3 points)

Jawablah pertanyaan-pertanyaan terkait transmisi pada transport layer berikut.

1.3.1 End-to-End Address Information (1.5 points)

Apa saja informasi yang dibutuhkan untuk menentukan secara unik alamat tujuan pengiriman paket pada transmisi data End-to-End? Kaitkan dengan penjelasan mengapa protokol UDP disebut sebagai "Simple Demultiplexer" (Read: Computer Networks: A Systems Approach, Chapter 5: End-to-End Protocols)

Answer:
informasi yang dibutuhkan:

- Alamat IP Tujuan (Destination)
- Port Number Tujuan

Transmisi end-to-end, terutama pada transport layer, berfokus pada komunikasi proses-ke-proses (process-to-process communication), bukan hanya host-ke-host. Mengingat ada banyak proses yang berjalan pada satu host, protokol transport perlu menambahkan mekanisme demultiplexing untuk mengarahkan paket ke proses aplikasi yang tepat di host tujuan. Demultiplexing adalah proses dimana dipenerima untuk mengarahkan paket yang masuk dari network layer ke soket atau antrian aplikasi di transport layer. UDP disebut simple karena melakukan demultiplexing hanya berdasarkan informasi yang tersedia pada Headernya yaitu Port Number Tujuan. Sehingga, UDP menyederhanakan tugas bagi aplikasi dengan menyediakan "saluran komunikasi proses-ke-proses" berdasarkan port, tetapi menyerahkan sebagian besar masalah keandalan dan kontrol aliran ke aplikasi itu sendiri atau protokol lain di atasnya

1.3.2 IP Address in Transport Layer (1 point)

Jika alamat IP dibutuhkan untuk menentukan tujuan pada protokol End-to-End, mengapa format paket pada protokol transport layer seperti TCP dan UDP tidak menyimpan informasi IP tujuan? Kaitkan dengan proses enkapsulasi-dekapsulasi dan relasi transport layer dengan lapisan-lapisan lainnya pada OSI model.

Answer:
Karena pada Layer 3 (Network) dan Layer 4 (Transport) memiliki responsibility masing2 dan menyediakan layanan ke layer atas maupun bawah.

Network Layer

- Bertanggung jawab untuk routing dan pengalamatan logical (IP addressing)
- Menentukan jalur terbaik untuk pengiriman data antar network
- Menyimpan informasi IP source dan destination dalam header IP

Transport Layer

- Fokus pada end-to-end communication antara aplikasi
- Menyediakan layanan seperti reliability, flow control, dan error detection
- Menggunakan port numbers untuk multiplexing/demultiplexing

Enkapsulasi dan Dekapsulasi

- Ketika sebuah aplikasi mengirim data, data tersebut melewati protokol-protokol dari lapisan atas ke lapisan bawah. Proses ini disebut enkapsulasi. Setiap lapisan menambahkan header (dan kadang trailer) sendiri ke unit data yang diterimanya dari lapisan di atas
- Pada lapisan transport (misalnya, TCP atau UDP), data aplikasi dipecah menjadi segmen atau datagram, dan header TCP/UDP (yang berisi nomor port) ditambahkan. Unit data ini kemudian diserahkan ke lapisan jaringan.
- Lapisan Jaringan (IP) kemudian mengenkapsulasi segmen/datagram TCP/UDP ini dengan menambahkan header IP-nya sendiri, yang berisi alamat IP sumber dan tujuan
- Saat paket tiba di host tujuan, proses dekapsulasi terjadi (dari lapisan bawah ke atas). Lapisan IP di host tujuan membaca header IP untuk menentukan bahwa paket tersebut ditujukan ke host ini dan kemudian melepas header IP. Paket yang tersisa (termasuk header TCP/UDP) kemudian diserahkan ke lapisan transport.
- Lapisan transport di host tujuan kemudian menggunakan header TCP/UDP (terutama nomor port tujuan) bersama dengan informasi alamat IP sumber dan tujuan yang telah diberikan oleh lapisan IP (sebagai bagian dari konteks paket) untuk mengarahkan data ke proses aplikasi yang benar. Misalnya, kunci demultiplexing TCP adalah 4-tuple: $\langle$SrcPort, SrcIPAddr, DstPort, DstIPAddr$\rangle$. Meskipun SrcIPAddr dan DstIPAddr tidak ada sebagai field langsung di header TCP, mereka adalah bagian penting dari identifikasi koneksi yang disediakan oleh lapisan bawah.

  Dengan demikian, protokol transport layer tidak perlu menambahkan informasi alamat IP di header mereka sendiri karena informasi ini sudah ditangani dan disediakan oleh lapisan IP di bawahnya.

1.3.3 TCP Checksum Calculation (0.5 points)

Pada perhitungan checksum TCP, data yang digunakan bukan hanya payload, melainkan ditambahkan dengan sebuah pseudo-header yang berisi header dari paket TCP tersebut, serta beberapa informasi tambahan. Menurutmu, apakah penambahan data-data tambahan ini penting atau tidak untuk kebutuhan error-detection pada protokol TCP? Jelaskan alasannya.

Answer:

Ya, sangat penting penambahan data pseudo-header pada perhitungan checksum TCP (dan juga UDP) sangat penting untuk kebutuhan error-detection pada protokol TCP.

Alasannya adalah:

- Verifikasi End-to-End yang Lebih Kuat - Checksum TCP dihitung tidak hanya atas header TCP dan payload (data) TCP, tetapi juga atas pseudo-header. Pseudo-header ini berisi beberapa field penting dari header IP, yaitu alamat IP sumber, alamat IP tujuan, dan nomor protokol, ditambah dengan field panjang pada UDP atau TCP

- Deteksi Error saat Pengiriman - Verifikasi isi pesan dan pada endpoint yang benar. Misal pada skenario di mana sebuah paket IP entah bagaimana mengalami korupsi pada field alamat IP tujuan saat dalam perjalanan melalui jaringan, sehingga paket tersebut salah alamat (misdelivered) ke host yang salah.

- Integritas Lintas Lapisan - Ini memberikan lapisan error-detection tambahan yang melampaui hanya integritas data dalam segmen TCP atau datagram UDP itu sendiri. Ini memastikan bahwa konteks pengiriman paket (yaitu, dari mana dan ke mana ia seharusnya pergi di tingkat jaringan) juga diverifikasi oleh lapisan transport.

1.4 QUIC Protocol Analogy (4 points)

Bayangkan seorang karyawan JNE yang sedang mengantar paket yang biasanya naik tangga satu persatu. Dia harus membuka gerbang, lalu masuk ke rumah, dan akhirnya baru menyerahkan paket. Tapi sekarang, dia bisa melakukan teleportasi dan langsung meletakkan paketnya di balkon tanpa melewati prosedur panjang tersebut. Cepat, tetapi juga membuat penjaga rumah bingung karena tidak sempat mengenali siapa dia, darimana asalnya, atau apakah dia memang berhak masuk.

1.4.1 QUIC Transport and Encryption Integration

Bagaimana "kurir" tersebut mencerminkan pendekatan QUIC dalam menyatukan transport dan enkripsi?

Answer:

Pada contoh yang awal merupakan mekanisme tradisional yaitu TCP+TLS dimana:

- Buka gerbang -¿ TCP handshake (SYN, SYN-ACK, ACK)

- Masuk rumah dan verifikasi identitas -¿ TLS handshake (Certificate, Key Exchange)

- Baru serahkan paket -¿ Data transfer

Sedangkan yang teleport adalah QUIC dimana:

- Langsung muncul di balkon dengan paket terenkripsi dan credential yang sudah terintegrasi
- Menggabungkan transport setup dan enkripsi dalam satu "lompatan"
- 0-RTT atau 1-RTT vs 2-3 RTT pada TCP+TLS

Ini terjadi karena QUIC mengintegrasikan TLS 1.3 langsung ke dalam transport layer, sehingga proses establishing connection dan enkripsi terjadi bersamaan, bukan bertahap.

1.4.2 Security Risks

Apa risiko keamanan dari kurir yang tidak melewati gerbang terlebih dahulu, tetapi langsung muncul dan meletakkan paketnya?

Answer:

Risiko keamanan yang dapat terjadi adalah:

- **Amplification Attacks**
  - Kurir palsu bisa mengirim paket besar ke alamat korban menggunakan IP spoofing
  - Tanpa validasi alamat pengirim di "gerbang", server bisa dimanfaatkan untuk DDoS reflection
- **Connection Hijacking**
  - Penjaga rumah bingung membedakan kurir asli vs penyusup karena tidak ada proses verifikasi bertahap
  - Connection ID bisa di-spoof jika tidak ada validasi yang cukup

1.4.3 Connection Migration

Jika kurir tersebut bisa pindah dari rumah ke rumah tanpa kembali ke kantor pusat terlebih dahulu, bahkan saat pindah kota, apa tantangan dan keuntungannya dalam QUIC?

Answer:

Keuntungan:

- Seamless handover - Kurir bisa pindah network (WiFi ke 4G, ganti ISP) tanpa memutus pengiriman
- No reconnection overhead - Tidak perlu ulang dari awal seperti TCP yang akan timeout
- Better user experience - Streaming video tidak terputus saat pindah jaringan

Tantangan:

- Path validation: Bagaimana memastikan jalur baru aman dan legitimate?
- NAT traversal: Kurir harus bisa menembus firewall/NAT di lokasi baru
- Load balancing: Server pool harus bisa "mengenali" kurir yang sama meski datang dari IP berbeda

1.5 Reverse Proxy Architecture (2.5 points)

Kamu adalah seorang engineer yang sedang mengembangkan aplikasi kesehatan bernama SATUSEHAT. Saat melakukan evaluasi arsitektur sistem, kamu menyadari bahwa komunikasi antara client dan server masih berlangsung secara langsung, tanpa menggunakan reverse proxy. Menurutmu, apakah perlu menambahkan reverse proxy dalam arsitektur sistem ini? Jelaskan pendapatmu dengan mempertimbangkan aspek keamanan, skalabilitas, performa, dan kemudahan pengelolaan. Sampaikan argumenmu seolah-olah kamu sedang menjelaskan kepada para stakeholder non-teknis yang perlu memahami manfaatnya dalam konteks bisnis dan operasional.

Answer:

Sebagai engineer SATUSEHAT, saya sangat merekomendasikan penambahan reverse proxy dalam arsitektur sistem kita.

Dari aspek keamanan, reverse proxy bertindak sebagai "benteng pertahanan" yang melindungi data sensitif pasien dengan menyembunyikan server internal, memfilter serangan, dan menerapkan enkripsi terpusat - hal ini krusial untuk compliance regulasi kesehatan.

Untuk skalabilitas, reverse proxy memungkinkan load balancing dan auto-scaling saat SATUSEHAT melayani lebih banyak fasilitas kesehatan, memastikan sistem tetap up dan responsif tanpa downtime yang merugikan layanan medis yang dapat berakibat pada kepuasan pasien.

Dari sisi performa, fitur caching dan compression akan mempercepat loading aplikasi, meningkatkan produktivitas tenaga medis sehingga pasien dapat lebih cepat dilayani yang meningkatkan kepuasan pasien.

Meskipun memerlukan investasi awal, manfaat jangka panjangnya jauh lebih besar: efisiensi operasional, mitigasi risiko keamanan, dan kesiapan untuk pertumbuhan masa depan. Reverse proxy bukan sekadar upgrade teknis, tetapi investasi strategis untuk mendukung digitalisasi kesehatan yang aman, stabil, dan dapat berkembang.

1.6 SOCKS5 Protocol (2.5 points)

Jelaskan mekanisme bagaimana SOCKS5 dapat establish TCP connections ke target servers melalui proxy layer. Mengapa SOCKS5 memerlukan additional protocol layer di atas TCP, dan apa fundamental advantages dari proxy approach dibandingkan direct TCP connections?

Answer:

SCOKS5 bekerja sebagai intermediary layer yang memungkinkan client untuk establish TCP connections ke target server melalui proxy server.

Mekanisme:

1. Authentication Phase - Client pertama-tama membuat TCP connection ke SOCKS5 proxy server dan melakukan handshake untuk menentukan authentication method yang akan digunakan (no authentication, username/password, atau GSSAPI).

2. Connection Request Phase - Setelah authentication berhasil, client mengirim connection request yang berisi:

    - Command type (CONNECT untuk TCP)
    - Address type (IPv4, IPv6, atau domain name)

- Target server address dan port
- Proxy server kemudian mencoba establish connection ke target server

3. Relay Phase - Jika connection ke target berhasil, proxy server memberikan response sukses ke client. Selanjutnya proxy server berfungsi sebagai transparent relay, meneruskan semua data antara client dan target server secara bidirectional.

SOCKS5 memerlukan protocol layer tambahan di atas TCP karena beberapa alasan:

- Control and Coordination - TCP hanya menyediakan reliable data transport, tetapi tidak memiliki mekanisme built-in untuk koordinasi proxy operations. SOCKS5 protocol menyediakan struktur untuk authentication, address resolution, dan connection establishment melalui proxy.

- Address Abstraction - SOCKS5 memungkinkan client untuk specify target menggunakan domain names, bukan hanya IP addresses. Proxy server yang melakukan DNS resolution, sehingga client tidak perlu tahu IP address sebenarnya dari target server.

- Security and Access Control - Protocol layer ini memungkinkan implementasi authentication mechanisms dan access control policies pada proxy server, yang tidak mungkin dilakukan dengan direct TCP connections.

Keunggulan:

- Enhanced Privacy and Anonymity - Karena target server hanya melihat connection dari proxy server, bukan dari client sebenarnya. Ini memberikan layer penyembunyian sehingga nampak anonim dan memungkinkan client untuk menyembunyikan IP address dan location asli mereka.

- Centralized Traffic Management - Semua outbound connections dapat di-route melalui single atau multiple proxy servers, memungkinkan centralized monitoring, logging, bandwidth management, dan content filtering karena melalui proxy.

- Load Distribution and Failover - Multiple proxy servers dapat digunakan untuk distribute load dan provide redundancy. Jika salah satu proxy server fails, client dapat failover ke proxy server lain.

1.7 Interior Gateway Protocol Design (4 points)

Misal kamu adalah seorang arsitek jaringan yang ditugaskan untuk merancang sistem routing internal (Interior Gateway Protocol/IGP) untuk dua klien yang sangat berbeda:

- **Klien A (Toko Ritel):** Sebuah jaringan kecil dengan 5 router yang menghubungkan kasir, gudang, dan kantor manajer.

- **Klien B (Pusat Data):** Sebuah jaringan perusahaan besar yang kompleks dengan 50 router, jalur redundan, koneksi fiber optik, dan membutuhkan kecepatan pemulihan yang kuat saat ada link yang putus.

Kamu memiliki dua pilihan jenis protokol routing utama: Distance Vector dan Link State. Dari dua jenis pilihan tersebut:

1.7.1 Client A Protocol Selection (1.5 points)

Untuk jaringan Klien A, filosofi protokol mana yang lebih cocok? Mengapa cara kerja fundamental protokol tersebut lebih sesuai dibanding protokol satunya lagi?

Answer:

Filosofi yang cocok untuk Klien A adalah "simplicity over sophistification" yaitu menggunakan distance vector.

Kenapa cara kerja distance vector cocok:

- Prinsip "Bellman-Ford" - Setiap router hanya perlu tahu "berapa hop" dan "ke arah mana" untuk mencapai tujuan, tanpa perlu memahami seluruh topologi jaringan karena jaringan kecil yang simple
- Periodic updates - Router berbagi routing table secara berkala dengan tetangga langsung - pendekatan yang stabil dan predictable untuk lingkungan sederhana
- "Rumor-based learning" - Informasi routing disebarkan seperti gossip dari tetangga ke tetangga, cocok untuk jaringan flat tanpa hierarki kompleks

Alasan lebih cocok dibanding link state:

- Resource efficiency - Tidak perlu membangun dan maintain Link State Database yang kompleks untuk hanya 5 router.
- Administrasi Simple - Staff IT toko ritel dapat mengelola protokol yang hanya fokus pada "jarak dan arah" tanpa perlu memahami algoritma SPF.
- Bandwidth conservation - Update periodik yang kecil lebih ekonomis dibanding flooding LSA untuk topologi sederhana
- Failure tolerance yang cukup - Karena non-critical kecepatan yang relatif lambat masih dapat ditolerir.

1.7.2 Client B Protocol Selection (1.5 points)

Untuk jaringan Klien B, mana yang menjadi pilihan yang jelas? Bagaimana mekanisme protokol ini secara langsung menjawab kebutuhan?

Answer:

Link State mutlak diperlukan untuk kompleksitas dan kritikalitas pusat data:

Mekanisme yang Menjawab Kebutuhan Langsung:

- Rapid convergence: Ketika fiber link putus, setiap router sudah memiliki complete topology map dalam LSDB, sehingga dapat langsung menjalankan Dijkstra algorithm untuk menghitung alternate path dalam hitungan detik
- Flooding mechanism: LSA (Link State Advertisement) disebarkan secara instant ke seluruh area ketika ada perubahan, memastikan semua router memiliki informasi terkini secara simultan
- Event-driven updates: Hanya mengirim update saat ada perubahan topology (bukan periodic), mengoptimalkan bandwidth precious di backbone fiber

- Hierarchical scaling: Mendukung area-based design yang memungkinkan segmentasi 50 router menjadi multiple areas untuk manageability dan performance

Selain itu dikarenakan memiliki path yang redundant, setiap router dapat mrnghitung shortest path tree secara independen menggunakan algoritma yang sama, sehingga menghasilkan jalur optimal berdasarkan biaya/jarak/cost yang real time.

1.7.3 Count to Infinity Problem (1 point)

Salah satu kelemahan fatal dari protokol Distance Vector adalah masalah "Count to Infinity". Mengapa protokol Link State, berdasarkan desain dasarnya, secara inheren kebal terhadap masalah spesifik ini?

Answer:

Count to Infinity terjadi pada Distance Vector karena information propagation delay dan dependency loop:

- Router A kehilangan route, tetapi masih menerima advertisement dari Router B yang belum tahu tentang failure
- Terjadi "bounce effect" dimana metric terus bertambah hingga mencapai infinity

Link State inherent kebal karena setiap router "aware" terhadap semua tetangga pada network. Karena pada LS setiap router memiliki complete copy of topology ytang disimpan dalam database pada router tersebut. Jika ada link putuh, LSA dengan seq number baru langsung diflood ke semua router, memberikan "single source of truth" yang konsisten (Router tidak mengandalkan informasi yang didapat dari tetangga). Selain itu karena tiap router memiliki complete copy, loop dapat dideteksi secara otomatis sebelum dibuat forwarding table.

1.8 Deep Packet Inspection and VPN (3 points)

Anda sudah belajar salah satu cara mem-bypass content filter Kominfo, yaitu via DNS over HTTPS (DoH). Namun, sekitar 2023, Kominfo mengeluarkan sebuah jurus pamungkas, sesuatu yang lebih dahsyat daripada sekadar blocking IP atau DNS poisoning, yaitu Deep Packet Inspection + TCP Reset Attack.

Untungnya, para pengguna internet di dunia sudah jauh lebih modern daripada manusia-manusia di Kominfo, jadi jurus pamungkas itu sudah tidak level bagi kita yang sudah punya jurus untuk meng-counter-nya.

Kita bisa menggunakan VPN untuk mengatasi serangan kominfo. Namun, TCP Reset Attack itu bekerja dengan mengirim TCP reset packet palsu ke client dan server. Karena VPN bekerja sebagai proxy, client di sini tetap sama, dan server berbeda (server sekarang adalah server VPN, bukan server situs tujuan). Lantas, apa yang menyebabkan serangan kominfo berhasil dilakukan ketika tujuannya adalah server situs terlarang, tetapi gagal untuk server VPN?

Answer:

VPN server tidak masuk dalam blacklist Kominfo karena:

- VPN providers menggunakan IP addresses dan domain names yang tidak terdaftar sebagai "situs terlarang"

- SNI dan Host headers menunjuk ke server VPN yang legitimate, bukan ke situs target
- Traffic ke VPN server terlihat seperti koneksi bisnis/personal yang normal

Setelah koneksi VPN terbentuk:

- Semua traffic di-enkripsi end-to-end dalam VPN tunnel
- DPI tidak bisa lagi membaca destination headers, SNI, atau content
- Yang terlihat oleh ISP hanya encrypted traffic ke VPN server
- Traffic sebenarnya ke situs "terlarang" terjadi dari VPN server, bukan dari connection point di Indonesia

Jadi in nutshell, TCP Reset Attack hanya triggered ketika DPI mendeteksi koneksi ke target yang masuk blacklist. VPN server sendiri bukan target, sehingga koneksi ke VPN berjalan normal, dan setelah tunnel terbentuk, semua traffic real sudah ter-enkripsi dan tidak bisa di-inspect lagi.

1.9 Network Troubleshooting (2 points)

Suatu hari kamu mencoba melakukan ping ke hiyoritomoe.com dan mendapatkan balasan yang normal, dengan waktu respons yang normal. Namun, ketika kamu mencoba curl hiyoritomoe.com, perintah tersebut gagal.

1.9.1 Protocol and Network Layer Analysis (0.5 points)

Jelaskan protokol dan lapisan jaringan yang digunakan oleh ping dan curl!

Answer:

Ping:

- Menggunakan protokol ICMP (Internet Control Message Protocol)
- Beroperasi pada Network Layer (Layer 3) dari model OSI
- Tidak menggunakan port TCP/UDP
- Mengirim ICMP Echo Request dan menerima ICMP Echo Reply

Curl:

- Menggunakan protokol HTTP/HTTPS (default port 80/443)
- Beroperasi pada Application Layer (Layer 7) dari model OSI
- Menggunakan TCP sebagai transport protocol (Layer 4)
- Memerlukan koneksi TCP three-way handshake sebelum transfer data

1.9.2 Possible Causes (0.5 points)

Mengapa skenario seperti di atas bisa terjadi? Sebutkan dua kemungkinan penyebab, dan bagaimana cara kamu bisa menyelidikinya.

Answer:

Dua kemungkinan penyebab:

1. Firewall memblokir traffic HTTP/HTTPS tetapi mengizinkan ICMP
    - Cara menyelidiki: Gunakan telnet hiyoritomoe.com 80 atau nmap -p 80,443 hiyoritomoe.com untuk mengecek apakah port HTTP/HTTPS terbuka
2. Web server di hiyoritomoe.com sedang down/tidak berjalan, tetapi host masih merespons ICMP
    - Cara menyelidiki: Gunakan nmap -sS hiyoritomoe.com untuk port scanning atau curl -v hiyoritomoe.com untuk melihat detail error message

1.9.3 Network Configuration (0.5 points)

Konfigurasi jaringan atau firewall seperti apa yang bisa menyebabkan gejala ini terjadi?

Answer:

Konfigurasi yang dapat menyebabkan gejala ini:

- Stateful firewall rules yang memblokir outbound HTTP/HTTPS connections tetapi mengizinkan ICMP
- Deep Packet Inspection (DPI) yang memfilter traffic berdasarkan content/protocol
- Proxy configuration yang mengharuskan HTTP traffic melalui proxy server

1.9.4 VirtualBox Network Issue (0.5 points)

Pada hari lain, saat kamu mengatur host-only network di VirtualBox pada mesin host Windows, kamu menyadari bahwa VM tidak bisa melakukan ping ke host, tetapi curl ke server web lokal di host berhasil. Ceritakan kemungkinan penyebab hal ini terjadi!

Answer:

Pada host-only network di Windows, kemungkinan besar disebabkan oleh Windows Firewall yang memblokir ICMP traffic tetapi mengizinkan HTTP traffic pada interface VirtualBox Host-Only Network.

2 Sistem Paralel dan Terdistribusi

Distributed and Parallel Sisters

2.1 Distributed File System Performance (2.5 points)

Dibanding dengan mesinmu sendiri, terkadang membuka file yang disimpan pada distributed file system memakan waktu yang jauh lebih lama.

2.1.1 Bottleneck Areas

Sebutkan dua area di mana bottleneck ini mungkin terjadi!

Answer:

Kemungkinan area bottleneck:

- Network latency dan bandwidth: Transfer data melalui jaringan jauh lebih lambat dibanding akses lokal disk/memory
- Metadata lookup overhead: Setiap operasi file memerlukan komunikasi dengan metadata server untuk lokasi dan permission

2.1.2 Solutions

Bagaimana masalah ini dapat diatasi? (hint: bisa me-refer ke filesystem yang sudah ada, seperti GFS atau HDFS)

Answer:

Solusi yang mungkin:

- Data locality - Penjadwalan komputasi di node yang sama dengan data (HDFS block placement)
- Caching dan replication - Banyak replica data mengurangi network hops, metadata caching di client
- Batch operations - Menggabungkan berbagai operasi kecil menjadi operasi yang lebih besar namun lebih sedikit

2.2 Vector Clock and Causal Ordering (2.5 points)

Jelaskan bagaimana vector clock dapat mendeteksi concurrent events dan causal relationships. Mengapa vector clock lebih powerful daripada logical clock untuk causal ordering?

Answer:

Vector clock menggunakan array counter [V1, V2, ..., Vn] untuk setiap process. Event A concurrent dengan B jika tidak ada process yang vector clock-nya lebih kecil di semua elemen.

Keunggulan:

- Causal detection - Logical clock hanya bisa ordering total, vector clock bisa deteksi hubungan kausal yang sesungguhnya
- Concurrent identification - Vector clock bisa membedakan events yang benar-benar concurrent vs yang hanya tampak concurrent
- Stronger consistency - Memberikan partial ordering yang lebih akurat sesuai happened-before relationship

2.3 Global Clock Absence (2.5 points)

Sistem terdistribusi tidak memiliki satu sumber waktu global yang disepakati oleh semua node. Menurut pendapatmu, apa konsekuensi dari tidak adanya global clock ini terhadap sinkronisasi proses, event ordering, dan koordinasi antar node?

Answer:

Kata yang cocok mewakili → *kacau*

Alasan:

- Ambiguitas - Karena tidak ada patokan global, akan sulit menentukan urutan kejadian yang terjadi pada node yang berbeda.
- Konsistensi - susah menentukan urutan koordinasi state changes tanpa menggunakan referensi waktu.
- Sinkronisasi Proses - Diperlukan algoritma khusus seperti vector clocks atau logical stamps.

2.4 False Sharing (3 points)

Apa itu False Sharing? Mengapa ia disebut 'silent killer' dalam parallel programming?

Follow up question: "Jika dua thread pada CPU berbeda mengakses variabel A dan B yang terletak dalam cache line yang sama, mengapa performa turun drastis meskipun tidak ada race condition? Jelaskan solusinya!"

Answer:

Definisi False Sharing: Ketika dua variabel independen berada dalam cache line yang sama, modifikasi pada salah satu variabel menyebabkan cache line di core lain menjadi invalid.

Kenapa disebut Silent Killer, karena:

- Tidak ada race condition atau error yang terdeteksi
- Performance degradation drastis tanpa warning
- Sulit di-debug karena code terlihat benar

Solusi follow up question:

- Padding: Memisahkan variabel dengan dummy data agar berbeda cache line
- Thread-local storage: Setiap thread punya copy lokal

2.5 Microservices Communication (3 points)

Video. Setelah menonton video tersebut, kita asumsikan komunikasi antar layanan utama seperti `Image`, `Metadata`, `Feed`, `Like`, `Comment`, dan `Fanout` dilakukan melalui Remote Procedure Call (RPC), sementara proses `Fanout` ke followers dilakukan secara asinkron melalui Message Queue (kalau ini sesuai dengan video).

Bagaimana layanan-layanan seperti `Image`, `Metadata`, dan `Feed` saling berinteraksi menggunakan RPC? Dan mengapa `Fanout` ke followers sebaiknya diproses secara asinkron menggunakan Message Queue, bukan RPC langsung?

Answer:

Interaksi Layanan melalui RPC:

1. Image Service ↔ Metadata Service

```
User upload image -> Image Service memproses file
                  ->RPC call ke Metadata Service untuk simpan:
                    - Caption, location, timestamp
                    - User ID, image URLs
                    - Hashtags dan mentions
```

2. Feed Service ↔ Multiple Services

```
User request news feed -> Feed Service:
                       -> RPC ke Metadata Service (post
                          details)
                       -> RPC ke Like Service (like counts)
                       -> RPC ke Comment Service (comment
                          counts)
                       -> Aggregate semua data untuk response
```

3. API Gateway koordinasi:

   - API Gateway menerima request dan routing ke service yang tepat via RPC
   - Authentication, rate limiting, dan validation dilakukan sinkron
   - Response langsung dikirim ke client

jika fanout menggunakan RPC langsung akan dapat muncul masalah seperti:

- Blocking dan Latency Ekstrem

```
Contoh real case:
- Influencer dengan 50 juta followers posting
- RPC langsung = 50 juta sequential/parallel calls
- Waktu response: 30-60 detik (atau timeout)
- User menunggu lama untuk konfirmasi upload
```

- Resource Exhaustion

```
- 50 juta RPC connections sekaligus
- Memory consumption: Memory yang besar sekali hanya untuk
   koneksi
- Network bandwidth: saturated
- Database overload: 50 juta UPDATE queries sekaligus
- Besar kemungkinan server crash
```

- Cascade Failure

```
Jika 1\% followers (500,000) mengalami network issue:
- 500,000 RPC calls timeout
- Seluruh fanout process gagal
- User tidak dapat posting
- System unavailable
```

Message Queue menyelesaikan masalah tersebut dengan:

- Decoupling dan Non-blocking

```
Dengan Message Queue:
User post -> Metadata saved (200ms)
          -> Message queued (10ms)
          -> Return "Success" ke user
          -> Background fanout processing

User experience: 210ms vs 30-60 seconds
```

- Controlled Processing Rate

```
Queue configuration:
- 1000 workers processing parallel
- Each worker: 100 followers/batch
- Processing rate: 100,000 followers/second
- Controlled resource usage
```

- Fault Isolation

```
Jika beberapa followers gagal:
- Message retry mechanism
- Dead letter queue untuk permanent failures
- 99.9% followers tetap mendapat update
- System tetap available
```

- Eventual Consistency yang masih tolerable

```
Social media context:
- User posting: immediate feedback needed (RPC)
- Followers seeing post: delay 1-5 seconds acceptable
- Trade-off: user experience vs system stability
```

2.6 Distributed vs Decentralized Systems (3 points)

Artikel. Setelah membaca artikel tersebut, jelaskan konklusi kamu (seolah-olah kamu sedang menceritakannya ke teman kamu yang kuliah di jurusan bisnis dan merupakan investor crypto akut) terkait perbedaan sistem terdistribusi dan terdesentralisasi.

Answer:

In nutshell distributed masih ada central authority tapi decentralized ngga.

Distributed System itu kayak McDonald, cabangnya banyak, ada dimana, tapi memiliki kantor pusat/pusat kontrol (head office), tapi operasionalnya tersebar.

Kalau Decentralized itu kaya mainan lu, Bitcoin. Ngga ada yang jadi pemiliki otoritas. Setiap node equal dan keputusan dibaut secara collective.

3 Teknologi Sistem Terintegrasi

Integrating the freeze team wheelchair

Penilaian pada soal berikut akan mengacu pada standar mata kuliah STI. Jawaban yang bertele-tele, terlalu panjang, dan menunjukkan kurangnya pemahaman konsep oleh penulis dapat mempengaruhi nilai secara negatif.

3.1 Furina Courthouse Corporation System Architecture (20 points)

Perusahaan bernama Furina Courthouse Corporated .inc adalah sebuah perusahaan yang bergerak di sektor layanan hukum digital dengan fokus utama pada efisiensi proses peradilan dan legalitas melalui transformasi digital. Visi perusahaan ini adalah menjadi pioneer dalam modernisasi pelayanan hukum di seluruh Fontaine dengan mengakomodasi kebutuhan masyarakat yang semakin terdigitalisasi.

Model bisnis FCC berupa B2B (Business to Business) dan B2G (Business to Government) dengan menyediakan platform dan layanan kepada:

- Opera Epiclese (Mahkamah Agung Fontaine) dan Gardes (Kepolisian Fontaine) untuk digitalisasi proses hukum, administrasi, dan manajemen arsip hukum.
- Perusahaan besar, terutama yang memiliki kebutuhan legalisasi dan compliance yang tinggi.

Layanan utama yang diberikan oleh FCC antara lain:

- Layanan manajemen kasus untuk pelacakan dan dokumentasi proses hukum dari awal hingga keputusan.
- Digital legal archive untuk menyimpan dan manajemen arsip hukum.
- E-Litigation platform untuk memfasilitasi proses persidangan jarak jauh dengan bukti elektronik.
- Layanan otentikasi & validasi dokumen untuk legalisasi digital dokumen hukum dan tanda tangan elektronik.

Strategi bisnis yang dimiliki oleh FCC berupa kemitraan strategis dengan institusi negara untuk memperkuat legitimasi serta layanan konsultasi hukum secara digital untuk lembaga atau perusahaan yang baru mulai melakukan transformasi digital dalam operasional hukumnya.

3.1.1 System Architecture Design

Bagaimana arsitektur sistem yang harus dibuat agar semua lembaga hukum (Opera Epiclese dan Gardes) dapat saling berjalan secara efisien, mempunyai skalabilitas yang tinggi, dan memiliki interoperabilitas yang tinggi. Buatlah dan jelaskan diagram dari arsitektur sistem tersebut!

Answer:

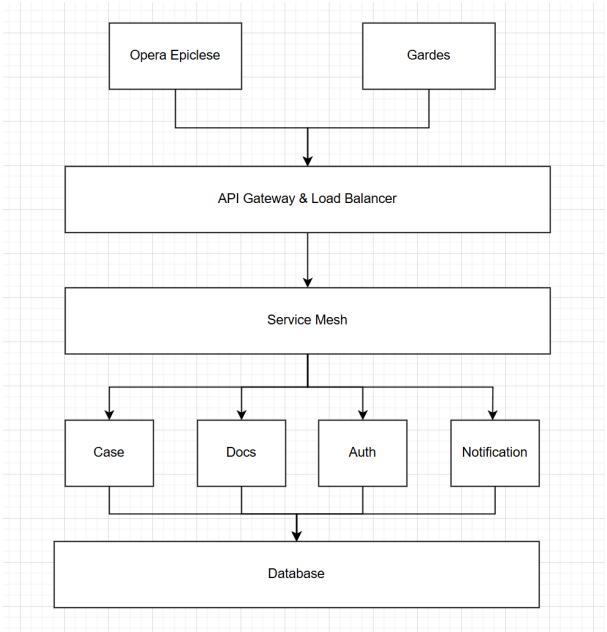

Figure 6: Arsitektur FCC

Komponen Utama:

- API Gateway - Mengelola routing, autentikasi, rate limiting
- Service Mesh - Komunikasi antar-service yang aman dan observable
- Microservices - Case Management, Document Management, Authentication, Notification
- Database Layer - PostgreSQL cluster untuk konsistensi, Redis untuk caching

3.1.2 Smart City IoT Integration

FCC berencana bekerja sama dengan smart city untuk mengintegrasikan data dari IoT (seperti kamera lalu lintas aquabus, bodycam gardes, dan sensor kejadian). Bagaimana arsitektur sistem terintegrasi harus dibangun agar data tersebut dapat digunakan secara sah dan real-time di pengadilan?

Answer:

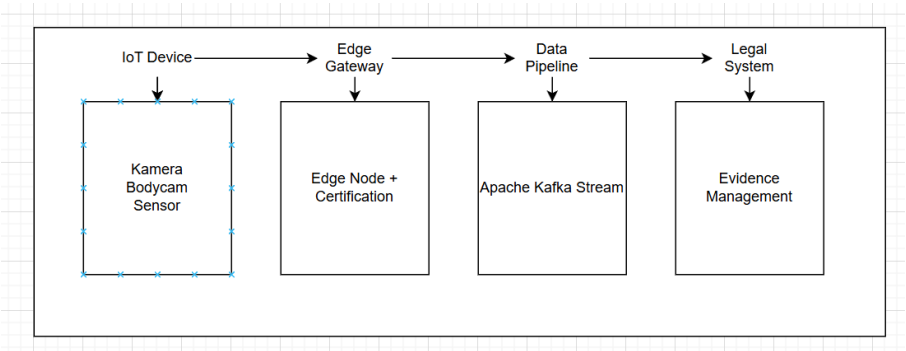

Figure 7: IoT Integration Scheme

Komponen penting:

- Real-time Processing - Apache Kafka + Apache Flink untuk streaming data
- Digital Signature - Setiap data IoT ditandatangani dengan certificate
- Blockchain Ledger - Immutable record untuk chain of custody
- API Gateway- Unified access dengan proper authentication

3.1.3 Security and Privacy Challenges

Apa tantangan keamanan dan privasi ketika sistem hukum digital seperti yang dikembangkan pada FCC harus mengintegrasikan perangkat IoT tersebut?

Answer:

- Data Integrity - Memastikan data IoT tidak dimanipulasi
- Privacy Protection - Menganonimasi data pribadi sesuai regulasi
- Secure Communication - End-to-end encryption dari device ke sistem
- Access Control - Role-based access untuk berbagai stakeholder
- Audit Trail - Logging semua akses dan modifikasi data

3.1.4 Electronic Evidence Integrity

Jelaskan bagaimana FCC dapat menjamin keabsahan bukti elektronik yang diambil dari perangkat IoT agar tidak dipengaruhi oleh faktor internal (misalnya oknum penegak hukum atau lembaga sendiri)?

Answer:

Mekanisme Keabsahan:

- Digital Timestamping - timestamping untuk validtas waktu
- Multi-level Signatures - signature untuk menghalangi pemalsuan
- Blockchain Immutability - Hash data disimpan di blockchain publik
- Witness Validation - Multiple independent validators
- Changed Detection - Cryptographic checksums dan integrity monitoring

3.1.5 Scalability and Cost Efficiency

Dengan meningkatnya volume data dari sistem terintegrasi dan IoT, bagaimana FCC sebaiknya mengatur strategi skalabilitas dan efisiensi biaya penyimpanan tanpa mengorbankan integritas dan ketersediaan data untuk keperluan hukum.

Answer:

Strategi Storage:

- Hot Storage - SSD untuk data aktif (3 bulan)

- Warm Storage - HDD untuk data semi-aktif (2 tahun)
- Cold Storage - Cloud storage untuk arsip jangka panjang
- Data Compression - Algoritma lossless untuk dokumen legal
- Deduplication - Menghilangkan data duplikat

Skalabilitas:

- Horizontal scaling dengan containerization (k8s)
- Database sharding berdasarkan kasus/wilayah
- Content Delivery Network (CDN) untuk akses global

3.1.6 Risk Assessment and Mitigation

Evaluasi risiko dan manfaat dari mengintegrasikan sistem peradilan digital dengan perangkat IoT di lapangan. Apa langkah mitigasi yang harus disiapkan Furina sebagai penyedia platform?

Answer:

Manfaat:

- Efisiensi proses peradilan (30-50% lebih cepat)
- Transparansi dan akuntabilitas tinggi
- Mengurangi biaya operasional jangka panjang
- Akses bukti real-time untuk investigasi

Risiko:

- Kerentanan cyber security
- Ketergantungan pada teknologi
- Potensi bias algoritma AI
- Kompleksitas integrasi sistem legacy

Mitigasi:

- Regular security audit dan penetration testing
- Backup sistem dan disaster recovery plan
- Training intensif untuk pengguna
- implementasi bertahap dengan pilot project

3.1.7 System Improvement Areas

Berdasarkan arsitektur sistem dan evaluasi risiko yang telah anda susun, identifikasi area-area yang masih bisa dikembangkan atau diperbaiki agar sistem terintegrasi FCC menjadi lebih aman dan mudah diterima oleh stakeholder. Buatlah penjelasan (non-teknis) yang dapat disampaikan kepada stakeholder non-teknis seperti hakim, jaksa, atau pejabat instansi pemerintah agar mereka dapat paham!

Answer:

Area Pengembangan:

- AI-Powered Analytics - Deteksi pola kriminal otomatis
- Mobile Integration - Aplikasi mobile untuk akses lapangan
- Multi-language Support - Mendukung bahasa daerah
- Predictive Maintenance - Monitoring kesehatan sistem IoT

Bayangkan jika sistem ini dapat meramalkan pola kriminal dan dengan membuat avaible di mobile, mengingat mayoritas masyarakat Indonesia memiliki smartphone memperluas jangkauan sistem, selain itu dengan menambahkan bahasa daerah, membantu akses bagi daerah dengan literasi yang mana didominasi oleh native speaker.

3.2 Legacy System Integration (2 points)

Jika kamu bekerja untuk suatu perusahaan kamu pasti akan menemukan sebuah legacy code. Legacy code adalah kode program yang sudah tua atau usang, tetapi masih digunakan dalam suatu sistem. Kenapa integrasi terhadap legacy sistem sering gagal walaupun secara teknis API telah diimplementasikan dan tinggal digunakan?

Answer:

Penyebab Kegagalan:

- Cultural Resistance - Tim legacy tua menolak perubahan dan mempertahankan cara kerja lama
- Data Inconsistency - Format dan struktur data berbeda, memerlukan transformasi ulang dari awal.
- Business Process - Legacy system dibuat untuk proses bisnis yang lama.
- Hidden Dependencies - Ketergantungan tidak terdokumentasi yang baru terungkap saat integrasi
- Performance Issues - Legacy system tidak dirancang untuk kompatibel dengan sistem modern
- Security Gaps - karena Standard keamanan legacy bisa saja sudah obselete bisa terjadi tidak kompatibel dengan sistem baru

4 Keamanan Informasi

She doesn't have anything to do with information security, I just really love her - Duke

Untuk setiap soal, berikan jawaban sesingkat dan sepadat mungkin. Jawaban yang terlalu bertele-tele berpotensi di-0-kan.

4.1 Information Security Concepts (4 points)

Meskipun sebenarnya berbeda, konsep-konsep keamanan informasi berikut seringkali dianggap sama dan atau saling tertukar. Menggunakan referensi, berikan definisi masing-masing konsep, beserta dengan bagaimana setiap konsep (dalam pasangan yang sama) dapat saling melengkapi atau berlawanan!

4.1.1 Security vs Privacy

Answer:

- Security - Perlindungan sistem, data, dan aset dari ancaman
- Privacy - Hak individu untuk mengontrol informasi pribadi mereka
- Hubungan - Security adalah enabler untuk privacy. Privacy membutuhkan security controls, tapi security tidak otomatis menjamin privacy.

4.1.2 Vulnerability vs Threat

Answer:

- Vulnerability - Kelemahan dalam sistem yang bisa dieksploitasi
- Threat - Potensi bahaya yang bisa mengeksploitasi vulnerability
- Hubungan - Risk = Threat $\times$ Vulnerability. Keduanya harus ada untuk menciptakan risiko.

4.1.3 Incident vs Attack

Answer:

- Incident - Peristiwa yang mengganggu operasi normal atau melanggar kebijakan security
- Attack - Tindakan sengaja untuk mengeksploitasi vulnerability
- Hubungan - Attack adalah subset dari incident. Tidak semua incident adalah attack (bisa human error).

4.1.4 Authentication vs Authorization

Answer:

- Authentication - Verifikasi identitas pengguna ("who you are")
- Authorization - Penentuan hak akses setelah terautentikasi ("what you can do")
- Hubungan - Sequential relationship - authentication harus terjadi sebelum authorization.

4.2 Security Operations Center Technology Stack (2 points)

Sebuah security operations center menggunakan Wazuh, Suricata, Kafka, Elasticsearch, dan Kibana. Jelaskan peran masing-masing teknologi serta aliran data yang ada!

Answer:

Peran Masing-masing:

- Wazuh - Host-based intrusion detection, log analysis, compliance monitoring
- Suricata - Network intrusion detection/prevention, signature-based detection
- Kafka - Message broker untuk streaming data real-time dengan high throughput
- Elasticsearch - Storage dan indexing untuk log analysis dan search
- Kibana - Visualization dan dashboard untuk security monitoring

4.3 SSO Security Policy (4 points)

Yayasan Rumah Perapian (YRP) sedang mempersiapkan sebuah sistem SSO untuk seluruh aplikasi internalnya. Buat sebuah kebijakan keamanan informasi yang berisi kebijakan, standar, prosedur, dan guideline terkait (kewajiban) login menggunakan SSO!

Answer:

SSO (Single Sign On) Mengatur bagaimana autentikasi user dan akses ke berbagai platform

STANDARD RULES:

- Multi-Factor Authentication (MFA) wajib untuk semua user
- Session timeout maksimal 8 jam untuk user biasa, 2 jam untuk admin
- Password policy: minimal 12 karakter, kompleks, diganti setiap 90 hari

PROCEDURES:

- User request akses melalui help desk
- Manager approval untuk pemberian akses
- IT Admin provision account di Identity Provider

  User training wajib sebelum akses diberikan
- Regular access review setiap 6 bulan

GUIDELINES:

- Gunakan corporate email sebagai username
- Jangan share credential SSO
- Logout setelah selesai bekerja
- Report aktivitas mencurigakan segera ke IT Security
- Akses dari device pribadi harus melalui VPN

4.4 Web Application Threat Modeling (10 points)

PT Inazuma Bersinar Selamanya (IBS) menyediakan berbagai layanan dalam bidang kelistrikan dan teknologi informasi. Untuk meningkatkan efisiensi, IBS membuat sebuah aplikasi web yang menyediakan portal interaksi terpusat bagi manajemen IBS, mitra kerja IBS, dan klien IBS.

4.4.1 Threat Model Creation

Buat threat model keamanan informasi untuk aplikasi web tersebut.

Asset and Actor Identification Identifikasi seluruh aset dan aktor yang mungkin terlibat dengan aplikasi!

Answer:

Aset:

- Customer data dan informasi billing
- Partner contracts dan commercial data
- Internal management information
- Application source code
- Database servers dan infrastructure

Aktor:

- Management IBS (internal user)
- Mitra kerja IBS (external partner)
- Klien IBS (customer)
- System administrators
- External attackers item Malicious insiders

STRIDE Threat Analysis Menggunakan metodologi STRIDE, identifikasi kemungkinan ancaman-ancaman terhadap aplikasi!

Answer:

1. Spoofing:
    - Impersonation mitra untuk akses data klien
    - Fake login pages untuk credential harvesting
2. Tampering:
    - Modifikasi data transaksi atau kontrak
    - Code injection untuk mengubah business logic
3. Repudiation:

- Mitra menyangkal transaksi yang telah dilakukan
- Log manipulation untuk menghilangkan jejak

4. Information Disclosure:
    - Data breach customer information
    - Unauthorized access ke commercial data
5. Denial of Service:
    - DDoS attack untuk mengganggu layanan
    - Resource exhaustion melalui API abuse
6. Elevation of Privilege:
    - Horizontal privilege escalation antar user
    - Vertical escalation dari user ke admin

4.4.2 Security Incident Analysis

Aplikasi dikembangkan tanpa melibatkan keamanan informasi secara menyeluruh. Setelah aplikasi diluncurkan, sekelompok hacker dari Snezhnaya berhasil mendapatkan akses tertinggi ke aplikasi. Penyelidikan menemukan bahwa para hacker mendapatkan akses awal dengan melakukan gabungan antara serangan social engineering dan brute force untuk mendapatkan akses ke akun mitra, sebelum kemudian memanfaatkan endpoint API tertentu dengan otentikasi lemah untuk menaikkan tingkat akses.

OWASP Top 10 Analysis Sebutkan dan jelaskan tiga kategori OWASP Top 10:2021 yang relevan terhadap kasus di atas!

Answer:

- A07:2021 – Identification and Authentication Failures
- A01:2021 – Broken Access Control
- A04:2021 – Insecure Design

Secure SDLC Prevention Jelaskan bagaimana Secure SDLC dapat mencegah terjadinya kasus tersebut, serta kegiatan-kegiatan yang relevan pada tiap tahap!

Answer:

- Planning - Threat modeling dan security requirements
- Design - Security architecture review, secure design patterns
- Implementation - Secure coding standards, static analysis
- Testing - Penetration testing, dynamic analysis
- Deployment - Security configuration review
- Maintenance - Regular security updates, monitoring

Similar Case Study Sebutkan dan jelaskan setidaknya satu contoh kasus atau insiden keamanan yang serupa. Berikan kronologi kasus tersebut beserta dengan pelajaran-pelajaran yang dapat digunakan pada kasus IBS; jangan lupa untuk juga mencantumkan referensi-referensi yang digunakan.

Answer:

Kasus: Equifax Data Breach 2017 Kronologi:

- Exploitasi vulnerability Apache Struts (CVE-2017-5638)
- Initial access melalui web application
- Privilege escalation untuk akses database
- Data 147 juta konsumen dicuri

Pelajaran untuk IBS:

- Patch management yang konsisten
- Web application firewall implementation
- Database access monitoring
- Incident response plan yang matang

Referensi:

- US House Committee Report on Equifax (2018)
- NIST Cybersecurity Framework v1.1

5 Miscellaneous

If you don't do these, he'll haunt your dreams tonight

5.1 Academic Integrity as Teaching Assistant (10.0 points - WAJIB)

Sebagai asisten, nantinya kamu akan dituntut untuk memastikan seluruh dan segala bentuk kecurangan akademis dalam mata kuliah-mata kuliah yang diasistensi ditemukan, dilaporkan, dan ditindaklanjuti.

5.1.1 Legal and Ethical Foundation

Sebutkan landasan (bebas; hukum, agama, etika, dan seterusnya) yang mewajibkan asisten untuk melakukan hal tersebut.

Answer:

- Buku Peraturan Akademik ITB, Pasal 35 Ayat 2 Tentang Pengawas Ujian
- Buku Peraturan Akademik ITB, Pasal 34 Ayat 1 Tentang Peserta Ujian
- Buku Peraturan Akademik ITB Pasal 38 Ayat 3 Tentang Penilaian Prestasi Akademik Mahasiswa

5.1.2 Detection Tools

Sebutkan tiga alat yang mampu membantumu menemukan dan atau mengonfirmasi tindak laku kecurangan.

Answer:

1. Hape/CCTV untuk Recorder saat Ujian.
2. Turnitin untuk plagiasi tulisan.
3. Github untuk cek plagiasi kode.

5.1.3 Academic Misconduct Guidelines

Jelaskan garis yang akan kamu tarik terkait kecurangan dan plagiarisme.

Answer:

5.1.4 Case Study Analysis

Berikut merupakan beberapa studi kasus. Untuk masing-masing kasus, jelaskan dengan singkat apa saja yang akan kamu lakukan untuk menentukan verdict akhir.

Case 1: Internet Code Similarity Terdapat kelompok tugas besar yang kodenya sangat mirip dengan kode yang tersedia di internet, namun mereka tidak memberikan pengakuan (credits) atasnya.

Answer:

Mengotak PIC Kelompok tersebut, kemudian menanyakan apakah kodenya terkait, jika dalam aturan tugas tidak diperbolehkan minta refactor atau akan diuji pemahaman akan kode sendiri untuk memastikan bahwa ditulis sendiri, jika diperbolehkan minta untuk cantumkan referensi.

Case 2: Senior Student Code Similarity Terdapat kelompok tugas besar yang kodenya sangat mirip dengan kode tugas besar karya kelompok kakak tingkat.

Answer:

Mengotak PIC Kelompok tersebut, kemudian menanyakan apakah kodenya terkait, jika dalam aturan tugas tidak diperbolehkan minta refactor atau akan diuji pemahaman akan kode sendiri untuk memastikan bahwa ditulis sendiri, jika diperbolehkan minta untuk cantumkan referensi.

Case 3: Exam Communication Terdapat beberapa mahasiswa yang saling berbincang dan tengok menengok ketika ujian.

Answer:

Menegur dan memberikan peringatan, jika tidak mengimbaukan peringantan sesuai dengan peraturan peserta ujian bisa diminta untuk keluar.

Case 4: Similar Non-Standard Answers Terdapat beberapa mahasiswa yang jawabannya sangat mirip satu sama lain, dan jawaban tersebut bukanlah jawaban standar.

Answer:

Melakukan background check kepada peserta-peserta tersebut, jika memang sering berdekatan akan dipertanyakan lebih lanjut,

Case 5: Tutor-Student Answer Similarity Terdapat mahasiswa yang jawabannya sangat mirip dengan mahasiswa lain yang kamu ketahui sering memberikan tutor, dan bukan merupakan seseorang yang sepertinya akan melakukan kecurangan.

Answer:

Mengobrol secara privat kepada tiap mahasiswa yang terlibat, menanyakan apakah sebelumnya ada tutor dan jika bisa cek rekaman tubay.

Case 6: Obvious Plagiarism Terdapat mahasiswa yang terbukti, berdasarkan sebuah detail yang sepertinya lupa diganti, mengumpulkan jawaban temannya.

Answer:

Jika sudah dikumpulkan dan deadline sudah terlewat, akan di beritahu ke dosen terkait untuk penindaklanjutan.

5.1.5 Personal Integrity Commitment

Apakah kamu akan meresikokan reputasi personal, hubungan-hubungan interpersonal, maupun kestabilan organisasi non-akademik (seperti himpunan atau unit) untuk memastikan integritas akademik?

Answer:

Yes, even though, harusnya dengan memastikan integritas akademik itu juga mempertahankan reputasi personal dan khususnya di ITB harusnya tiap mahasiswa sadar sendiri.