

BAB 2

TINJAUAN PUSTAKA

2.1 Profil Perusahaan

Pada sub bab ini menjelaskan tentang profil tempat penelitian yaitu di PT Kahatex Cijerah.

2.1.1 Sejarah Perusahaan

Kahatex berdiri pada tahun 1979 oleh pengusaha bernama LH Song. Kahatex bergerak di bidang pembuatan tekstil seperti benang dan bahan. Seiring dengan perkembangan bisnisnya, Kahatex juga memulai usaha garmen, seperti membuat pakaian jadi dan kaos kaki. Banyaknya permintaan atas produknya, membuat Kahatex memutuskan untuk menambah jumlah pabriknya. Pabrik Kahatex terdapat di beberapa kota di Jawa Barat, mulai dari Sumedang, Cimahi dan Bandung. Dengan penambahan pabrik ini, produk-produk buatan Kahatex juga bertambah, tidak hanya tekstil dan garmen saja, namun juga memproduksi plastik.

Kantor pusat Kahatex berada di Tanah Abang. Untuk lokasi pabrik-pabrik Kahatex berada di Bandung. Untuk mendukung proses pabrik, Kahatex juga memiliki beberapa sertifikat ISO, seperti ISO 9001:2025 untuk standarisasi mutu, ISO 140001:2005 dan OHSAS 180001:2007 mengenai aspek keselamatan kerja untuk semua karyawannya.. Kahatex terus mengembangkan produk-produk buaatannya agar tetap bisa bersaing dengan produk impor. Kualitas produk buatan Kahatex yang baik, membuat banyak negara yang membeli bahan Kahatex. Produk buatan Kahatex sudah mulai diekspor hingga 80 negara dan dijual di dalam negeri [6].

2.1.2 Visi & Misi

VISI

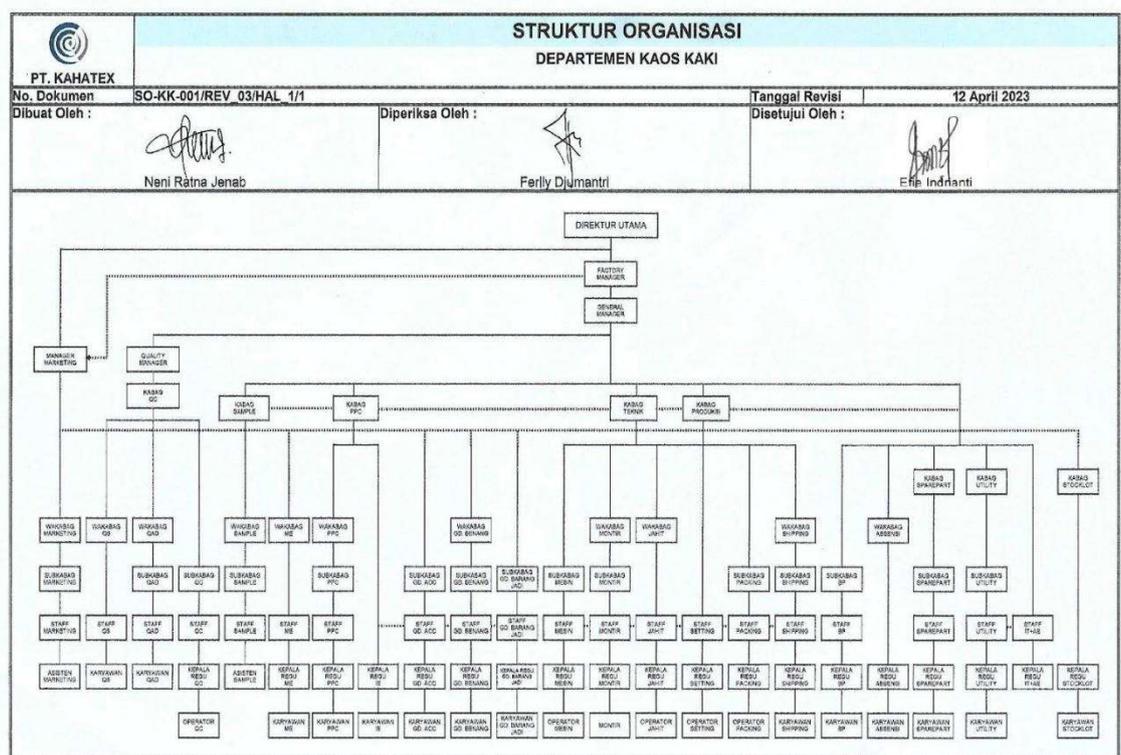
Menjadi produsen garmen dan tekstil terkemuka, yang menyediakan layanan berkualitas tinggi dan efisien.

MISI

Untuk membuat produk berkualitas tinggi dengan harga yang kompetitif.

2.1.3 Struktur Organisasi

Pada gambar dibawah merupakan representasi grafis dari bagaimana unit-unit organisasi saling terhubung dan berinteraksi.



Gambar 2. 1 Struktur Organisasi

Tabel 2. 1 Deskripsi Jobdesc

No	Jabatan	Job Desc
1	Kabag (Kepala Produksi)	1. Menerima dan memeriksa order No Model, style dan bahan baku yang akan digunakan pada order. 2. Menerima form jalan mesin per order. 3. Menjalankan mesin sesuai dengan ordernya. 4. Memberikan IK (instruksi kerja). 5. Mengontrol pekerjaan Operator Produksi (penjaga mesin) 6. Mengontrol pekerjaan Montir Mesin. 7. Mengontrol pekerjaan pengguna aplikasi TLS 8. Memantau report produksi perhari.
2	Operator Produksi	1. Mengetahui order yang berjalan pada mesin yang ia jaga. 2. Mengetahui jenis bahan baku yang digunakan pada order tersebut 3. Memberikan komplain apabila bahan baku/ order tidak sesuai. 4. Membalikan produk yang keluar dari mesin. 5. Memisahkan produk gagal/berhasil.
3	Montir	1. Melakukan service terhadap mesin yang bermasalah. 2. Membersihkan mesin 3. Memberikan kanban saat kondisi mesin berubah. 4. Melakukan <i>Maintenance</i> pada mesin setiap 1 bulan sekali.
4	Pengguna aplikasi TLS	1. Mengimport database order pada aplikasi. 2. Menginput data produksi mesin setiap hari. 3. Melakukan print report produksi perhari dan diberikan kepada Kepala Produksi.

2.2 Landasan Teori

2.2.1 Optimasi

Optimasi adalah proses mencari solusi terbaik dari suatu masalah, di mana "terbaik" didefinisikan berdasarkan kriteria tertentu. Dalam berbagai bidang, mulai dari matematika hingga ilmu komputer, optimasi memiliki peran penting. Menurut Nurrohman (2017) Optimalisasi adalah upaya meningkatkan kinerja pada suatu unit kerja ataupun pribadi yang berkaitan dengan kepentingan umum, demi tercapainya kepuasan dan keberhasilan dari penyelenggaraan kegiatan tersebut.

1. Optimasi Kinerja Aplikasi:

- Responsivitas: Mengurangi waktu loading aplikasi dan meningkatkan kecepatan akses data produksi.
- Stabilitas: Meningkatkan kehandalan aplikasi agar tidak sering mengalami crash atau error.
- Skalabilitas: Memastikan aplikasi dapat menangani peningkatan volume data produksi tanpa penurunan kinerja.

2. Optimasi Penggunaan Sumber Daya:

- Penggunaan CPU: Meminimalkan penggunaan CPU agar sumber daya komputasi dapat dialokasikan untuk tugas lain.
- Penggunaan Memori: Mengurangi penggunaan memori agar aplikasi dapat berjalan lebih efisien.
- Penggunaan Bandwidth: Mengoptimalkan penggunaan bandwidth untuk transfer data antara aplikasi dan cloud storage.

3. Optimasi Proses Bisnis:

- Efisiensi Pencatatan: Mempercepat dan menyederhanakan proses pencatatan data produksi.
- Akurasi Data: Meningkatkan akurasi data yang dicatat untuk menghindari kesalahan dalam pengambilan keputusan.
- Aksesibilitas Data: Memudahkan akses data produksi bagi seluruh pihak yang berkepentingan.

4. Optimasi Biaya:

- Pengurangan Biaya Operasional: Mengurangi biaya pemeliharaan server dan infrastruktur IT.
- Peningkatan Produktivitas: Meningkatkan produktivitas karyawan dengan mempermudah akses dan pengelolaan data produksi.

Konsep Optimasi dengan Teknologi Cloud Computing:

- Pemanfaatan Scalability Cloud: Menyesuaikan kapasitas komputasi sesuai dengan kebutuhan, sehingga biaya dapat dihemat.
- Penggunaan Layanan Database Cloud: Memilih layanan database cloud yang sesuai dengan kebutuhan aplikasi dan data yang akan disimpan.
- Implementasi Microservices Architecture: Membagi aplikasi menjadi layanan-layanan kecil yang dapat dikembangkan dan di-scale secara independen.
- Penerapan DevOps: Menggabungkan pengembangan dan operasi untuk mempercepat proses pengembangan dan deployment aplikasi.

2.2.1.1 Optimasi Query

Optimisasi Query adalah suatu proses untuk menganalisa query untuk menentukan sumber-sumber apa saja yang digunakan oleh query tersebut dan apakah penggunaan dari sumber tersebut dapat dikurangi tanpa merubah output.

2.2.2 Sistem

Sistem menurut Ludwig von Bertalanffy adalah kumpulan elemen-elemen yang saling berinteraksi, sehingga perubahan pada satu elemen dapat memengaruhi elemen-elemen lainnya. Sedangkan menurut Daniel Katz dan Robert L. Kahn sistem adalah sekelompok orang yang berinteraksi untuk mencapai tujuan tertentu[7].

2.2.3 Informasi

Menurut Kelly informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang. Definisi tersebut merupakan definisi informasi dalam pemakaian sistem informasi. Sedangkan menurut Carlos Coronel and Steven Morris (2016:4) informasi adalah hasil dari data mentah yang telah diproses untuk memberikan hasil di dalamnya. Dari pendapat diatas dapat disimpulkan bahwa informasi adalah hasil dari data mentah yang telah diolah sehingga mempunyai makna[8].

2.2.3 Sistem Informasi

Sistem informasi adalah kumpulan informasi. Ada bagian-bagian yang saling berhubungan dan saling bergantung. Hubungan ini berbentuk hubungan arus informasi, yaitu tingkat sistem organisasi yang mendukung informasi yang dibutuhkan oleh semua pihak.

Menurut McLeod Sistem informasi adalah suatu sistem yang memiliki kemampuan untuk mengumpulkan informasi dari semua sumber dan menggunakan berbagai media untuk menampilkan informasi.

2.2.4 Produksi

Produksi adalah suatu proses untuk mengubah input menjadi output yang akan menambah nilai guna suatu barang. Fungsi produksi menurut Soekartawi adalah hubungan fisik antara input dan output yaitu antara variabel yang dijelaskan berupa output dan variabel yang menjelaskan berupa input. Fungsi produksi juga dapat diartikan suatu hubungan yang ketergantungan antara tingkat input yang digunakan dalam proses produksi dengan tingkat output yang dihasilkan dari proses produksi.

2.2.5 Pencatatan Data

Pengertian Pencatatan menurut Mulyadi adalah suatu urutan ketiga klerikal biasanya melibatkan beberapa orang dalam suatu departemen atau lebih yang dibuat untuk menjamin penanganan secara seragam terhadap transaksi perusahaan yang terjadi berulang-ulang.

2.2.6 Job Queue

Job Queue atau Antrian Pekerjaan adalah sistem yang mengelola urutan tugas-tugas yang harus dieksekusi. Tugas-tugas ini disebut *job* atau pekerjaan. *Job queue* memastikan bahwa pekerjaan diproses secara berurutan, mencegah konflik, dan meningkatkan efisiensi[9].

2.2.6.1 Manfaat Job Queue

1. Meningkatkan efisiensi: *Job queue* membantu memastikan bahwa pekerjaan diproses secara optimal, mencegah konflik sumber daya, dan meminimalkan waktu.
2. Meningkatkan skalabilitas: *Job queue* memungkinkan sistem untuk menangani lebih banyak pekerjaan dengan mendistribusikan pekerjaan di beberapa server atau proses.
3. Meningkatkan keandalan: *Job queue* memastikan bahwa pekerjaan tidak hilang jika terjadi kegagalan sistem.

2.2.6.2 Cara Kerja Job Queue

1. Menambahkan tugas ke antrian, yaitu tugas dapat ditambahkan ke antrian dengan berbagai cara, seperti melalui API web, CLI (command-line interface), atau secara manual. Ketika tugas ditambahkan ke antrian, tugas tersebut akan diberi ID unik dan ditempatkan di akhir antrian.
2. Memproses tugas, yaitu worker secara berkala akan mengambil tugas dari antrian dan memprosesnya. Worker dapat menjalankan tugas secara sinkron

- (menunggu tugas selesai sebelum mengambil tugas lain) atau secara asinkron (melanjutkan ke tugas berikutnya tanpa menunggu tugas sebelumnya selesai).
3. Menangani kegagalan, yaitu jika terjadi kesalahan saat memproses tugas, tugas tersebut dapat dimasukkan kembali ke antrian untuk di coba lagi nanti. Jumlah percobaan ulang dapat dikonfigurasi, dan setelah batas percobaan ulang tercapai, tugas tersebut dapat ditandai sebagai gagal dan diberitahukan kepada administrator.
 4. Memantau antrian, penting untuk memantau antrian job untuk memastikan bahwa semua tugas diproses dengan lancar. Alat pemantauan dapat digunakan untuk melacak jumlah tugas dalam antrian, waktu tunggu rata-rata, dan tingkat kegagalan.

2.2.7 Internet

Internet adalah singkatan dari istilah Internet. Internet adalah jaringan komputer global atau global yang sangat besar dan besar yang menghubungkan pengguna komputer di suatu negara dengan negara lain di dunia, terdapat berbagai sumber informasi dari statis sampai dinamis. interaktif. Internet menggunakan Internet Protocol Suite (TCP / IP) untuk menyediakan layanan kepada miliaran pengguna di seluruh dunia. Dan TCP / IP adalah protokol (aturan) untuk pertukaran paket (protokol komunikasi pertukaran paket)[10].

2.2.8 Konsep Dasar Internet

Konsep dasar internet[11].

1. **TCP/IP (*Transmission Control Protocol/Internet Protocol*)**: Protokol ini merupakan dasar komunikasi di Internet. TCP bertanggung jawab untuk memecah pesan menjadi paket-paket data yang dapat dikirimkan melalui jaringan, sedangkan IP bertanggung jawab untuk mengarahkan paket-paket tersebut ke tujuan yang tepat.

2. **URL (*Uniform Resource Locator*)**: URL adalah alamat yang digunakan untuk mengidentifikasi lokasi suatu sumber daya di Internet, seperti halaman web, gambar, atau file.
3. **HTTP (*Hypertext Transfer Protocol*)**: HTTP adalah protokol komunikasi yang digunakan untuk mentransfer halaman web dari server ke browser pengguna.
4. **HTML (*Hypertext Markup Language*)**: HTML adalah bahasa markup yang digunakan untuk membuat halaman web. Halaman web dibuat dengan menambahkan tag-tag HTML ke teks biasa untuk mengatur tata letak dan menambahkan elemen-elemen seperti teks, gambar, dan tautan.
5. **DNS (*Domain Name System*)**: DNS adalah sistem yang menghubungkan alamat IP dengan nama domain yang mudah diingat. Misalnya, DNS menerjemahkan nama domain seperti "google.com" menjadi alamat IP yang sesuai.
6. **WWW (*World Wide Web*)**: WWW adalah sistem informasi yang terdiri dari berbagai halaman web yang terhubung melalui Internet. Ini memungkinkan pengguna untuk menjelajahi dan mengakses berbagai sumber daya di Internet.

2.2.9 Website

Website dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang diseluruh dunia. Halaman website dibuat menggunakan Bahasa standar yaitu HTML. Skrip HTML ini akan diterjemahkan oleh web browser sehingga dapat ditampilkan dalam bentuk informasi yang dapat dibaca oleh semua orang.

Menurut Rohi Abdulloh Jenis-jenis *website* secara umum, *website* dibagi menjadi 3 jenis, yaitu :

1. *Website* Statis

Website statis yaitu jenis website yang isinya tidak diperbaharui secara berkala, sehingga isinya dari waktu ke waktu akan selalu tetap. *website* jenis ini biasanya hanya digunakan untuk menampilkan profil dari pemilik *website* seperti profil perusahaan atau organisasi.

2. *Website* Dinamis

Website dinamis yaitu jenis *website* yang isinya terus diperbaharui secara berkala oleh pengelola web atau pemilik *website*. *Website* jenis ini banyak dimiliki oleh perusahaan atau perorangan yang aktivitas bisnisnya memang berkaitan dengan internet. Contoh paling mudah dari *website* jenis ini yaitu web blog dan *website* berita.

3. *Website* Interaktif

Website interaktif pada dasarnya termasuk dalam kategori *website* dinamis, dimana isi informasinya selalu diperbaharui dari waktu ke waktu. Hanya saja, isi informasi tidak hanya diubah oleh pengelola *website* tetapi lebih banyak dilakukan oleh pengguna *website* itu sendiri. Contoh *website* jenis ini yaitu *website* jejaring sosial seperti *facebook*, *twitter* atau *website marketplace* seperti *bukalapak*, *tokopedia* dan sebagainya.

2.2.10 Cloud Computing

Dalam istilah yang paling sederhana, komputasi awan berarti menyimpan dan mengakses data dan program melalui Internet dari lokasi atau komputer yang jauh, bukan dari hard drive komputer kita. Lokasi jauh yang disebut ini memiliki beberapa sifat seperti skalabilitas, elastisitas, dll., Yang sangat berbeda dari mesin jarak jauh yang sederhana. Awan hanyalah metafora untuk Internet. Saat kami menyimpan data atau menjalankan program dari hard drive komputer lokal, itu disebut penyimpanan dan komputasi lokal. Agar dapat dianggap komputasi awan, kita perlu mengakses data atau program kita melalui Internet. Hasil akhirnya adalah sama; Namun, dengan koneksi online, komputasi awan dapat dilakukan di mana saja, kapan saja, dan dengan perangkat apa saja.

Definisi formal komputasi awan berasal dari *National Institute of Standards and Technology (NIST)*: “Komputasi awan adalah model untuk

memungkinkan akses jaringan di mana-mana, nyaman, sesuai permintaan ke kumpulan bersama sumber daya komputasi yang dapat dikonfigurasi (misalnya, jaringan, server, penyimpanan, aplikasi, dan layanan) yang dapat disediakan dan dirilis dengan cepat dengan upaya manajemen minimal atau interaksi penyedia layanan. Model cloud ini terdiri dari lima karakteristik penting, tiga model layanan, dan empat model penyebaran. Ini berarti bahwa sumber daya atau infrastruktur komputasi, baik itu perangkat keras server, penyimpanan, jaringan, atau perangkat lunak aplikasi semuanya tersedia dari vendor cloud atau situs/premis penyedia, dapat diakses melalui Internet dari lokasi jarak jauh dimanapun dan oleh perangkat komputasi lokal mana pun.

Selain itu, penggunaan atau aksesibilitas adalah biaya hanya untuk tingkat penggunaan kepada pelanggan berdasarkan kebutuhan dan permintaan mereka, juga dikenal sebagai model *pay-as-you-go* atau *pay-as-per-use*. Jika kebutuhan lebih banyak, lebih banyak sumber daya komputasi kuantum disediakan (penyediaan dengan elastisitas) oleh penyedia. Upaya manajemen minimal menyiratkan bahwa di sisi pelanggan, pemeliharaan sistem komputasi sangat minim karena mereka harus melihat tugas-tugas ini hanya untuk perangkat komputasi [12].

Layanan cloud memiliki tiga karakteristik khusus yang membedakannya dari hosting tradisional. Layanan ini dijual berdasarkan permintaan, yang biasanya per menit atau per jam dan bersifat elastis, user boleh memiliki berapapun layanan yang diinginkan sesuai waktu yang diberikan, dan layanan ini dikelola penuh oleh provider (pelanggan hanya perlu komputer dan akses Internet). Inovasi-inovasi yang signifikan dalam hal virtualisasi dan distributed computing, termasuk juga peningkatan akses ke Internet berkecepatan tinggi dan perbaikan ekonomi, telah meningkatkan ketertarikan orang kepada cloud computing. Berikut adalah 3 jenis layanan cloud computing :

1. ***Software as a Service (SaaS)*** Model ini memberikan user sebuah aplikasi bisnis yang diakses melalui web. Umumnya user melakukan sewa aplikasi sehingga dapat mengakses fitur-fitur yang ada, user juga dapat membayar biaya tambahan untuk mengakses kapasitas / fitur yang lebih banyak.

Dengan naiknya teknologi web seperti AJAX, memungkinkan web memiliki tingkat user experience yang mendekati desktop application.

2. **Platform as a Service (PaaS)** Disini, konsumen diberikan sebuah platform untuk pengembangan sampai implementasi sistem. Konsumen harus untuk membuat dan mengimplementasikan sistemnya sendiri. Umumnya tools untuk development disediakan dalam bentuk web application.
3. **Infrastructure as a Service (IaaS)** Model ini hanya memberikan user aspek dasar dari computing seperti jaringan, *storage*, *processor* untuk *computing*. Infrastruktur komputasi cloud sangat bergantung pada virtualisasi.

2.2.11 AWS

Amazon Web Services (AWS) adalah *cloud* yang paling lengkap dan diadopsi secara luas di dunia, menawarkan lebih dari 200 layanan berfitur lengkap dari pusat data secara global. Jutaan pelanggan termasuk beberapa perusahaan rintisan dengan pertumbuhan tercepat, perusahaan terbesar, dan lembaga pemerintah terkemuka—menggunakan AWS untuk menurunkan biaya, menjadi lebih tangkas, dan berinovasi lebih cepat. AWS secara signifikan memiliki lebih banyak layanan, dan lebih banyak fitur dalam layanannya, dibandingkan penyedia cloud lainnya—mulai dari teknologi infrastruktur seperti penghitungan, penyimpanan, dan basis data—hingga teknologi yang berkembang, seperti pembelajaran mesin dan kecerdasan buatan, data lake dan analitik, dan Internet of Things. Ini membuat lebih cepat, lebih mudah, dan lebih hemat biaya untuk memindahkan aplikasi Anda yang sudah ada ke cloud dan build hampir semua yang dapat Anda bayangkan[13].

2.2.11.1 Layanan AWS

Berikut beberapa layanan yang disediakan oleh AWS:

1. **Amazon EC2 (*Elastic Compute Cloud*)**: Layanan ini menyediakan kapasitas komputasi *resizable* di *cloud*. Pengguna dapat menyewa server virtual (*instance*) dan menjalankan aplikasi mereka di dalamnya.

2. **Amazon S3 (*Simple Storage Service*)**: Layanan penyimpanan objek yang skalabel dan tahan terhadap kesalahan. Amazon S3 memungkinkan pengguna untuk menyimpan dan mengambil data dari mana saja di web.
3. **Amazon RDS (*Relational Database Service*)**: Layanan manajemen basis data yang memungkinkan pengguna untuk dengan mudah mengatur, mengoperasikan, dan menskalakan basis data relasional di cloud. RDS mendukung berbagai jenis basis data seperti MySQL, PostgreSQL, Oracle, SQL Server, dan MariaDB.
4. **Amazon Lambda**: Layanan komputasi tanpa server yang memungkinkan pengguna untuk menjalankan kode tanpa harus mengelola server. Lambda menjalankan kode dalam respons terhadap peristiwa seperti permintaan HTTP, pembaruan basis data, atau peristiwa dari layanan AWS lainnya.
5. **Amazon DynamoDB**: Layanan basis data NoSQL yang menyediakan penyimpanan data yang cepat, skalabel, dan fleksibel. DynamoDB cocok untuk aplikasi dengan kebutuhan kinerja tinggi dan skalabilitas otomatis.
6. **Amazon SNS (*Simple Notification Service*)**: Layanan notifikasi yang memungkinkan pengguna untuk mengirim pesan push kepada pelanggan atau aplikasi. SNS mendukung pengiriman pesan ke berbagai platform seperti perangkat mobile, email, SMS, dan banyak lagi.
7. **Amazon SQS (*Simple Queue Service*)**: Layanan antrian yang memungkinkan pengguna untuk mengantrikan pesan-pesan yang harus diproses oleh aplikasi atau sistem. SQS membantu dalam membangun sistem yang tahan terhadap lonjakan beban dan memisahkan komponen-komponen aplikasi.
8. **Amazon ECS (*Elastic Container Service*)**: Layanan manajemen kontainer yang memungkinkan pengguna untuk menjalankan dan mengelola aplikasi berbasis Docker di lingkungan cloud AWS.
9. **Amazon CloudFront**: Layanan CDN (*Content Delivery Network*) yang mempercepat pengiriman konten web dengan mendistribusikan konten ke server-server di seluruh dunia.

10. **Amazon Route 53:** Layanan DNS (**Domain Name System**) yang dapat diandalkan dan skalabel yang membantu pengguna untuk mengarahkan lalu lintas web ke aplikasi dan sumber daya di AWS atau di luar AWS.

2.2.11.2 Fitur

Berikut merupakan analisis Fitur yang digunakan pada aws untuk optimasi aplikasi pencatatan data produksi ini :

Tabel 2. 2 Fitur AWS

Layanan AWS	Fitur	Deskripsi
EC2	Auto Scaling	Memungkinkan otomatisasi penambahan atau pengurangan instance berdasarkan metrik yang telah ditentukan, seperti penggunaan CPU.
	Elastic Load Balancing (ELB)	Mendistribusikan lalu lintas aplikasi ke beberapa instance EC2 untuk memastikan aplikasi tetap tersedia dan dapat diakses dengan baik.
	Amazon Machine Images (AMI)	Menggunakan AMI untuk memulai instance dengan konfigurasi yang telah ditentukan, memastikan konsistensi dan efisiensi dalam penyebaran aplikasi.
RDS	Multi-AZ Deployments	Menyediakan replikasi database di beberapa Availability Zones untuk meningkatkan ketersediaan dan ketahanan terhadap kegagalan.
	Automated Backups	Mendukung backup otomatis yang dapat diatur sesuai kebutuhan retensi data dan pemulihan point-in-time.

2.2.11.3 Keamanan

Berikut merupakan analisis keamanan yang ada pada AWS :

Tabel 2. 3 Keamanan AWS

Layanan AWS	Fitur	Deskripsi
-------------	-------	-----------

EC2	Enkripsi Data	Mendukung enkripsi data saat transit dan saat penyimpanan
	Identity and Access Management (IAM)	Mengelola akses ke resource EC2 menggunakan IAM untuk memberikan izin berbasis peran.
	Key Management Service (KMS)	Untuk mengelola kunci enkripsi dengan aman menggunakan AWS KMS.
RDS	Enkripsi Data	Mengenkripsi data saat penyimpanan menggunakan Amazon KMS, serta enkripsi data saat transit.
	Network Isolation	Menggunakan Amazon VPC untuk mengisolasi database dalam jaringan virtual yang aman.

2.2.11.4 Skalabilitas dan Fleksibilitas

Berikut merupakan analisis Skalabilitas dan Fleksibilitas pada aws :

Tabel 2. 4 SkalabilitasAWS

Layanan AWS	Fitur	Deskripsi
EC2	Vertical Scaling	Meningkatkan sumber daya (CPU, memori) pada instance yang ada.
	Horizontal Scaling	Menambah jumlah instance untuk menangani peningkatan beban kerja.
	Reserved Instances	Menghemat biaya dengan komitmen jangka panjang melalui Reserved Instances.
RDS	Storage Scaling	Peningkatan otomatis kapasitas penyimpanan berdasarkan kebutuhan aplikasi.
	Compute Scaling	Meningkatkan ukuran instance database untuk menangani beban kerja yang lebih besar.
	On-Demand dan Reserved Pricing	Memilih antara pembayaran berdasarkan penggunaan atau komitmen jangka panjang untuk penghematan biaya.

2.2.11.5 Biaya

Berikut analisis biaya pada layanan AWS :

Tabel 2. 5 Biaya AWS

Layanan AWS	Fitur	Deskripsi
EC2	Pay-as-you-go	Pembayaran berdasarkan penggunaan yang sebenarnya, memungkinkan penghematan biaya dengan hanya membayar sumber daya yang digunakan.
	Reserved Instances dan Spot Instances	Mengurangi biaya dengan komitmen jangka panjang atau menggunakan kapasitas yang tidak terpakai dengan harga lebih rendah melalui Spot Instances.
RDS	Pay-as-you-go	Pembayaran berdasarkan penggunaan yang sebenarnya, memungkinkan efisiensi biaya
	Reserved Instances	Mengurangi biaya dengan komitmen jangka panjang melalui Reserved Instances.
	Backup Storage	Biaya penyimpanan untuk backup otomatis yang ditentukan oleh kebijakan retensi.

Dan berikut merupakan perkiraan harga dari instance yang diambil, perkiraan ini diambil dengan estimasi yang AWS berikan pada <https://calculator.aws/> :

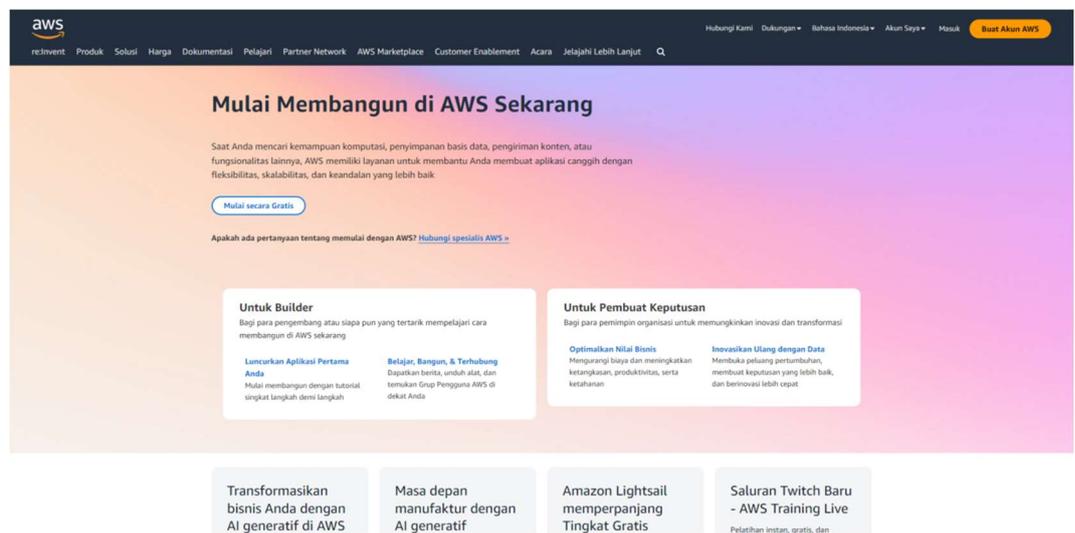
Layanan	Tipe Instance/Storage	Harga Per Jam	Jumlah Jam/Bulan	Total Biaya Per Bulan	Rupiah
Amazon EC2	t3.medium (2 vCPUs, 4 GB RAM)	\$0.0416	720 (24 jam * 30 hari)	\$29.95	Rp449.250
Amazon RDS	db.t3.medium (2 vCPUs, 4 GB RAM)	\$0.037	720 (24 jam * 30 hari)	\$26.64	Rp399.600
	Storage 100 GB GP2 SSD	\$0.115 per GB	100 GB	\$11.50	Rp172.500
ElastiCache	cache.t3.medium (2 vCPUs, 2.5 GB RAM)	\$0.026	720 (24 jam * 30 hari)	\$18.72	Rp280.800

Total Biaya	\$86.81	Rp1.302.150
-------------	---------	-------------

2.2.11.6 Membuat Akun AWS

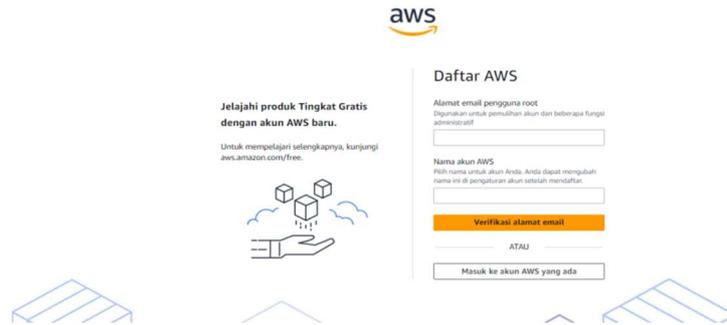
Berikut langkah-langkah pada pembuatan akun AWS:

1. Membuka situs AWS, dengan mengunjungi situs web resmi AWS di <https://aws.amazon.com/>.



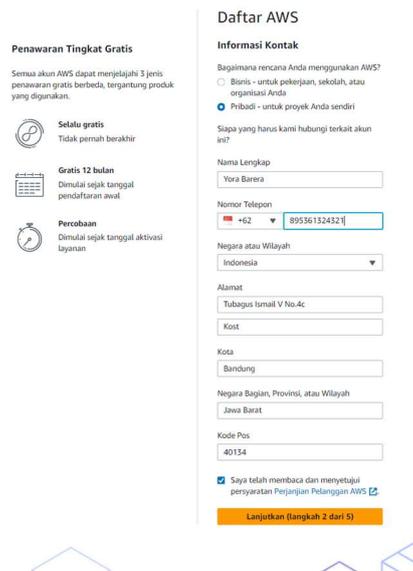
Gambar 2. 2 Interface web resmi AWS

2. Klik "Create an AWS Account", tombol ini terletak di pojok kanan atas halaman utama AWS. Klik tombol tersebut untuk memulai proses pembuatan akun. Berikut halaman untuk membuat akun AWS :



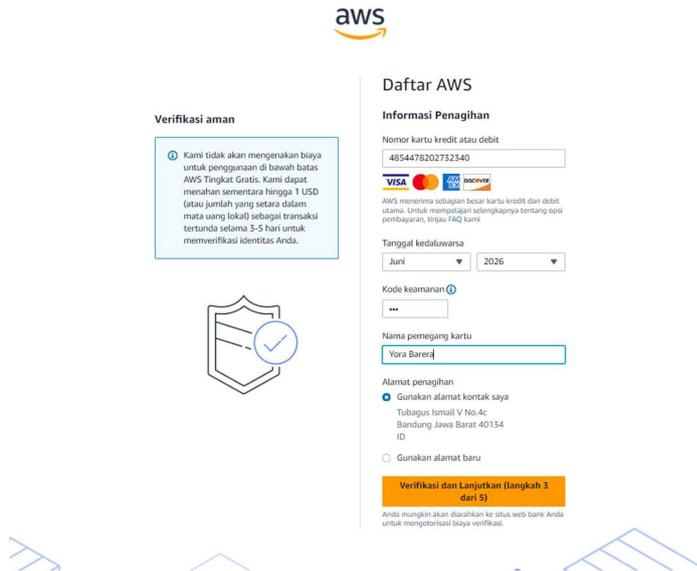
Gambar 2. 3 Interface web pendaftaran akun AWS

3. Masukkan informasi pribadi dengan mengisi formulir pendaftaran.



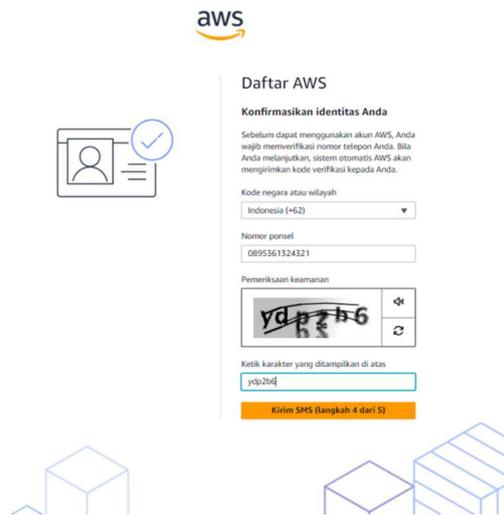
Gambar 2. 4 Interface formulir pendaftaran AWS

4. Isi Informasi penagihan pada kartu kredit.



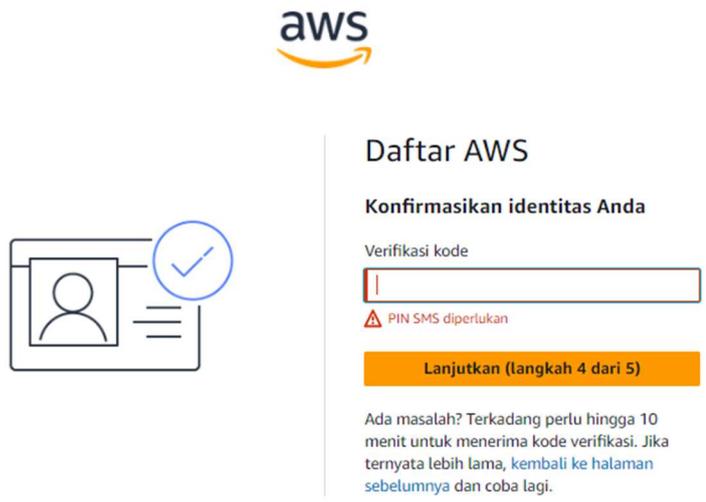
Gambar 2. 5 Interface CC

5. Mengkonfirmasi identitas



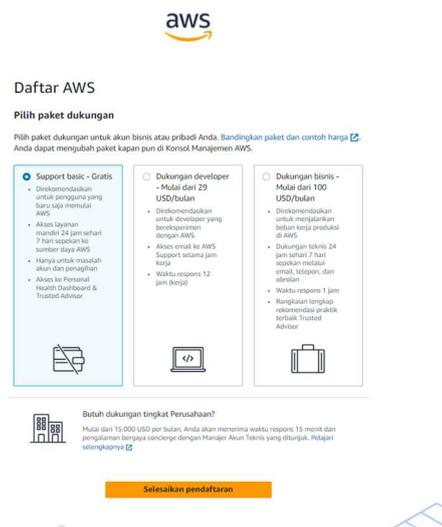
Gambar 2. 6 Interface Konfirmasi Identitas

- Masukan kode verifikasi yang dikirimkan oleh AWS ke no telpon yang sudah terdaftar.



Gambar 2. 7 Interface Verifikasi

7. Memilih paket yang akan digunakan dan menyelesaikan pendaftaran.



Gambar 2. 8 Interface Pilih Paket

8. Akun telah selesai dibuat



Gambar 2. 9 Interface Selesai Daftar

2.2.12 EC2

Amazon Elastic Compute Cloud (Amazon EC2) adalah layanan yang disediakan oleh Amazon Web Services (AWS) yang memungkinkan pengguna untuk menjalankan aplikasi komputasi di lingkungan cloud. EC2 memberikan kapasitas komputasi yang resizable di cloud, memungkinkan pengguna untuk menskalakan sumber daya komputasi sesuai kebutuhan mereka.

2.2.12.1 Fitur Utama EC2

Berikut adalah beberapa fitur utama dan jenis instance EC2 yang tersedia di AWS:

1. **Elasticity and Scalability:** EC2 memungkinkan penyesuaian kapasitas komputasi dengan cepat dan mudah untuk menangani perubahan dalam beban kerja.
2. **Flexible Instance Types:** Menyediakan berbagai tipe instans yang dioptimalkan untuk berbagai macam beban kerja, termasuk instans yang dioptimalkan untuk komputasi, memori, storage, dan GPU.
3. **Pay-As-You-Go:** Pengguna hanya membayar kapasitas yang mereka gunakan, yang membantu mengontrol biaya.

4. **Integration with AWS Services:** EC2 terintegrasi dengan layanan AWS lainnya seperti S3, RDS, VPC, dan CloudWatch.
5. **Security:** EC2 menyediakan fitur keamanan tingkat tinggi dengan penggunaan Virtual Private Clouds (VPC), kontrol akses berbasis peran (IAM), dan enkripsi data.

2.2.12.2 Jenis Instance EC2

EC2 menawarkan berbagai jenis instance yang dirancang untuk memenuhi berbagai kebutuhan komputasi. Beberapa kategori utama instance EC2 meliputi:

1. **General Purpose Instances:**
 - a. T3, T3a, T4g: Instans burstable yang menawarkan keseimbangan antara CPU, memori, dan jaringan dengan biaya rendah.
 - b. M5, M5a, M5n, M6g: Instans serbaguna yang menawarkan keseimbangan antara komputasi, memori, dan jaringan.
2. **Compute Optimized Instances:**
 - a. C5, C5n, C6g: Dirancang untuk beban kerja yang memerlukan kinerja komputasi tinggi seperti batch processing, machine learning, dan gaming.
3. **Memory Optimized Instances:**
 - a. R5, R5a, R5n, R6g: Dirancang untuk aplikasi yang memerlukan memori besar seperti database in-memory, analitik big data, dan caching.
 - b. X1, X1e: Instans dengan kapasitas memori sangat besar untuk beban kerja seperti SAP HANA dan aplikasi analitik real-time.
4. **Storage Optimized Instances:**
 - a. I3, I3en: Dirancang untuk aplikasi yang memerlukan throughput IO yang tinggi dan latensi rendah seperti database NoSQL dan big data.
5. **Accelerated Computing Instances:**
 - a. P3, P4: Dirancang untuk komputasi yang memerlukan GPU untuk deep learning, AI, dan simulasi.

- b. G4, G5: Dirancang untuk beban kerja yang memerlukan GPU untuk inferensi machine learning dan aplikasi grafis.
- 6. Arm-based Instances:
 - a. A1, T4g, M6g, C6g, R6g: Instans berbasis prosesor AWS Graviton yang menawarkan efisiensi biaya tinggi untuk skala horizontal beban kerja.

2.2.12.3 Contoh Penggunaan EC2

1. Web Hosting: Menggunakan EC2 untuk menjalankan server web dan aplikasi web.
2. Big Data Processing: Memanfaatkan instans dengan kapasitas tinggi untuk analitik big data dan pemrosesan batch.
3. Machine Learning: Menggunakan instans GPU untuk training dan inferensi model machine learning.
4. Database Hosting: Menjalankan database relasional dan NoSQL dengan kinerja tinggi.

2.2.12.4 Mengaktifkan EC2

Berikut merupakan langkah-langkah untuk mengaktifkan AWS EC2 :

1. Buka [AWS Management Console](#).
2. Masukkan kredensial AWS Anda.
3. Di konsol AWS, pilih "Services" di pojok kiri atas.
4. Pilih "EC2" di bagian Compute.
5. Klik tombol "Launch Instance" pada dashboard EC2.
6. Buat Key Pair ED25519
7. Klik "Create key pair" dan unduh file .pem
8. Kembali ke dashboard EC2.
9. Klik tombol "Launch Instance".
10. Pilih "Ubuntu Server" dari daftar AMI yang tersedia (misalnya, Ubuntu Server 20.04 LTS).
11. Pilih jenis instance yang diinginkan (misalnya, t2.micro untuk penggunaan gratis).

12. Konfigurasi pengaturan instance sesuai kebutuhan, atau biarkan pengaturan default untuk opsi dasar.
13. Klik "Next: Add Storage" setelah selesai.
14. Atur ukuran dan tipe penyimpanan sesuai kebutuhan.
15. Klik "Next: Add Tags" setelah selesai.
16. (Opsional) Tambah tag untuk mengorganisir dan mengelola instance.
17. Klik "Next: Configure Security Group" setelah selesai.
18. Buat security group baru atau pilih yang sudah ada.
19. Tambahkan aturan untuk mengizinkan akses SSH (port 22).
20. Klik "Review and Launch" setelah selesai.
21. Tinjau pengaturan instance Anda.
22. Klik "Launch".
23. Pilih key pair "MyED25519Key" yang telah Anda buat sebelumnya.
24. Centang kotak konfirmasi dan klik "Launch Instances".
25. Klik "View Instances" untuk melihat instance yang sedang diaktifkan.
26. Tunggu beberapa saat hingga status instance berubah menjadi "running".

2.2.13 RDS Instance

Menurut Amazon Web Services (AWS), **RDS Instance** atau **DB Instance** adalah lingkungan database yang terisolasi yang berjalan di cloud. DB Instance ini merupakan elemen dasar dari layanan Amazon Relational Database Service (Amazon RDS). Setiap DB Instance dapat berisi satu atau lebih database yang dibuat oleh pengguna dan dapat diakses menggunakan alat dan aplikasi yang sama yang digunakan untuk mengakses database yang berdiri sendiri. Amazon RDS mendukung berbagai mesin database, termasuk Amazon Aurora, MySQL, MariaDB, PostgreSQL, Oracle, dan Microsoft SQL Server. AWS menangani banyak tugas manajemen untuk DB Instance ini, seperti pencadangan, patching perangkat lunak, deteksi kegagalan otomatis, dan pemulihan.

2.2.13.1 Mengaktifkan RDS Instance

Berikut merupakan langkah-langkah untuk mengaktifkan RDS Instance :

1. Buka *AWS Management Console*.
2. Masukkan kredensial AWS.
3. Di konsol AWS, pilih "*Services*" di pojok kiri atas.
4. Pilih "RDS" di bagian Database.
5. Di dashboard RDS, klik tombol "*Create database*".
6. Pilih "*Standard create*" untuk opsi konfigurasi yang lebih lengkap.
7. Di bagian "*Engine options*", pilih "MySQL".
8. Pilih template sesuai kebutuhan Anda (misalnya, "*Free tier*" untuk penggunaan gratis, "*Production*" untuk penggunaan produksi).
9. Isi "*DB instance identifier*" (misalnya, "my-mysql-instance").
10. Pilih "*Master username*" (misalnya, "admin").
11. Masukkan dan konfirmasi "*Master password*".
12. Di bagian "DB instance class", pilih jenis instance sesuai kebutuhan (misalnya, db.t3.micro untuk penggunaan gratis).
13. Pilih jenis penyimpanan dan atur ukuran penyimpanan (misalnya, 20 GiB untuk penggunaan gratis).
14. Centang atau hapus centang "Enable storage autoscaling" sesuai kebutuhan.
15. Pilih "Multi-AZ deployment" jika membutuhkan high availability (opsional).
16. Atur pengaturan backup dan retention sesuai kebutuhan.
17. Pilih "Virtual Private Cloud (VPC)" yang diinginkan.
18. Pilih "Subnet group" jika diperlukan.
19. Atur "Public access" menjadi "Yes" jika ingin mengakses database secara publik.
20. Pilih "VPC security group" yang sesuai atau buat yang baru untuk mengatur akses ke RDS instance.
21. Konfigurasi parameter lainnya seperti "Availability Zone", "Database port", dll.
22. Di bagian "Database authentication", pilih metode autentikasi yang diinginkan (misalnya, Password authentication).

23. Konfigurasi pengaturan lainnya seperti "Initial database name", "Parameter group", "Option group", dll.
24. Tinjau semua pengaturan yang telah Anda konfigurasi.
25. Klik "Create database".
26. Tunggu beberapa menit hingga status instance berubah menjadi "available".

2.2.14 ElastiCache

Amazon ElastiCache adalah layanan web yang memudahkan pengaturan, pengelolaan, dan skala penyimpanan data dalam memori terdistribusi atau lingkungan cache di cloud. Layanan ini menyediakan solusi caching berperforma tinggi, dapat diskalakan, dan hemat biaya. Layanan ini juga membantu menghilangkan kompleksitas yang terkait deployment dan manajemen lingkungan cache terdistribusi.

2.2.15 Framework

Framework adalah sebuah struktur konseptual dasar yang digunakan untuk memecahkan sebuah permasalahan, bahkan isu-isu kompleks yang ada. Sebuah *framework* telah berisi sekumpulan arsitektur/konsep-konsep yang dapat mempermudah dalam pemecahan sebuah permasalahan[15].

Salah satu alasan mengapa orang menggunakan *framework* terutama dalam membangun sebuah aplikasi adalah kemudahan yang ditawarkan. Didalam sebuah *framework* biasanya sudah tersedia struktur aplikasi yang baik, standard coding, best practice dan design pattern , dan common function. Dengan menggunakan *framework* kita dapat langsung fokus kepada business process yang dihadapi tanpa harus berfikir banyak masalah struktur aplikasi, standar coding dan lain-lain

2.2.16 Codeigniter

CodeIgniter adalah sebuah web application *framework* yang bersifat open source digunakan untuk membangun aplikasi php dinamis. Tujuan utama pengembangan Codeigniter adalah untuk membantu developer untuk mengerjakan

aplikasi lebih cepat daripada menulis semua code dari awal. Codeigniter menyediakan berbagai macam library yang dapat mempermudah dalam pengembangan. CodeIgniter diperkenalkan kepada publik pada tanggal 28 februari 2006. CodeIgniter sendiri dibangun menggunakan konsep Model-View-Controller development pattern. CodeIgniter sendiri merupakan salah satu *framework* tercepat dibandingkan dengan *framework* lainnya[15].



Gambar 2. 10 Logo Codeigniter

2.2.17 Database

Menurut Abdulloh *Database* atau basis data adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi. Fungsi dari *database* menurut Ramadhan and Mukhaiyar adalah :

1. Mempermudah identifikasi data dengan cara pengelompokan data, salah satu contoh nya dengan pembuatan beberapa tabel atau field yang berbeda-beda.
2. Meminimalisir suatu data ganda.
3. Mempermudah penggunaan user dalam berbagai hal, misalnya pada saat penginputan data baru.
4. Penyimpanan secara digital.
5. Menjadi alternatif lain terkait masalah penyimpanan ruang dalam suatu aplikasi.

2.2.18 Mysql

MySQL adalah jenis *server database* yang sangat terkenal. *MySQL* menggunakan bahasa *SQL* untuk mengakses *database*. Lisensi *MySQL* adalah pengecualian untuk lisensi *FOSS*, dan ada juga versi komersial. Tag *Mysql* adalah "*Database open source* paling populer di dunia". *MySQL* tersedia di berbagai platform, termasuk versi *Windows* dan *Linux*. Dapat juga menggunakan software tertentu seperti *phpmyadmin* atau *mysqlqog* untuk mempermudah pengelolaan *MySQL*.

Menurut Setiawan dari jurnal Khairilet al, *MySQL* adalah *Database* yang menghubungkan skrip PHP menggunakan *query* yang sama dan perintah *escape* seperti *PHP*. Secara umum bahasa *SQL* terbagi menjadi tiga bagian:

A. DDL (*Data Definition Language*)

DDL digunakan untuk membuat objek dalam database seperti meja. Menggunakan DDL untuk menentukan tata letak baris, definisi kolom, dan kolom kunci, lokasi file dan strategi penyimpanan.

B. DML (*Data Manipulasi Language*)

Digunakan dalam manipulasi suatu tabel didalam *database* (menambah, mengedit, mencari dan menghapus). Perintah-perintah DML adalah *SELECT*, *INSERT*, *UPDATE* dan *DELETE*.

C. DCL (*Data Control Language*)

Digunakan untuk menangani masalah keamanan dalam *database server*, dan hak-hak pada objek-objek *database* untuk perintah yang digunakan.

2.2.19 HTML

HTML adalah singkatan dari *Hypertext Markup Language* dan merupakan bahasa web standar yang dikelola oleh W3C (*World Wide Web Consortium*) dalam bentuk tag yang membentuk semua elemen situs web. HTML bertindak sebagai pembangun struktur halaman situs web, mengatur elemen setiap situs web sesuai dengan tata letak yang diinginkan.

HTML Biasanya disimpan dalam file dengan ekstensi .html. Untuk memasukkan skrip HTML, gunakan editor teks seperti Notepad sebagai bentuk paling sederhana, atau gunakan editor teks khusus yang dapat mengenali setiap elemen skrip HTML dan menampilkannya dalam berbagai warna. Anda dapat membuat banyak hal serupa lebih mudah dibaca seperti *Notepad++*, *Sublime Text*, dan lainnya.

HTML terdiri atas unsur-unsur yang membentuk struktur skrip HTML yaitu :

a) Tag

Tag adalah simbol khusus (*markup*) berupa dua karakter “<” dan “>” yang mengapit suatu teks sebagai nama tag.

b) Atribut

Atribut yaitu property yang mengatur bagaimana elemen dari suatu tag akan ditampilkan.

a) Element

Element adalah bagian dari skrip HTML yang terdiri dari tag pembuka, isi elemen dan tag penutup.

HTML disimpan dengan ekstensi *.html, jika di dalamnya tidak mengandung skrip PHP. Jika di dalamnya mengandung skrip PHP, maka disimpan dengan ekstensi *.php. Dalam penulisan HTML ada beberapa tag yang wajib dituliskan dengan struktur yang sudah ditentukan yaitu sebagai berikut (Rohi Abdulloh, 2018) :

a) `<!DOCTYPE HTML>`, adalah tag awal dari setiap dokumen HTML yang berfungsi untuk menginformasikan pada browser bahwa dokumen yang sedang dibuka adalah dokumen HTML.

b) `<html> ... </html>`, adalah tag yang menunjukkan pembuka dan penutup dokumen HTML.

c) `<head> ... </head>`, adalah tag yang digunakan untuk menyimpan berbagai informasi tentang dokumen HTML.

d) `<title> ... </title>`, adalah tag yang digunakan untuk membuat judul website yang akan muncul di title bar browser.

- e) `<body> ... </body>`, adalah tag yang menunjukkan bagian utama website.

2.2.20 PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *open-source* yang umumnya digunakan untuk membangun aplikasi web dinamis dan interaktif. PHP dapat dijalankan pada server web dan dikombinasikan dengan HTML, CSS, dan JavaScript untuk membuat halaman web yang dinamis.

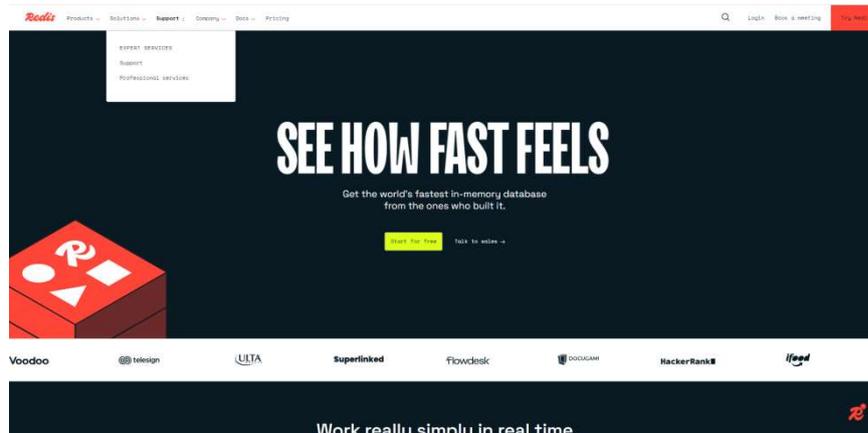
Menurut Supono & Putratama mengemukakan bahwa “PHP (PHP: hypertext preprocessor) adalah suatu bahasa pemrograman yang digunakan untuk menterjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat server-side yang ditambahkan ke HTML”.

2.2.21 PHP Redis (*Remote Dictionary Server*)

PHP Redis adalah ekstensi PHP yang memungkinkan Anda untuk berinteraksi dengan server Redis. Redis adalah database *in-memory open-source* yang terkenal dengan performanya yang tinggi dan skalabilitasnya yang baik[16]. Berikut adalah beberapa manfaat menggunakan PHP Redis:

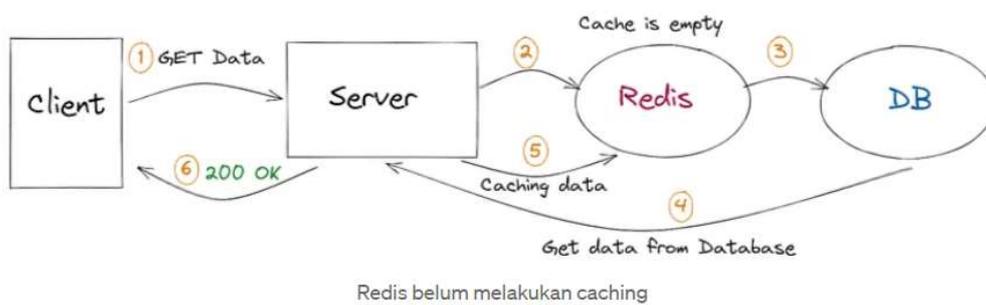
1. Performa Tinggi: Redis menyimpan data di memori, sehingga data dapat diakses dengan sangat cepat.
2. Skalabilitas Baik: Redis dapat di-scale secara horizontal dengan menambahkan lebih banyak server Redis.
3. Serbaguna: Redis dapat digunakan untuk berbagai keperluan, seperti caching, database, message broker, dan real-time data processing.
4. Mudah Digunakan: PHP Redis memiliki API yang sederhana dan mudah digunakan

Berikut merupakan web resmi dari Redis (<https://redis.io/>) :



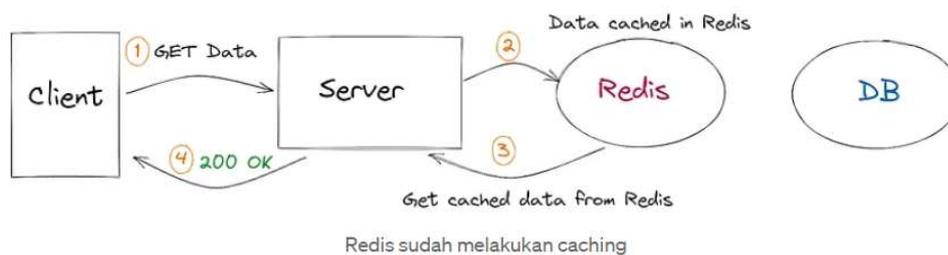
Gambar 2. 11 Website Resmi Redis

2.2.21.1 Cara Kerja Redis (*Remote Dictionary Server*)



Gambar 2. 12 Redis sebelum Caching

Sumber : *Caching Menggunakan Redis (Medium)* [18]



Gambar 2. 13 Redis setelah Caching

Sumber : *Caching Menggunakan Redis (Medium)* [18]

2.2.22 Navicat

Navicat adalah software manajemen database yang handal, termasuk untuk MySQL. Kelebihannya terletak pada GUI yang intuitif untuk konfigurasi database tanpa skrip, sehingga menghemat waktu dan usaha, terutama bagi pengguna yang tidak familiar dengan skrip. Navicat memungkinkan pengelolaan remote database MySQL dengan mudah melalui antarmuka visual yang user-friendly.

Navicat, sebagai software manajemen database yang handal, menawarkan berbagai fitur menarik untuk memudahkan Anda dalam mengelola database. Berikut beberapa fitur utamanya :

1. Editor Sql Visual

Menulis query SQL seringkali menjadi tantangan tersendiri. Navicat hadir dengan editor SQL visual. Fitur ini memungkinkan Anda untuk membangun query dengan mudah menggunakan drag-and-drop dan menu context-sensitive, tanpa perlu menghafal syntax SQL yang rumit.

2. Antarmuka User-Friendly

Antarmuka Navicat dirancang dengan mengedepankan kemudahan penggunaan. Tampilannya yang intuitif memudahkan Anda untuk bernavigasi dan menemukan fitur yang dibutuhkan dengan cepat.

3. Manajemen Database Lengkap

Navicat menawarkan berbagai tools untuk mengelola database secara menyeluruh. Anda dapat membuat, mengedit, dan menghapus tabel, serta mengelola data, view, stored procedure, dan user accounts dengan mudah.

4. **Migrasi Data Antar Database**

Memindahkan data antar database bisa menjadi proses yang rumit. Navicat memiliki fitur migrasi data yang memudahkan Anda untuk memindahkan data antar database yang berbeda platform dengan mudah dan aman.

5. **Backup dan Restore**

Keamanan data adalah hal yang krusial. Navicat menyediakan fitur backup dan restore untuk memastikan data Anda terlindungi. Anda dapat membuat jadwal backup secara otomatis dan mengembalikan data dengan mudah jika terjadi kerusakan atau kehilangan data.

6. **Enkripsi Data**

Navicat dilengkapi dengan fitur enkripsi data untuk melindungi data Anda dari akses yang tidak sah. Dengan fitur ini, data Anda akan dienkripsi dan didekripsi menggunakan algoritma yang kuat, sehingga menjamin keamanan data Anda.

7. **Dukungan Berbagai Platform**

Navicat tersedia untuk berbagai platform:

Windows : Microsoft Windows 7, Windows 8, Windows 8.1, Windows 10, Windows 11, Server 2012, Server 2016, Server 2019, Server 2022.

macOS : macOS 10.14 Mojave, macOS 10.15 Catalina, macOS 11 Big Sur, macOS 12 Monterey, macOS 13 Ventura, macOS 14 Sonoma.

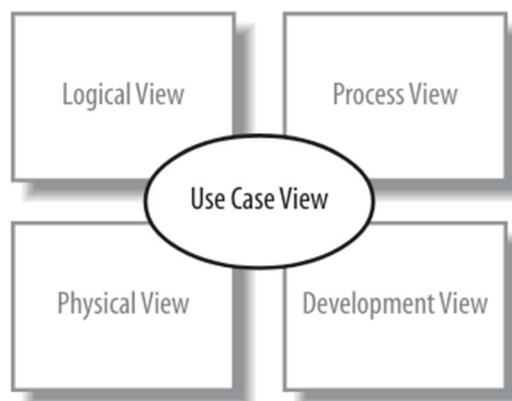
Linux : Debian 9, Debian 10, Debian 11, Ubuntu 18.04, Ubuntu 20.04, Ubuntu 22.04, Fedora 33, Fedora 34, Fedora 35, Linux Mint 19, Linux Mint 20, Linux Mint 21, Deepin 20, KylinOS .

2.2.23 UML

Unified Modeling Language (UML) dipergunakan untuk menunjukkan pengoperasian kerangka kerja, para peneliti memanfaatkan UML, yang menjadikan standar umum untuk menggambarkan secara visual bahasa dalam perangkat lunak dan pemrograman komputer. Oleh karena itu maka struktur yang digunakan memanfaatkan use case diagram untuk menggambarkan proses alur kerja dalam sistem. Kerangka kerja model *UML* adalah alat yang digunakan untuk mencirikan dan menggambarkan sistem perangkat lunak secara visual seperti tipe diagram standar untuk aplikasi komputer dan desain serta struktur sistem basis data.[17]

2.2.24 Use Case Diagram

Use case adalah sebuah teknik dalam rekayasa perangkat lunak yang digunakan untuk mendeskripsikan interaksi antara sistem dan penggunanya. Use case biasanya digunakan untuk merancang atau menganalisis kebutuhan fungsional sistem dengan fokus pada perilaku sistem dalam menanggapi permintaan dari aktor atau pengguna. Dalam praktiknya, use case digunakan untuk menggambarkan skenario penggunaan sistem dan memfasilitasi pemahaman yang lebih baik tentang kebutuhan sistem secara keseluruhan[17]. Berikut struktur penggambaran use case :



Gambar 2. 14 Struktur usecase

Sumber : Learning UML 2.0 [17]

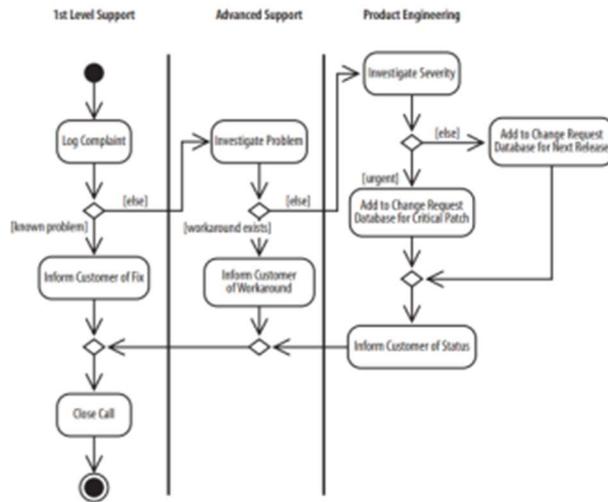
2.2.25 Use Case Scenario

Skenario use case adalah rangkaian langkah-langkah atau urutan kejadian yang menggambarkan bagaimana aktor atau pengguna berinteraksi dengan sistem untuk mencapai tujuan tertentu. Skenario use case memberikan deskripsi yang lebih rinci tentang alur kerja atau proses yang terjadi saat aktor menggunakan sistem untuk mencapai suatu hasil yang diinginkan. Skenario use case dapat digunakan untuk memahami secara mendalam bagaimana sistem akan digunakan dalam situasi-situasi tertentu dan membantu dalam perancangan dan pengembangan sistem yang responsif terhadap kebutuhan pengguna[17].

2.2.26 Activity Diagram

Dalam konteks *UML (Unified Modeling Language)*, sebuah activity diagram adalah jenis diagram yang digunakan untuk menggambarkan alur kerja atau aktivitas yang terjadi dalam suatu proses atau use case. Activity diagram menunjukkan serangkaian aktivitas dan tindakan yang dilakukan oleh objek atau aktor dalam sistem, serta bagaimana aktivitas tersebut berinteraksi satu sama lain dan mengalir dari satu ke yang lain[17].

Dalam konteks use case, activity diagram dapat digunakan untuk menunjukkan alur kerja atau langkah-langkah yang diperlukan untuk menyelesaikan suatu skenario use case. Ini membantu dalam memvisualisasikan bagaimana aktor atau pengguna berinteraksi dengan sistem dalam menyelesaikan tugas tertentu dan memberikan pemahaman yang lebih baik tentang proses yang terlibat.



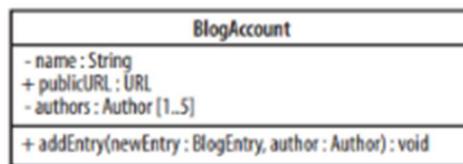
Gambar 2. 15 Activity Diagram

Sumber : Learning UML 2.0 [17]

2.2.27 Class Diagram

Dalam konteks *UML (Unified Modeling Language)*, sebuah class diagram adalah jenis diagram yang digunakan untuk menggambarkan struktur statis dari suatu sistem atau aplikasi. Class diagram menunjukkan kelas-kelas atau objek-objek dalam sistem, serta hubungan antara mereka seperti pewarisan (*inheritance*), *asosiasi*, *agregasi*, dan komposisi.

Class diagram memberikan pandangan tingkat tinggi tentang struktur sistem dan hubungan antara komponen-komponennya. Ini digunakan oleh pengembang perangkat lunak untuk merancang dan memodelkan struktur data dan perilaku sistem secara visual, sehingga memudahkan pemahaman tentang struktur program dan interaksi antara komponen-komponen dalam sistem.



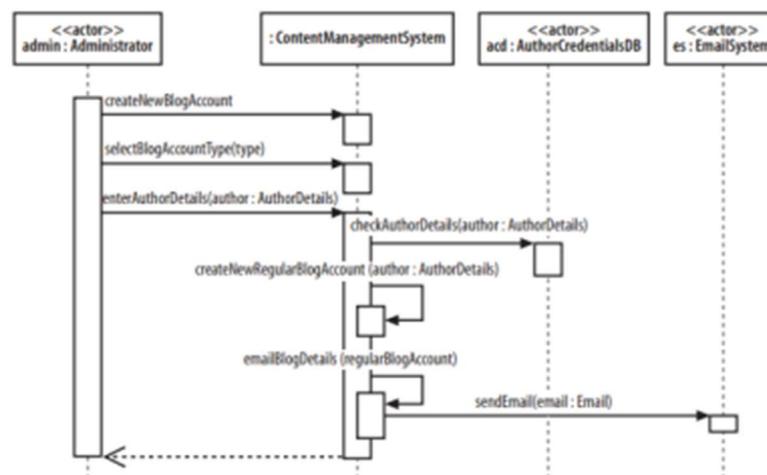
Gambar 2. 16 Class Diagram

Sumber : Learning UML 2.0 [17]

2.2.28 Sequence Diagram

Dalam konteks UML (*Unified Modeling Language*), sebuah sequence diagram adalah jenis diagram interaksi yang digunakan untuk menggambarkan bagaimana objek-objek dalam sistem saling berinteraksi dalam suatu skenario tertentu. Sequence diagram menunjukkan urutan pesan atau panggilan metode yang dikirim antara objek-objek, serta kapan pesan tersebut dikirim dan bagaimana objek-objek meresponsnya.

Sequence diagram biasanya digunakan untuk memodelkan interaksi antara objek-objek dalam suatu skenario use case atau proses bisnis. Mereka membantu dalam memvisualisasikan alur logika dari suatu interaksi, termasuk pesan yang dikirim antara objek-objek dan respon yang diterima, sehingga memudahkan pemahaman tentang bagaimana sistem berperilaku dalam berbagai situasi.



Gambar 2. 17 Sequence diagram

Sumber : Learning UML 2.0 [17]

2.2.29 BPMN

BPMN merupakan tahapan awal dalam rangkaian aktivitas pemodelan proses bisnis yang dikeluarkan oleh BPMI (*Business Process Management initiative*). Fungsi utama dari BPMN adalah untuk menyediakan suatu notasi

yang mudah dipahami oleh seluruh pemakai bisnis, mulai dari analis bisnis yang menciptakan draft awal proses, pengembang teknis yang bertanggung jawab dalam menerapkan teknologi yang mengimplementasikan proses, dan orang-orang bisnis (pihak manajemen) yang akan mengatur dan memonitor proses tersebut. BPMN menggambarkan suatu Diagram Proses Bisnis (BPD) yang didasarkan pada suatu flowcharting teknik yang dikhususkan untuk menciptakan model grafis dari operasi proses bisnis. Pada model ini dapat digambarkan kaitan antara aliran proses, serta dapat menampilkan keterkaitan organisasi lain yang terlibat, sehingga dapat dihasilkan bentuk diagram yang sederhana, mudah dipahami dan cukup menggambarkan proses secara keseluruhan. Dalam penggunaannya BPMN menyediakan sejumlah notasi dasar dan notasi-notasi tambahan. Notasi tambahan dapat digunakan untuk menggambarkan proses yang lebih kompleks[18].