



Memodernisasi strategi data perawatan kesehatan Anda

# AWS Bimbingan Preskriptif



# AWS Bimbingan Preskriptif: Memodernisasi strategi data perawatan kesehatan Anda

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

---

# Table of Contents

Pengantar .....	1
Gambaran umum .....	1
Tantangan data .....	2
Manfaat .....	4
Komponen-komponen .....	6
Menerapkan strategi .....	9
Contoh implementasi strategi .....	11
AI Generatif .....	13
Memenuhi tujuan pemangku kepentingan .....	16
Kesimpulan .....	17
Sumber daya .....	18
Lampiran A .....	19
Tingkatkan pengalaman pasien .....	19
Meningkatkan hasil di seluruh populasi .....	19
Mengurangi biaya dengan mengoptimalkan operasi .....	20
Mengotomatiskan tugas untuk meningkatkan pengalaman penyedia .....	21
Meningkatkan ekuitas dengan menggunakan data untuk memahami dan mengidentifikasi disparitas .....	22
Memajukan perawatan kesehatan melalui penelitian genom .....	22
Meningkatkan keberlanjutan sistem perawatan kesehatan .....	23
Lampiran B .....	25
Mengelola persetujuan untuk perawatan dan penelitian .....	26
Memberikan informasi pribadi kepada pasien .....	26
Menghubungkan pasien dengan uji klinis .....	27
Menyediakan portabilitas catatan kesehatan multimodal .....	27
Lampiran C .....	28
Meningkatkan kelincahan dan kemampuan untuk berinovasi .....	28
Mengurangi biaya operasional .....	29
Memodernisasi penyimpanan dan analitik data .....	29
Lampiran D .....	31
Kontributor .....	34
Riwayat dokumen .....	35
Glosarium .....	36
# .....	36

---

A .....	37
B .....	40
C .....	42
D .....	45
E .....	49
F .....	51
G .....	53
H .....	54
I .....	55
L .....	58
M .....	59
O .....	63
P .....	66
Q .....	69
R .....	69
D .....	72
T .....	76
U .....	77
V .....	78
W .....	78
Z .....	79
.....	lxxxi

# Modernisasi strategi data perawatan kesehatan Anda

Amazon Web Services ([kontributor](#))

November 2023 ([riwayat dokumen](#))

Dokumen ini memberikan strategi data untuk eksekutif perawatan kesehatan. Strategi ini mencakup bimbingan prosedural, organisasi, dan teknis bagi para pemimpin yang ingin memajukan misi institusi mereka dengan membuatnya lebih berbasis data.

## Gambaran umum

Sebagai eksekutif perawatan kesehatan, Anda bekerja di lingkungan yang menantang di mana data perawatan kesehatan tumbuh dalam ukuran, variasi, dan kompleksitas. Tim layanan kesehatan membutuhkan lebih banyak data, lebih cepat, dan kepatuhan terhadap peraturan memerlukan peningkatan ketelitian seputar penanganan data dan berbagi data. Aktor jahat yang canggih sering mengancam keamanan data. Terlepas dari tantangan ini, Anda harus meningkatkan perawatan pasien dan hasil pasien, membuat data tersedia untuk penelitian klinis atau translasi, dan mengoptimalkan biaya sehingga Anda dapat mempertahankan organisasi Anda dalam jangka panjang. Dokumen ini menyajikan bagaimana Anda dapat menggunakan data untuk mengatasi tantangan ini dan memenuhi tujuan Anda.

Strategi data kesehatan modern dapat membantu para pemimpin organisasi memenuhi banyak tujuan umum dan spesifik. Ini dapat membantu organisasi Anda untuk meningkatkan di semua aspek Tujuan [Quadruple](#). Misalnya, Anda dapat meningkatkan pengalaman pasien dengan meningkatkan komunikasi dan mengoptimalkan akses ke data mereka. Pengalaman dokter diperkaya dengan membuat data dapat diakses untuk penelitian, operasi, dan peningkatan kualitas dan keamanan. Otomatisasi alur kerja mendorong pengurangan biaya sekaligus meningkatkan efisiensi dan akses ke informasi penting bagi mereka yang membuat keputusan. Hasil pada tingkat individu dan populasi ditingkatkan dengan strategi data multimodal yang kohesif yang mempertimbangkan keseluruhan pengalaman pasien di dalam dan di luar organisasi perawatan kesehatan langsung.

# Tantangan data organisasi perawatan kesehatan

Untuk memberikan perawatan optimal bagi pasien dan bimbingan yang membantu pasien membuat keputusan perawatan kesehatan yang baik, petugas kesehatan membutuhkan data klinis berkualitas tinggi tentang pasien mereka. Menyampaikan data yang tepat, dalam format yang tepat, kepada orang yang tepat pada waktu yang tepat, merupakan tantangan bagi TI kesehatan, terutama mengingat persyaratan etika dan peraturan untuk penanganan data kesehatan. Selain itu, inovasi medis terus meningkatkan jumlah dan kompleksitas data perawatan kesehatan. Menurut [RBC Capital Markets](#), 30 persen data dunia dihasilkan oleh layanan kesehatan pada tahun 2018. Pada tahun 2025, data perawatan kesehatan akan tumbuh setiap tahun sebesar 36 persen. Strategi pemrosesan data kesehatan tradisional berjuang untuk mendukung peningkatan volume dan kompleksitas data yang cepat ini.

Banyak organisasi kesehatan meningkatkan hasil pasien dengan menggunakan analisis kesehatan populasi. Organizations juga menggunakan [obat presisi](#), yang didefinisikan sebagai “pendekatan inovatif yang mempertimbangkan perbedaan individu dalam gen, lingkungan, dan gaya hidup pasien.” Pengobatan presisi meningkatkan efektivitas perawatan kesehatan, tetapi juga menciptakan tantangan pemrosesan data baru bagi organisasi perawatan kesehatan. Pendekatan kedokteran presisi standar juga sulit untuk diskalakan di luar paradigma one-patient-at-a waktu. Organisasi kesehatan harus mengurangi waktu dari memperoleh data mentah hingga memberikan informasi yang dapat digunakan kepada pekerja garis depan. Informasi itu harus akurat, dan harus disajikan dalam bentuk yang dapat diakses, dipahami, dan diterapkan oleh dokter dengan mudah.

Data kesehatan tidak tergantikan dan merupakan aset yang sangat berharga dari banyak organisasi perawatan kesehatan. Oleh karena itu, Anda harus memperlakukan data perawatan kesehatan sebagai aset. Organisasi perawatan kesehatan Anda harus mendapatkan kepercayaan pasien dan mengelola risiko reputasi dengan mengumpulkan dan menghormati persetujuan pasien dan melindungi data dari akses dan penggunaan yang tidak tepat. Organisasi perawatan kesehatan Anda harus secara bersamaan melindungi privasi pasien, mematuhi batasan peraturan yang ketat dan beragam, dan memberikan data berkualitas tinggi dengan cepat kepada petugas kesehatan, kolaborator, dan pasien. Anda juga harus memutuskan apakah Anda dapat memonetisasi data perawatan kesehatan dengan aman dengan cara yang konsisten dengan misi Anda, kebijakan keamanan dan privasi data Anda, dan persetujuan pasien. Tantangan meliputi:

- Pipa data perawatan kesehatan tradisional kewalahan karena tidak dibangun untuk menangani persyaratan yang semakin ketat dan menantang ini.

- Sistem tradisional biasanya dibungkam. Untuk memberikan pandangan yang komprehensif tentang data yang relevan dan pasien individu, sistem modern harus terintegrasi dan interoperable.
- Sistem tradisional sering diatur di sekitar modalitas data tunggal. Sistem modern harus secara inheren multimodal.
- Sistem tradisional tidak dirancang untuk menangani data pada skala dan kecepatan yang dibutuhkan sistem modern.
- Sistem tradisional biasanya dirancang untuk berjalan di tempat dan dioptimalkan untuk sumber daya TI yang tersedia. Sistem modern harus dapat memanfaatkan penyimpanan data dan memproses sumber daya di lingkungan hybrid on-premis-cloud dan terkadang lingkungan multicloud.

Organisasi kesehatan yang mengadopsi dan menjalankan strategi data kesehatan modern memposisikan diri mereka untuk maju seiring dengan percepatan inovasi dalam perawatan kesehatan dan ilmu kehidupan.

# Manfaat mengadopsi strategi data kesehatan modern

Strategi data perawatan kesehatan modern membantu organisasi Anda membuat arsitektur data yang mengubah data mentah menjadi informasi lengkap yang dapat digunakan dengan kecepatan dan skala. Ini mendukung pengumpulan dan penggunaan data organisasi Anda dari sumber yang berbeda dan dalam berbagai bentuk, termasuk:

- Siklus pendapatan layanan kesehatan — data manajemen, termasuk klaim, pengiriman uang, dan manfaat
- Data klinis multimodal, termasuk data catatan kesehatan elektronik (EHR) terstruktur dan tidak terstruktur, hasil laboratorium, data genom, dan data pencitraan medis
- Data farmasi, seperti data pengisian resep
- Data kesehatan eksternal dari biobank, data commons, kumpulan data penelitian, dan sumber lainnya
- Data pasien, termasuk data perilaku (dari perangkat yang dapat dikenakan atau IoT) dan data perangkat rumah

Organisasi kesehatan harus membangun jaringan data untuk menyerap, menyelaraskan, membersihkan, dan menganalisis data ini. Data kemudian harus disampaikan tepat waktu sebagai informasi yang dapat digunakan untuk pekerja garis depan pada titik perawatan. Setiap langkah pipa data harus [dirancang dengan baik](#): aman dan patuh, andal, berkinerja, elastis, dan berkelanjutan.

Organisasi kesehatan menggunakan data dan layanan berorientasi data untuk mempercepat penelitian dan pengembangan. Mereka juga membangun algoritma prediktif yang dapat membantu dokter dalam mengidentifikasi masalah sebelum terjadi. Untuk mencapai tujuan ini, organisasi perawatan kesehatan menerapkan teknologi analitik canggih, kecerdasan buatan (AI), dan pembelajaran mesin (ML), termasuk kemajuan terbaru dalam AI generatif.

Sebagaimana dijelaskan di bagian berikut, Amazon Web Services (AWS) dan AWS Partner Network memberikan Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan (HIPAA) - layanan yang memenuhi syarat, aman, andal, berkinerja, elastis, dan elastis untuk setiap tahap pipa data perawatan kesehatan. Panduan ini mencakup praktik terbaik untuk membantu organisasi perawatan kesehatan Anda memenuhi tujuan sistem Anda dan sasaran pasien organisasi Anda.

Dokumen strategi ini memberikan contoh bagaimana AWS layanan dapat mendukung pembangun di Industri Kesehatan dan Ilmu Hayati. Contoh-contoh ini tidak lengkap, dan tidak termasuk AWS

Partner solusi yang dapat membantu Anda membangun dan mengelola solusi dengan lebih cepat dan hemat biaya. Untuk daftar solusi Kesehatan dan Ilmu Hayati dari AWS Partner Network, kunjungi [AWS Marketplace](#).

# Komponen strategi data kesehatan modern

Untuk menjalankan strategi data perawatan kesehatan modern, adopsi metodologi tangkas, dengan fokus pada penyampaian kasus penggunaan yang terkait langsung dengan strategi bisnis. Dengan mengadopsi pendekatan data yang gesit, organisasi Anda dapat dengan cepat mencapai tujuan bisnisnya. Metodologi tangkas untuk data meliputi:

- **Perspektif** - Fokus pada merancang dan membuat penawaran yang stabil dan berkemampuan data. Mengembangkan persyaratan bisnis yang mendukung pekerja garis depan, meminimalkan beban entri data, dan meningkatkan pengalaman pasien. Ciptakan lingkungan yang aman untuk menguji ide, bereksperimen, dan menangkap pelajaran yang dipetik. Gunakan pelajaran ini untuk mendorong iterasi masa depan. Perlakukan data sebagai aset organisasi yang kritis, dan berikan tingkat kepentingan yang sama yang terkait dengan aset penting lainnya.
- **Kepemilikan** — Berbagi kepemilikan atas masalah dan hasil antara pemimpin bisnis dan teknologi. Mereka harus menentukan tujuan bisnis strategis untuk organisasi, termasuk hasil pasien, efisiensi biaya, dan kepatuhan terhadap peraturan. Misalnya, Anda dapat mendirikan Cloud Center of Excellence ([CCoE](#)) dengan keterlibatan kepemimpinan bisnis dan TI. CCoE membantu menciptakan tanggung jawab bersama untuk mempercepat adopsi dan nilai bisnis. Pada saat yang sama, CCoE merangkul potensi inovasi cloud dan membantu memastikan solusi data yang dirancang dengan baik.
- **Literasi data** — Mempromosikan literasi data dengan membentuk komite data yang mencakup representasi klinis dan operasional. Para pemimpin komite harus berkomitmen untuk mempromosikan kelincahan, inovasi, dan pola pikir berorientasi data di seluruh organisasi dan dalam unit bisnis masing-masing. Buat peta jalan yang menyelaraskan literasi data dan transformasi bisnis berbasis data. Melatih dan mendorong line-of-business para pemimpin untuk menggunakan sistem pendukung keputusan dan membuat keputusan berbasis data.
- **Tata Kelola** — Menetapkan kerangka kerja tata kelola data yang menguraikan kebijakan, prosedur, dan standar untuk mengelola data dalam organisasi Anda. Mengembangkan pedoman untuk kualitas data, privasi data, keamanan data, dan akses data. Rancang pedoman ini untuk memfasilitasi kepatuhan terhadap peraturan. Terapkan kerangka tata kelola secara bertahap saat Anda menerapkan kasus penggunaan bisnis. Buat model tata kelola federasi atau terdistribusi untuk menyeimbangkan masalah keamanan, privasi, dan peraturan yang tidak dapat dinegosiasikan dengan kebutuhan untuk berinovasi. Identifikasi peluang manajemen data pusat (misalnya, indeks pasien pusat, katalog data terpadu). Menilai dampak potensial pada perusahaan dalam menyatukan data multimodal.

Secara bersamaan, tata kelola harus memfasilitasi demokratisasi data untuk akses data yang cepat dan intuitif bagi mereka yang membutuhkannya, membantu pengguna merasa diberdayakan, tidak terkontrol. Untuk memenuhi persyaratan tata kelola secara lebih efisien dan dengan lebih sedikit beban pada staf garis depan, gunakan alat kepatuhan AWS [perawatan kesehatan](#) yang dibuat khusus dan praktik terbaik. Sedapat mungkin, sediakan alat swalayan untuk mengurangi dampak pada data dan tim analis.

- **Artefak** — Tentukan dan gunakan artefak yang meningkatkan kolaborasi dan berbagi data di berbagai tim dan departemen. Artefak utama termasuk katalog data, kamus data, dan model data. Misalnya, gunakan [AWS Glue Data Catalog](#) untuk membuat katalog data. Gunakan [Amazon DataZone](#) dan [AWS Clean Rooms](#) untuk berbagi data atau wawasan data tertentu di dalam dan di seluruh organisasi perawatan kesehatan tanpa mengorbankan privasi pasien atau melanggar persyaratan kepatuhan HIPAA.
- **Arsitektur data** — Desain dan terus menyempurnakan arsitektur data Anda. Arsitektur yang mendukung strategi data kesehatan modern harus mencakup aset data multimodal. Mengadopsi pendekatan berbasis domain untuk menangani data multimodal dengan memisahkan produsen data dari konsumen dalam arsitektur. Pertimbangkan penyimpanan, retensi, dan format. Tempatkan penekanan pada kemudahan akses dan penggunaan, difasilitasi oleh manajemen metadata yang kuat.

Kebutuhan khusus perawatan kesehatan, seperti kepatuhan terhadap peraturan dan manajemen persetujuan, harus membantu menentukan kebijakan dan prosedur penanganan data.

Pertimbangkan untuk mendefinisikan standar data pusat yang diperlukan untuk mendefinisikan entitas bisnis secara unik seperti pasien, penyedia, dan karyawan. Kurangi kompleksitas proses dengan mendefinisikan dan membuat kumpulan data yang tidak teridentifikasi untuk membantu mempercepat kasus penggunaan yang tidak memerlukan akses ke Informasi Kesehatan yang Dilindungi (PHI).

- **Teknologi** — Mengadopsi arsitektur berbasis cloud yang menggunakan layanan yang dibangun khusus berdasarkan kebutuhan bisnis yang ada. Buat solusi di mana organisasi Anda perlu berinovasi, tetapi gunakan off-the-shelf solusi dan layanan terkelola bila memungkinkan untuk mengurangi menjaga tim Anda tetap fokus pada inovasi. Misalnya, gunakan [analitik prediktif](#) untuk mengidentifikasi pasien yang rentan atau berisiko untuk penjangkauan dan perawatan proaktif. Gunakan [Amazon Comprehend Medical](#) untuk menanyakan dan mengekstrak informasi dari data tidak terstruktur dan semi-terstruktur seperti catatan medis. Gunakan [AWS HealthImaging](#) untuk membantu pekerja garis depan memproses gambar medis dengan lebih akurat dan efisien.

- Akses data yang demokratisasi — [Mempromosikan transparansi dan visibilitas ke data organisasi dengan menggunakan alat katalogisasi seperti Amazon. DataZone](#) Alat-alat ini menyediakan kemampuan untuk mencari dan mengeksplorasi data organisasi yang tersedia, untuk memahami definisi data, siklus hidup, dan garis keturunan, dan untuk meminta akses ke data.
- Kemudahan penggunaan — Keberhasilan strategi data kesehatan modern Anda tergantung pada kemudahan penggunaan. Menilai berbagai tingkat literasi data dalam organisasi, dan mengembangkan rencana untuk mengatasi konsumsi di seluruh spektrum pengguna. Menilai tingkat literasi data saat ini di seluruh organisasi, menyusun kurikulum literasi data, dan mengidentifikasi peluang proyek untuk mengembangkan staf dan rencana pelatihan. Pertimbangkan tiga kategori pengguna luas berikut yang mungkin termasuk dalam staf Anda, dengan fokus pada kebutuhan mereka untuk pelatihan dan adopsi:
  - Data wranglers — Pengguna ini paham data, dan mereka memiliki keahlian teknologi untuk menjelajahi kumpulan data semikurasi dan tidak terkurasi. Untuk meningkatkan produktivitas, penting untuk melengkapi pengguna ini dengan set alat yang mereka butuhkan. AWS Layanan seperti [Amazon Athena](#), [Amazon Redshift Spectrum](#) [AWS Glue](#) [DataBrew](#), dan [SageMaker Amazon AI Data Wrangler](#) membantu pengguna ini untuk terhubung dan mengintegrasikan kumpulan data yang berbeda tanpa harus menulis kode rekayasa data yang kompleks.
  - Pengguna daya — Pengguna ini biasanya ahli materi pelajaran bisnis (). SMEs Mereka cerdas data, tetapi mereka memiliki keterampilan teknis yang terbatas. Mereka mengandalkan kumpulan data yang dikuratori untuk membuka nilai dalam data. Pengguna ini mendapat manfaat dari alat grafis untuk melakukan operasi modifikasi data ringan dan membuat visual yang menarik. Layanan AWS seperti [Amazon Quick](#) membantu pengguna ini menjelajahi, mengedit, membersihkan, menyelaraskan, memvisualisasikan, dan berbagi data.
  - Konsumen — Ini adalah eksekutif dan line-of-business pemimpin nonteknis. Pengguna ini biasanya lebih suka menggunakan laporan bawaan dan dasbor interaktif. Memberi pengguna ini cara untuk melakukan eksplorasi data yang dipandu dapat mempercepat inovasi dan keputusan bisnis yang penting. Alat intelijen bisnis generatif (BI) seperti [Amazon Quick Q](#), yang memungkinkan interaksi bahasa alami untuk memperoleh wawasan berbasis data, dapat membantu kategori pengguna ini.

Secara keseluruhan, strategi data kesehatan modern harus berakar pada kasus penggunaan dan tindakan yang terkait langsung dengan strategi bisnis. Ini juga harus mempertimbangkan pola pikir, kepemilikan, artefak, tata kelola, dan teknologi sebagai komponen yang sama pentingnya. Dengan demikian, organisasi perawatan kesehatan Anda dapat menjadi berbasis data, gesit, dan dapat berputar dengan cepat dalam menanggapi kondisi di luar kendali organisasi Anda.

# Menerapkan strategi data kesehatan modern

Untuk menerapkan strategi data perawatan kesehatan modern Anda, kami sarankan mengikuti prinsip-prinsip ini:

- Buat model operasi untuk organisasi berbasis data — Identifikasi peran, kompetensi, dan model operasi target yang diperlukan untuk membuat organisasi berbasis data. Menumbuhkan literasi data dalam bisnis, TI, dan siapa saja yang terlibat dalam perawatan pasien, termasuk pasien. Merangkul potensi inovatif cloud untuk mempercepat pengiriman nilai bisnis. Mulailah dengan strategi data hybrid sehingga organisasi Anda dapat bergerak dengan cepat. Manfaatkan alat dan teknologi lokal yang ada dengan solusi berbasis cloud untuk menciptakan produk data yang gesit dan efisien. AWS menawarkan serangkaian produk untuk mengadopsi [model cloud hybrid](#) untuk membantu mempercepat transisi Anda ke cloud.
- Bekerja mundur dari kebutuhan garis depan — Untuk setiap peran organisasi, identifikasi data apa yang dibutuhkan, kapan, dan dalam format apa. Selanjutnya, tentukan asal data dan cara mengirimkannya tepat waktu. Menyampaikan data dalam format yang dapat dengan mudah dipahami dan diterapkan oleh pengguna. Misalnya, gunakan [AWS HealthLake](#) dan [Amazon QuickSight](#) untuk membuat dasbor yang menyertakan visualisasi data yang dapat dimengerti. Jika memungkinkan, buat solusi swalayan yang dapat diakses dan dimanipulasi oleh pengguna akhir tanpa perlu intervensi analis atau ilmuwan data.
- Otomatiskan pipa data — Jika petugas kesehatan garis depan harus mentransfer data secara manual dari satu sistem ke sistem lainnya, langkah itu menunda pengiriman data. Ini memperkenalkan kesenjangan dan kesalahan data, mengalihkan perhatian staf garis depan dari perawatan pasien, mengikis moral staf, dan mengurangi produktivitas staf. Otomatisasi mungkin tampak mahal, tetapi pertimbangkan total biaya pemrosesan data manual dalam perhitungan return-on-investment (ROI) Anda. Jika sumber data memerlukan transfer data manual, pertimbangkan apakah Anda dapat menyimpan data di tempatnya. Untuk memperoleh data dari perangkat medis, Anda dapat menggunakan [AWS integrasi dengan perangkat medis](#), dan gunakan [AWS Glue](#) untuk membangun pipa data yang efisien secara operasional.
- Pindah dari monolit ke modular — Sistem monolitik memiliki saling ketergantungan yang mencegah inovasi dalam komponen apa pun dan yang mempersulit pemecahan masalah ketika ada yang salah. Strategi data kesehatan modern harus modular: terdiri dari komponen independen dengan antarmuka yang terdefinisi dengan baik sehingga Anda dapat berinovasi di setiap modul tanpa mengganggu modul lain. Gunakan penyimpanan data yang mendukung standar interoperabilitas. Misalnya, pertimbangkan untuk menggunakan [HealthLake](#), penyimpanan data yang kompatibel dengan Sumber Daya Interoperabilitas Perawatan Kesehatan Cepat (FHIR) HIPAA, bersama

dengan perangkat lunak off-the-shelf konsumsi data, dan gunakan [AWS HealthOmics](#) untuk mengubah data genomik, transkriptomik, dan omics lainnya.

- Gunakan layanan terkelola dan tanpa server — Kurangi peningkatan berat konfigurasi server dan sistem operasi, manajemen tambalan, dan pemantauan dengan menggunakan layanan terkelola, tempat penyedia layanan cloud mengelola infrastruktur yang mendasarinya untuk Anda. Pindahkan sumber daya staf TI Anda dari manajemen sistem (menjaga lampu menyala) ke inovasi data. Misalnya, gunakan [AWS Lambda](#) atau [AWS Fargate](#) untuk layanan komputasi, [Amazon Aurora](#) Tanpa Server untuk database relasional, dan [Amazon Redshift](#) Tanpa Server untuk gudang data Anda.
- Menyederhanakan dan mempersingkat jalur data — Memindahkan dan mengubah data berpotensi mahal dan memakan waktu. Itu juga dapat memperkenalkan kesalahan ke dalam solusi data. Untuk mengoptimalkan biaya, mempercepat pengiriman data, dan meningkatkan kualitas data, lakukan hal berikut:
  - Gunakan data di tempat tinggalnya.
  - Minimalkan operasi ekstrak, transformasi, dan beban (ETL).
  - Gunakan akses data federasi.

Misalnya, gunakan layanan AWS terkelola untuk mengimplementasikan [arsitektur data mesh](#), meminimalkan overhead yang terlibat dalam pergerakan data, dan menggunakan kueri [federasi](#).

Untuk informasi dan detail tambahan tentang penerapan arsitektur untuk mendukung strategi data kesehatan modern, lihat [Lampiran D: Panduan tambahan untuk menerapkan strategi data kesehatan modern](#).

# Contoh implementasi strategi data kesehatan modern

AWS menyediakan arsitektur referensi yang dapat digunakan organisasi perawatan kesehatan untuk memahami dan membangun platform data yang mendukung pendekatan data yang gesit. Arsitektur referensi berikut menggambarkan arsitektur [data mesh](#) untuk perawatan kesehatan. Dalam arsitektur ini, tanggung jawab manajemen data diatur di sekitar fungsi bisnis atau domain teknis. Pengguna dapat mencari, berbagi, dan menemukan data dalam skala besar melintasi batas-batas organisasi. Tim domain bertanggung jawab untuk mengumpulkan, mengubah, dan menyediakan data yang terkait dengan atau dibuat oleh fungsi bisnis mereka.

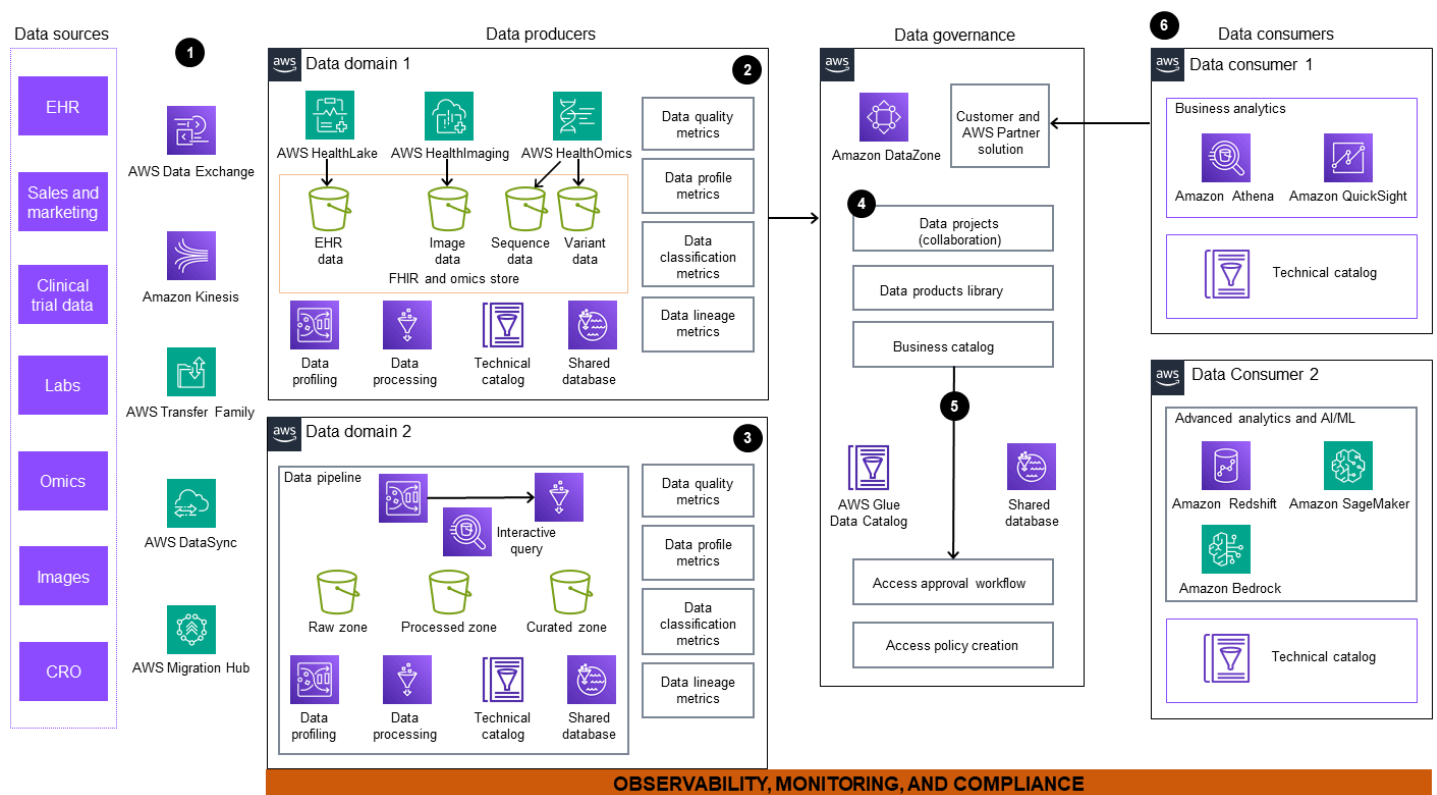


Diagram arsitektur mencakup komponen-komponen berikut:

1. Data dicerna dari sumber data eksternal dan internal. Sumber-sumber ini termasuk, tetapi tidak terbatas pada, sistem Catatan Kesehatan Elektronik (EHR), laboratorium, fasilitas pengurutan, dan pusat pencitraan. AWS menawarkan serangkaian layanan seperti [AWS Data Exchange](#), [Amazon Kinesis](#), [AWS Transfer Family](#), [AWS DataSync](#), [AWS Migration Hub](#), [AWS HealthLake](#), dan [AWS Glue](#) (ETL). Anda dapat menggunakan layanan ini untuk membantu memigrasikan kumpulan data internal Anda dan berlangganan kumpulan data internal dan eksternal.

2. Domain data 1 terdiri dari alur kerja yang komprehensif untuk memproses data berorientasi pasien multimodal, termasuk data klinis, omics, dan pencitraan. Data klinis EHR dicerna dan disimpan dalam penyimpanan HealthLake data, layanan terkelola yang dibuat khusus untuk data klinis. [AWS HealthOmics](#), layanan yang dibuat khusus untuk data omics, menangani urutan dan varian penyimpanan dan alur kerja. Data pencitraan dicerna dan disimpan di [AWS HealthImaging](#). Data ini kemudian diubah menjadi produk siap konsumsi dan dipublikasikan di pasar data perusahaan untuk aksesibilitas dan penggunaan yang luas.
3. Di domain data 2, Amazon Kinesis AWS Glue, dan menyerap data AWS Data Exchange mentah ke dalam pipeline data. Sumber data dapat mencakup pendaftar publik, pemantauan pasien jarak jauh, dan program Enterprise Resource Planning (ERP). Pipeline memuat data mentah ke dalam bucket [Amazon Simple Storage Service \(Amazon S3\)](#). Data ini dibersihkan, dikuratori, diubah, dan disimpan untuk dipublikasikan sebagai produk data. [Amazon Athena](#) menawarkan mesin kueri interaktif yang dapat digunakan produsen data untuk mengubah data menggunakan SQL. [AWS Glue DataBrew](#) menyediakan transformasi data visual, normalisasi, dan kemampuan profil.
4. [Amazon DataZone](#) menangani penerbitan metadata, proyek data kolaboratif, dan pustaka produk data ke katalog bisnis pusat.
5. Portal analitik data terpadu memungkinkan kolaborasi seputar data dengan memberikan tampilan produk data melalui tata kelola federasi. Amazon DataZone memungkinkan alur kerja layanan mandiri dengan AWS Glue Data Catalog didukung oleh AWS Lake Formation, sehingga pengguna dapat berbagi, mencari, menemukan data, dan meminta izin untuk konsumsi.
6. [Konsumen data dapat mengakses data, membuat tampilan hilir, dan menggunakan alat yang dibuat khusus seperti Amazon Athena, Amazon Quick, Amazon Redshift, Amazon AI SageMaker ,dan Amazon Bedrock untuk melakukan hal berikut:](#)
  - Analisis operasional
  - Informatika klinis
  - Penelitian
  - Keterlibatan pasien dan klinis

Konsumen data juga dapat mengembangkan aplikasi inovatif dengan menggunakan AI generatif, dan mereka dapat mempublikasikan produk data ke katalog bisnis.

Untuk informasi selengkapnya tentang arsitektur data mesh, lihat [Apa itu Data Mesh?](#)

# AI Generatif

Organisasi perawatan kesehatan menggunakan AI generatif untuk berbagai aplikasi, mulai dari mengotomatiskan interpretasi citra medis hingga menghasilkan rekomendasi diagnostik dan rencana perawatan berdasarkan gambar dan data tekstual. Adopsi AI generatif mempercepat inovasi dan meningkatkan efisiensi di seluruh rangkaian perawatan. Fokus baru pada AI generatif telah memaksa perawatan kesehatan untuk memperluas fokus datanya untuk memasukkan lebih banyak bentuk data tidak terstruktur, memperluas jumlah dan berbagai kasus penggunaan yang dapat diterima oleh AI. Secara umum, ada empat pola yang dapat dipilih organisasi, tergantung pada kasus penggunaannya, untuk menerapkan solusi AI generatif:

- **Rekayasa cepat** — Dalam rekayasa yang cepat, pengguna menyediakan data yang relevan sebagai konteks, memandu model AI generatif untuk membuat konten yang mereka inginkan. Organizations dengan strategi data kesehatan modern dapat memastikan bahwa data yang relevan mudah ditemukan, dibagikan, dan dikonsumsi.
- **Retrieval Augmented Generation (RAG)** — Pola RAG dibangun di atas rekayasa yang cepat. Alih-alih pengguna memberikan data yang relevan, program mencegat pertanyaan atau masukan pengguna. Program mencari di seluruh repositori data untuk mengambil konten yang relevan dengan pertanyaan atau masukan. Program ini memasukkan data yang ditemukannya ke model AI generatif untuk menghasilkan konten. Strategi data perawatan kesehatan modern memungkinkan kurasi dan pengindeksan data perusahaan. Data kemudian dapat dicari dan digunakan sebagai konteks untuk petunjuk atau pertanyaan, membantu model bahasa besar (LLM) dalam menghasilkan tanggapan.

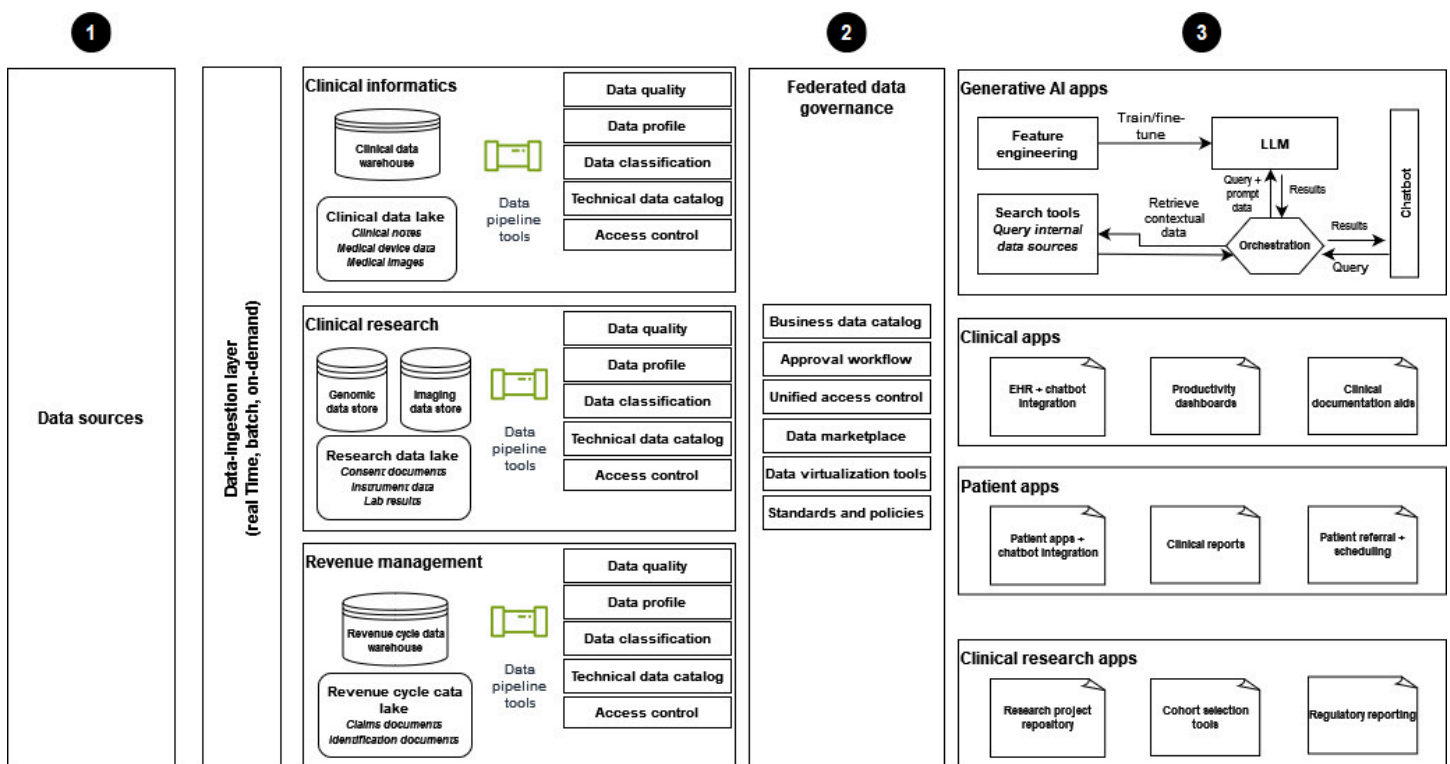
Organisasi Anda dapat menggunakan dua pola berikut untuk memfokuskan output model AI generatif pada menghasilkan konten yang sesuai dengan konteks data mereka.

- **Fine-tuning** — Dengan menggunakan pola ini, organisasi Anda dapat melangkah lebih jauh dengan menyesuaikan model AI generatif. Ini melibatkan fine-tuning model pada sampel kecil data khusus untuk organisasi. Karena ukuran sampel kecil, pola ini memberikan keseimbangan biaya dan kustomisasi. Untuk menghindari bias dalam keluaran model, gunakan kumpulan data sampel kecil yang beragam dan representatif dari pola data organisasi Anda. Strategi data kesehatan modern mendukung akses yang efisien ke berbagai data untuk menyiapkan kumpulan data sampel.
- **Bangun model Anda sendiri** — Jika organisasi Anda perlu menghasilkan konten di seluruh volume data yang sangat terspesialisasi dan besar, dan tiga pola sebelumnya tidak memadai, Anda dapat membuat model Anda sendiri.

Strategi data modern memainkan peran penting dalam solusi AI generatif dengan membantu memastikan bahwa data memiliki karakteristik sebagai berikut:

- Data berkualitas tinggi untuk mendukung akurasi
- Data real-time atau mendekati real-time untuk membantu memastikan bahwa output model relevan
- Beberapa modalitas data di berbagai sumber data untuk menyediakan model dengan akses ke kumpulan data yang diperkaya untuk menghasilkan konten

Diagram berikut menunjukkan implementasi strategi data kesehatan modern yang menggunakan arsitektur data mesh untuk mendukung solusi AI generatif.



1. Data dicerna dari beragam sumber data di domain Informatika Klinis, Penelitian Klinis, dan Manajemen Pendapatan, dan data tersedia untuk organisasi perawatan kesehatan.
2. Tata kelola data federasi membantu memastikan kontrol akses yang ketat untuk berbagi data dan akses terpadu.
3. Konsumen data meliputi:
  - Aplikasi AI generatif, terutama yang menggunakan data untuk melatih dan LLMs menyempurnakan. Aplikasi ini menggunakan data perusahaan untuk chatbot Tanya Jawab untuk meningkatkan efisiensi operasional dan pengalaman pasien dan penyedia.

- Aplikasi klinis yang dilengkapi dengan alat-alat seperti chatbot terintegrasi EHR, dasbor produktivitas, dan alat bantu dokumentasi.
- Aplikasi yang berpusat pada pasien untuk meningkatkan pengalaman pasien. Aplikasi ini menampilkan interaksi chatbot, laporan klinis, dan proses rujukan dan penjadwalan yang efisien.
- Penelitian klinis, dengan repositori proyek penelitian dan aplikasi yang dirancang untuk analisis kohort dan pelaporan peraturan.

Dengan arsitektur ini, pemangku kepentingan di organisasi Anda dapat fokus pada kurasi dan pengelolaan data yang mereka kumpulkan dari sumber lain sambil membuat data mereka sendiri dapat diakses oleh seluruh organisasi. Mereka dapat menggunakan alat yang tersedia di lapisan tata kelola data federasi untuk menentukan metadata, mengelola alur kerja persetujuan akses, dan menentukan serta menegakkan kebijakan. Selain itu, lapisan tata kelola data federasi menyediakan kontrol akses terpusat. Ini menciptakan lingkungan untuk kurasi berbagai sumber data dan untuk menyegarkan aset data berkualitas tinggi pada frekuensi tertentu untuk mempertahankan relevansi. AWS menawarkan serangkaian kemampuan yang komprehensif untuk memenuhi kebutuhan AI generatif Anda. [Amazon Bedrock](#) adalah cara entry-level bagi organisasi Anda untuk membangun dan menskalakan aplikasi berbasis AI generatif. [AWS Trainium](#) dan [AWS Inferentia](#) chip menawarkan biaya terendah untuk model pelatihan dan menjalankan inferensi di cloud. Untuk informasi lebih lanjut, lihat [Generative AI di AWS](#).

# Memenuhi tujuan pemangku kepentingan untuk strategi data kesehatan modern

Organisasi layanan kesehatan berusaha untuk meningkatkan pengalaman dan hasil pasien secara adil, meminimalkan biaya operasi dan modal, mematuhi hukum dan peraturan, dan menghormati hak-hak pasien. Untuk panduan terperinci tentang bagaimana strategi data perawatan kesehatan modern dapat membantu organisasi perawatan kesehatan Anda memenuhi tujuan ini, lihat [Lampiran A. Memenuhi tujuan perawatan kesehatan](#).

Pasien dan pengasuh mereka memiliki beragam tujuan dan harapan dalam hal perawatan kesehatan. Mereka ingin menerima perawatan yang aman dan efektif, dan membuat keputusan berdasarkan informasi tentang perawatan kesehatan mereka. Mereka juga ingin mengontrol siapa yang memiliki akses ke data perawatan kesehatan mereka dan bagaimana data itu digunakan. Untuk informasi lebih lanjut tentang tujuan pasien, lihat [Lampiran B. Memenuhi tujuan pasien](#).

Organisasi kesehatan perlu meningkatkan kelincahan dan kemampuan mereka untuk berinovasi dengan mengadopsi sistem teknis yang fleksibel dan mudah beradaptasi dengan perubahan kondisi. Untuk informasi lebih lanjut tentang tujuan sistem perawatan kesehatan, lihat [Lampiran C. Memenuhi tujuan TI sistem kesehatan](#).

Arsitek sistem kesehatan dapat mengikuti AWS panduan dan arsitektur referensi. Untuk arsitektur tingkat tinggi yang memenuhi kebutuhan kesehatan umum, lihat [Lampiran D. Panduan tambahan tentang penerapan strategi data kesehatan modern](#).

# Kesimpulan

AWS membantu organisasi perawatan kesehatan berubah menjadi organisasi perawatan kesehatan berbasis data. Dalam dokumen ini, kami membahas mengapa inovasi dalam perawatan kesehatan dan ilmu kehidupan melampaui sistem pemrosesan data tradisional. Kami menggambarkan bagaimana strategi data kesehatan modern yang terdiri dari strategi budaya, organisasi, dan arsitektur membantu organisasi perawatan kesehatan merangkul dan menerapkan inovasi ini. Akibatnya, organisasi perawatan kesehatan dapat meningkatkan pengalaman dan hasil pasien, menjaga kepatuhan dan postur keamanan, mengoptimalkan biaya, dan meningkatkan produktivitas dan moral bagi staf perawatan kesehatan.

Ebook [The Data Driven Enterprise](#) menjelaskan apa yang diperlukan untuk menjadi berbasis data, dan mengapa itu penting dalam lingkungan digital saat ini.

Untuk bimbingan teknis dan arsitektur, [situs AWS for Healthcare & Life Sciences](#) telah mengatur sumber daya ini untuk membantu Anda menemukan tempat yang tepat untuk memulai. Situs ini mencakup [studi kasus](#) untuk eksplorasi lebih lanjut. Ini juga mencakup [Mitra Kompetensi AWS Kesehatan](#) untuk menemukan dukungan pihak ketiga untuk perjalanan data cloud Anda. Akhirnya, ini mencakup tautan ke solusi dan teknologi yang dapat membantu Anda menerapkan komponen kunci dari arsitektur data kesehatan.

Untuk mempelajari lebih lanjut tentang bagaimana AWS dapat membantu Anda menerapkan strategi data perawatan kesehatan modern, [terhubung dengan perwakilan AWS penjualan yang berspesialisasi dalam industri perawatan kesehatan](#).

# Sumber daya

Halaman-halaman berikut dapat membantu memandu Anda melalui proses penerapan strategi data perawatan kesehatan modern untuk organisasi Anda:

- [AWS untuk Kesehatan & Ilmu Hayati](#)
- [Arsitektur untuk Keamanan dan Kepatuhan HIPAA di Amazon Web Services](#) (whitepaper)
- [Arsitektur Data Modern pada AWS](#)
- [Alasan Arsitektur Data Modern tentang AWS](#)

## AWS Perpustakaan Solusi

Perpustakaan AWS Solusi menawarkan solusi yang diperiksa dan dikuratori oleh AWS para ahli. Perpustakaan Solusi mencakup tautan ke AWS layanan, solusi yang dikembangkan oleh anggota AWS Partner Network, dan solusi panduan yang memberikan saran teknis dan arsitektur. Solusi ini sangat membantu untuk memberikan panduan kepada tim teknis yang mereka butuhkan untuk membangun alur kerja berbasis cloud baru atau memperluas alur kerja yang sudah ada. Kategori solusi berikut relevan dengan industri perawatan kesehatan:

- [Bagian Kesehatan, Ilmu Hayati, dan Genomik](#)
- [Bagian Penelitian Nirlaba](#)

## AWS Marketplace

AWS Marketplace Dapat membantu memulai atau mempercepat inovasi. Ini menampilkan solusi berbasis cloud yang dibangun oleh Mitra pihak ketiga AWS . Solusi ini dapat membantu organisasi Anda menurunkan biaya TI, mengelola risiko, dan meningkatkan efisiensi. AWS Marketplace Kategori berikut relevan untuk pelanggan layanan kesehatan:

- [Bagian perawatan kesehatan](#)
- [Bagian nirlaba](#)

## Lampiran A. Memenuhi tujuan organisasi kesehatan

Sederhanakan akses data, kurangi overhead administratif, minimalkan entri data pasien, dan berikan informasi yang dipersonalisasi.

### Tingkatkan pengalaman pasien

Pengalaman pasien mencakup berbagai interaksi yang dimiliki pasien dengan sistem perawatan kesehatan. Strategi data perawatan kesehatan modern dapat meningkatkan pengalaman pasien dengan:

- Menyederhanakan akses data untuk pasien dan dokter
- Mengurangi overhead administratif
- Meminimalkan persyaratan entri data pasien
- Memberikan informasi yang dipersonalisasi tentang kondisi, perawatan, risiko, manajemen penyakit, uji klinis, dan terapi yang muncul

Organisasi Anda dapat menggunakan pintu depan digital atau layanan portal pasien yang diaktifkan oleh strategi data perawatan kesehatan modern. Layanan ini, yang ditawarkan oleh AWS Mitra, memandu setiap pasien dari penemuan layanan kesehatan melalui pemulangan dan tindak lanjut. Kemampuan pintu depan digital utama mencakup opsi penjadwalan online, survei kesehatan online, dan akses pasien ke data kesehatan multimodal terintegrasi. Data itu mencakup pencitraan dan data genom di beberapa penyedia layanan kesehatan dan laboratorium. [Strategi data perawatan kesehatan modern mendukung modernisasi call center, termasuk chatbots untuk memberikan informasi dasar 24/7/365, didukung oleh pusat kontak multibahasa omnichannel menggunakan Amazon Connect.](#)

### Meningkatkan hasil di seluruh populasi

Kesehatan populasi berfokus pada kondisi dan faktor yang saling terkait yang mempengaruhi kesehatan populasi. Ini juga mengidentifikasi variasi sistemik dalam pola yang terkait dengan faktor-faktor ini. Akhirnya, ini menerapkan pengetahuan yang dihasilkan untuk mengembangkan dan menerapkan kebijakan dan praktik untuk meningkatkan kesehatan dan kesejahteraan populasi tersebut. Sistem Kesehatan dapat mencapai hasil kesehatan yang lebih baik dengan mengurangi

biaya dengan menjembatani kesenjangan antara kesehatan populasi dan pemberian layanan kesehatan.

Strategi data perawatan kesehatan modern dapat membantu meningkatkan hasil kesehatan penduduk dengan:

- Mensegmentasi populasi pasien berdasarkan atributnya
- Mengidentifikasi faktor risiko di seluruh komunitas
- Menggunakan model pengiriman rumah perawatan medis primer
- Menggunakan skrining dan pencegahan berbasis bukti pada populasi yang ditugaskan
- Berfokus pada kesehatan secara keseluruhan
- Pindah dari perawatan berbasis volume ke perawatan berbasis nilai

Untuk mengembangkan sistem data perawatan kesehatan yang meningkatkan kesehatan penduduk, organisasi kesehatan harus dapat mengintegrasikan sumber data internal dan eksternal. Data dapat mencakup data klinis dan data yang terkait dengan perilaku kesehatan, status sosial dan ekonomi, lingkungan fisik, klaim, biaya, dan keterlibatan pasien.

Organisasi perawatan kesehatan Anda juga harus dapat menghasilkan garis dasar untuk populasi target relatif terhadap suatu tujuan. Misalnya, untuk mencegah penyalahgunaan zat, sistem kesehatan harus memahami prevalensi pelecehan fisik, emosional, dan seksual dalam populasi. Mereka juga harus dapat menentukan populasi yang dapat mengambil manfaat dari intervensi, memahami total biaya perawatan, dan melakukan analisis berkelanjutan untuk memvalidasi apakah inisiatif memiliki efek yang diinginkan.

## Mengurangi biaya dengan mengoptimalkan operasi

Sistem Kesehatan menghadapi tantangan fiskal yang disebabkan oleh perubahan tingkat penggantian, peningkatan biaya tenaga kerja, peningkatan biaya untuk obat-obatan dan persediaan, dan inflasi. Sistem Kesehatan — yang biasanya beroperasi pada margin tipis dengan sumber daya terbatas — mendapat manfaat dari mengadopsi langkah-langkah penghematan biaya untuk mengoptimalkan penggunaan sumber daya mereka yang terbatas.

Data agregat yang komprehensif meningkatkan visibilitas ke dalam biaya yang terkait dengan intervensi di seluruh rangkaian perawatan. Sistem Kesehatan dapat menggunakan data ini untuk menemukan mekanisme baru yang mengurangi pengeluaran, menghasilkan pendapatan, dan

mempercepat arus kas. Dengan demikian, mereka dapat fokus pada menjaga pasien tetap sehat dan menjaga pintu rumah sakit tetap terbuka.

Strategi data perawatan kesehatan modern dapat membantu sistem kesehatan dapat menghemat biaya dengan:

- Mengoptimalkan penjadwalan dan perencanaan kapasitas berdasarkan alur pasien. Optimalisasi ini dapat mengurangi kelelahan penyedia sekaligus meningkatkan keterlibatan pasien.
- Memperkirakan kecenderungan membayar dengan menggunakan model prediktif, dan menggunakan data ini untuk mengembangkan strategi yang berbeda untuk mengumpulkan pembayaran.
- Memberi praktisi akses untuk menilai secara kritis data penelitian, pedoman klinis, dan sumber informasi lainnya untuk mengidentifikasi masalah klinis dengan benar. Praktisi kemudian dapat menerapkan intervensi berkualitas tinggi, dan mengevaluasi kembali hasil untuk hasil yang lebih baik di masa depan.

## Mengotomatiskan tugas untuk meningkatkan pengalaman penyedia

Dokter berjuang untuk menyeimbangkan perawatan pasien dengan volume tugas rutin yang harus mereka lakukan setiap hari. Mereka menjadi frustrasi ketika mereka tidak dapat mengakses data spesifik pasien yang komprehensif pada titik perawatan. Beban kerja dan jam kerja berlebihan, catatan medis tidak lengkap, dan lingkungan kerja seringkali menantang. Faktor-faktor ini berkontribusi pada tingkat kelelahan dan ketidakpuasan yang terus meningkat di antara para pekerja di organisasi terkait perawatan kesehatan.

Strategi data perawatan kesehatan modern dapat membantu meningkatkan pengalaman kerja dokter dan penyedia dengan:

- Memberikan dokter akses ke informasi historis tentang pasien sehingga mereka dapat memberikan perawatan berkualitas lebih tinggi kepada lebih banyak pasien, yang mengoptimalkan hasil pasien
- Mengotomatiskan tugas administratif, mengurangi beban penyedia
- Menciptakan pandangan pasien holistik dengan memberikan catatan medis yang komprehensif pada titik perawatan
- Menciptakan sistem yang memfasilitasi pertukaran catatan yang mulus antar penyedia
- Memfasilitasi pengelolaan persetujuan pasien dan persyaratan terkait kepatuhan lainnya

## Meningkatkan ekuitas dengan menggunakan data untuk memahami dan mengidentifikasi disparitas

Untuk meningkatkan hasil perawatan kesehatan bagi populasi yang luas, sistem kesehatan harus memahami di mana kesenjangan perawatan ada, berapa besarnya, dan alasan mengapa hal itu terjadi. Dengan informasi ini, organisasi dapat mulai mengembangkan rencana untuk meningkatkan perawatan semua pasien.

Organisasi perawatan kesehatan mungkin tidak menyadari hambatan yang dihadapi pasien selama perawatan biasa. Organizations mungkin juga tidak menyadari faktor-faktor di luar sistem perawatan kesehatan yang berperan dalam ketidakadilan kesehatan. Data hasil kesehatan adalah cara yang paling dapat diandalkan untuk mengidentifikasi jenis dan besarnya disparitas.

Strategi data perawatan kesehatan modern dapat membantu mengurangi kesenjangan perawatan kesehatan dengan:

- Memberikan pilihan perawatan yang mengatasi hambatan jarak, seperti sistem perawatan virtual, portal pasien, dan pemantauan pasien jarak jauh
- Memberikan solusi untuk meningkatkan akses ke layanan sosial, ketahanan pangan, transportasi, perumahan, atau peluang ekonomi
- Membuat atau mengkonsolidasikan kumpulan data untuk membuat kumpulan data yang kuat dan informatif
- Membersihkan kumpulan data yang ada untuk meningkatkan akurasi mengenai ras, etnis, jenis kelamin, kecacatan, atau faktor penentu ketidaksetaraan lainnya yang diketahui
- Mengoreksi bias algoritmik

## Memajukan perawatan kesehatan melalui penelitian genom

Informasi genom berperan penting dalam mengidentifikasi kelainan bawaan dan langka. Ini juga merupakan alat penting untuk mengkaraktirasi mutasi yang mendorong perkembangan kanker, dan untuk melacak wabah penyakit. Genomik merupakan inti dari kesehatan yang dipersonalisasi. Dengan mempertimbangkan variabilitas individu di antara orang-orang dan penyakit, dokter dapat membuat perjalanan perawatan yang dipersonalisasi dan perawatan yang ditargetkan.

Dengan mengadopsi strategi data perawatan kesehatan modern, organisasi penelitian dapat memajukan perawatan kesehatan dengan:

- Menentukan varian genetik untuk membantu diagnosis dan pengobatan penyakit, membantu menemukan biomarker penyakit dan target terapeutik potensial, dan memandu terapi yang ditargetkan.
- Mengidentifikasi informasi genotipe yang dapat digunakan untuk aplikasi klinis. Informasi ini dapat digunakan dalam pengembangan skor risiko poligenik yang digunakan untuk deteksi dini, pencegahan, atau pengobatan penyakit.
- Mengembangkan wawasan biologis dari data genom, yang dapat menginformasikan penemuan obat dan aplikasi klinis.
- Menggunakan genomik untuk lebih memahami evolusi suatu penyakit, melacak perkembangannya, dan mengembangkan tes dengan cepat.
- Menggunakan data multi-omik bersama dengan informasi klinis untuk memperoleh wawasan yang berguna tentang fungsi seluler.

## Meningkatkan keberlanjutan sistem perawatan kesehatan

Sistem perawatan kesehatan mengadopsi tujuan keberlanjutan baru. Untuk menentukan dan memenuhi tujuan sistem mereka, mereka mengeksplorasi alat-alat baru. Alat-alat ini dapat membantu mereka memahami dan mengoptimalkan tidak hanya jejak karbon TI mereka, tetapi juga bahan yang mereka gunakan, dan seluruh rantai pasokan yang memproduksi bahan-bahan ini. Untuk TI, penyimpanan dan pemrosesan data adalah komponen yang besar dan berkembang dari jejak karbon organisasi.

Dengan mengadopsi strategi data perawatan kesehatan modern, organisasi perawatan kesehatan dapat:

- Gunakan layanan cloud untuk mengoptimalkan penyimpanan TI dan penggunaan sumber daya pemrosesan data, dan migrasi beban kerja TI kesehatan ke energi terbarukan dan sumber daya air yang berkelanjutan.
- Menganalisis rantai pasokan untuk mengidentifikasi produk yang lebih berkelanjutan.

Seperti yang dinyatakan Amazon dalam [Ikrar Iklim](#), “Kami percaya kami memiliki kewajiban untuk menghentikan perubahan iklim, dan mengurangi emisi karbon menjadi nol akan berdampak besar. “Kami ingin mencapai emisi karbon nol bersih pada tahun 2040, satu dekade menjelang Perjanjian Iklim Paris, dan kami berada di jalur untuk memberdayakan operasi kami dengan 100% energi terbarukan pada tahun 2025 sebagai bagian dari tujuan kami untuk mencapai nol karbon bersih.”

Amazon mendokumentasikan pendekatan dan program keberlanjutannya di halaman beranda [Keberlanjutan Amazon](#). Secara khusus, AWS infrastruktur [3,6 kali lebih hemat energi](#) daripada median pusat data perusahaan AS yang disurvei oleh 451 Research, dan akan positif [air](#) pada tahun 2030. Keberlanjutan adalah pilar dalam [AWS Well-Architected](#) Framework, yang memandu pelanggan untuk mencapai praktik TI dan rantai pasokan yang berkelanjutan. AWS menyediakan [alat jejak karbon pelanggan](#) yang dapat digunakan pelanggan untuk memahami jejak karbon TI mereka. Pelanggan dapat menggunakan [Rantai Pasokan AWS](#) fitur untuk mengoptimalkan rantai pasokan mereka, termasuk dampak keberlanjutannya.

## Lampiran B. Memenuhi tujuan pasien

Pasien dan pengasuh mereka memiliki beragam tujuan dan harapan dalam hal perawatan kesehatan. Mereka ingin menerima perawatan yang aman dan efektif, dan membuat keputusan berdasarkan informasi tentang perawatan kesehatan mereka. Mereka juga ingin mengontrol siapa yang memiliki akses ke data perawatan kesehatan mereka dan bagaimana data itu digunakan.

Penyedia layanan kesehatan memiliki tanggung jawab etis dan hukum untuk memberikan kontrol kepada pasien atas Informasi Kesehatan yang Dilindungi (PHI) mereka. Di Amerika Serikat, Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan (HIPAA) menyatakan bahwa “individu memiliki hak untuk meninjau dan mendapatkan salinan PHI mereka, hak untuk membatasi pengungkapan PHI mereka, dan hak untuk akuntansi pengungkapan PHI mereka.” Untuk informasi selengkapnya, lihat [Ringkasan Aturan Privasi HIPAA](#). Sebagian besar negara anggota Uni Eropa mengakui hak pasien untuk menentukan nasib sendiri dan kerahasiaan sehubungan dengan PHI. Untuk informasi lebih lanjut, lihat laporan [Hak pasien di Uni Eropa](#). Di Jepang, kerangka peraturan dan sistem perawatan kesehatan memberi pasien hak dan kemampuan untuk mengelola, mendistribusikan, dan menggunakan PHI mereka. Untuk informasi lebih lanjut, lihat Proyek [Pemanfaatan Catatan Kesehatan Pribadi \(PHR\)](#).

Hak penentuan nasib sendiri dan privasi ini berarti penyedia layanan kesehatan harus dapat melacak dan melindungi data melalui setiap aspek arsitektur data, termasuk:

- Konsumsi data
- Pemrosesan
- Tetap
- Keamanan
- Tata kelola
- Federation
- Pembagian

Pada saat yang sama, pasien mengharapkan pengobatan yang cepat dan efektif dalam keadaan darurat. Oleh karena itu, perlindungan data harus dirancang agar tidak mengganggu kemampuan penyedia layanan kesehatan untuk merawat pasien secara efektif.

Bagian berikut membahas tujuan-tujuan ini, dan cara-cara di mana strategi data kesehatan modern dapat membantu memenuhinya.

## Mengelola persetujuan untuk perawatan dan penelitian

Saat menerima perawatan atau menjalani tes, pasien setuju untuk berbagi data perawatan kesehatan dengan penyedia layanan kesehatan. Ketentuan persetujuan itu biasanya jenis dan volume data yang dikumpulkan, siapa yang dapat mengakses data, dan bagaimana hal itu dapat digunakan. Di sebagian besar lingkungan peraturan, persyaratan ini harus mengikuti data terlepas dari bagaimana penyedia mengubah dan menyimpannya. Setiap orang yang mengakses data harus melakukannya dengan cara yang konsisten dengan persetujuan pasien.

Strategi data perawatan kesehatan modern harus secara eksplisit mendefinisikan hal-hal berikut:

- Bagaimana persetujuan pasien dibuat
- Bagaimana persetujuan itu tetap melekat pada data pasien
- Bagaimana sistem mengontrol akses dengan cara yang menghormati persetujuan pasien

Penting juga bagi sistem pelacakan persetujuan untuk menyertakan mekanisme untuk mengaudit akses data guna mengonfirmasi kepatuhan terhadap peraturan.

## Memberikan informasi pribadi kepada pasien

Pesatnya pertumbuhan informasi medis di internet telah membuatnya lebih menantang bagi pasien untuk menemukan informasi yang dapat dipercaya tentang kondisi dan standar perawatan mereka. Obat presisi menambah tantangan ini. Pengobatan presisi memperhitungkan perbedaan individu dalam gen, lingkungan, dan gaya hidup orang. Ada sejumlah besar kemungkinan genotipe. Ketika mereka dikalikan dengan jumlah variabel yang terkait dengan lingkungan dan gaya hidup, menjadi jelas bahwa setiap individu secara medis unik.

Ketika pasien mencari di internet untuk informasi tentang kondisi medis spesifik mereka—pilihan pengobatan, obat-obatan, terapi, pedoman diet dan olahraga, atau panduan lainnya—mereka menemukan banyak informasi. Namun, informasi tersebut dapat dibatasi dalam penerapannya pada situasi medis pribadi pasien. Pasien mungkin juga merasa sulit untuk memahami cakupan asuransi dan out-of-pocket biaya untuk pilihan perawatan yang berbeda. Dengan menggunakan strategi data perawatan kesehatan modern, organisasi perawatan kesehatan dapat membuka data dari silo dan membuatnya tersedia sehingga pasien dapat mengakses dan memahami informasi kesehatan pribadi mereka, menemukan informasi yang akurat tentang kondisi mereka, dan mendapatkan panduan yang bermanfaat dan tepat.

## Menghubungkan pasien dengan uji klinis

Penyakit langka, yang didefinisikan sebagai penyakit atau kondisi yang mempengaruhi sebagian kecil populasi, berdampak pada satu dari 17 orang, berjumlah lebih dari 400 juta orang di seluruh dunia. Tetapi sementara 7.000 penyakit langka telah diidentifikasi di AS saja, hanya 500 terapi yang telah disetujui oleh regulator. Uji coba penyakit langka berbeda secara signifikan dari uji coba 'biasa'. ... Pasien bisa sulit ditemukan, jumlahnya kecil dan menyebar ke seluruh dunia, berpotensi mempersulit proses rekrutmen dan pendaftaran.” Peter Buckman dan Dewan Pengembangan Bisnis Forbes, [Penyakit Langka: Unik Tapi Kurang Tertangani](#) Dalam Pengembangan Klinis

Pasien dengan kondisi yang tidak ada pengobatan yang disetujui, terutama penyakit langka, sangat tertarik untuk menemukan uji klinis untuk terapi baru. Tetapi bagi para peneliti, perekrutan pasien — kemampuan untuk mengidentifikasi dan mendaftarkan jumlah pasien yang tepat — adalah alasan utama mengapa uji klinis gagal. Strategi data perawatan kesehatan modern membantu pasien menemukan uji klinis yang paling cocok untuk kondisi pribadi mereka. Hal ini juga meningkatkan tingkat keberhasilan untuk uji klinis dengan membantu peneliti mengidentifikasi dan merekrut pasien yang tepat.

## Menyediakan portabilitas catatan kesehatan multimodal

Catatan kesehatan modern bersifat multimodal. Mereka berisi data catatan kesehatan elektronik tradisional (EHR), catatan radiologi, data sekuensing genom, data mikroskop elektron, sampel jaringan, data perangkat pasien, dan banyak lagi. Akibatnya, catatan medis pasien seringkali besar dan beragam. Pasien mungkin menerima data dari banyak penyedia dan berbagi data itu dengan penyedia dan pembayar lain.

Menyampaikan data yang besar dan kompleks menggunakan media fisik tidak lagi layak. Kesenjangan dalam catatan kesehatan dapat mengakibatkan kualitas perawatan yang buruk dan out-of-pocket biaya berlebih untuk pasien. Strategi data perawatan kesehatan modern mencakup mekanisme yang menyederhanakan proses penyampaian catatan kesehatan multimodal antara laboratorium, penyedia, dan pembayar.

## Lampiran C. Memenuhi tujuan TI sistem kesehatan

Industri perawatan kesehatan menghadapi tantangan untuk mengikuti lanskap politik, peraturan, ekonomi, dan teknologi yang berubah dengan cepat. Organizations perlu meningkatkan kelincahan dan kemampuan mereka untuk berinovasi dengan mengadopsi sistem teknis yang fleksibel dan mudah beradaptasi dengan perubahan kondisi.

Volume data perawatan kesehatan yang dikelola organisasi meningkat setiap tahun, membawa serta peningkatan biaya untuk penyimpanan, pencadangan dan pemulihan, manajemen basis data, dan daya komputasi. Pada saat yang sama, organisasi perawatan kesehatan menghadapi tekanan biaya dan peraturan. Sebagai akibat dari tekanan ini, organisasi sering mencari cara untuk mengurangi biaya operasional sambil tetap mematuhi persyaratan peraturan.

Bagian berikut menjelaskan cara-cara di mana strategi data perawatan kesehatan modern dapat membantu organisasi memenuhi tujuan dan persyaratan terkait TI.

### Meningkatkan kelincahan dan kemampuan untuk berinovasi

Organizations dalam industri kesehatan harus semakin gesit agar berhasil. Industri terus melihat pertumbuhan sebagai berikut:

- Jumlah merger dan akuisisi
- Kepemilikan praktik dokter oleh organisasi perawatan kesehatan besar
- Adopsi pengaturan perawatan berbasis nilai

Sementara itu, konsumen semakin diberdayakan dalam membuat keputusan perawatan, sementara pembayar dan penyedia sedang mengeksplorasi teknologi seperti pemantauan kesehatan rumah, telehealth, dan aplikasi seluler.

Penting bagi organisasi perawatan kesehatan untuk memiliki sistem teknologi yang mampu beradaptasi dengan perubahan kondisi, termasuk perubahan tak terduga dalam kebutuhan perawatan kesehatan. Misalnya, ketika pandemi COVID-19 mengganggu industri perawatan kesehatan, organisasi perawatan kesehatan, produsen, dan lembaga pendidikan membutuhkan teknologi yang memungkinkan individu untuk bekerja dari lokasi yang aman. Banyak organisasi perawatan kesehatan juga perlu meningkatkan operasi mereka secara besar-besaran untuk melakukan penelitian dalam ilmu dasar, ilmu klinis, dan ilmu kesehatan masyarakat.

## Mengurangi biaya operasional

Organisasi perawatan kesehatan menghadapi kekurangan profesional medis, masalah aksesibilitas perawatan kesehatan, populasi yang menua, peningkatan penyalahgunaan zat, dan peningkatan tingkat penyakit kronis. Pada saat yang sama, mereka menghadapi tekanan dari pasien untuk memberikan perawatan berkualitas lebih tinggi dengan biaya lebih rendah out-of-pocket.

Pemerintah di seluruh dunia sedang mengevaluasi atau menerapkan reformasi pembayaran untuk membantu penyedia mengurangi biaya dan meningkatkan efisiensi sambil meningkatkan hasil dan mendorong keterlibatan pasien. Program-program ini kadang-kadang disebut sebagai gaji untuk kinerja, perawatan berbasis nilai, atau perawatan yang bertanggung jawab. Reformasi ini, bagaimanapun, memerlukan informasi rinci tentang kondisi, prosedur, dan biaya dalam sistem kesehatan.

Organisasi kesehatan dapat berinovasi dan menurunkan biaya dengan mengadopsi strategi data perawatan kesehatan modern. Dengan strategi modern, organisasi dapat mengidentifikasi data yang perlu mereka simpan untuk memenuhi persyaratan peraturan dan menghapus data yang berlebihan. Mereka juga dapat menggunakan penyimpanan tingkat arsip di cloud untuk menurunkan biaya penyimpanan jangka panjang. Data arsip ini dapat diambil dalam hitungan jam untuk penggunaan jangka pendek, seperti studi longitudinal atau untuk menghasilkan statistik kesehatan populasi.

## Memodernisasi penyimpanan dan analitik data

Selama dekade terakhir, volume data perawatan kesehatan yang dikumpulkan organisasi telah meningkat secara eksponensial. Penyedia layanan kesehatan dan pembayar menggunakan data ini untuk mendukung analitik canggih, pembelajaran mesin, dan sistem kecerdasan buatan yang meningkatkan kualitas perawatan. Penyedia juga menggunakan data ini untuk mengidentifikasi dan mengatasi risiko dengan lebih cepat dan akurat terhadap beban kerja operasional dan klinis inti. Demikian juga, pembayar dapat menilai risiko secara lebih akurat dan efisien melalui otomatisasi jaringan pipa pemrosesan klaim. Dengan menggunakan pintu depan digital modern yang mengakomodasi data dari perangkat kesehatan konsumen seperti perangkat yang dapat dikenakan, penyedia layanan dapat lebih memahami gaya hidup pasien dan memprediksi hasil kesehatan dengan lebih baik.

Untuk menggunakan kumpulan data besar ini secara efektif, penting bagi penyedia untuk menerapkan sistem manajemen operasi data. Selain itu, untuk melindungi kelangsungan dan ketahanan bisnis, mereka perlu membuat sistem dan proses yang mengelola keamanan data, ketersediaan data, dan daya tahan. Mereka membutuhkan penyimpanan data yang elastis

(penyimpanan yang dapat menyusut atau tumbuh seiring kebutuhan data berubah). Sistem penyimpanan harus memenuhi persyaratan kinerja untuk berbagai macam beban kerja. Akhirnya, sistem harus dioptimalkan untuk menciptakan keseimbangan akses, ketekunan, dan biaya yang diperlukan. Strategi data perawatan kesehatan modern yang dirancang dengan baik dapat memenuhi semua persyaratan ini.

## Lampiran D. Panduan tambahan tentang penerapan strategi data kesehatan modern

Organizations dapat menerapkan strategi data perawatan kesehatan modern dengan berbagai cara. Rincian implementasi spesifik untuk organisasi bergantung pada infrastruktur data yang ada, ketersediaan insinyur untuk membangun dan menyebarkan komponen teknis, dan waktu yang dialokasikan untuk implementasi.

Organisasi layanan kesehatan dapat membangun atau membeli komponen sistem data, tergantung pada infrastruktur, kemampuan, dan hubungan mereka yang ada dengan penyedia teknologi. Organizations yang membutuhkan solusi data siap pakai dapat memilih solusi perangkat lunak sebagai layanan (SaaS), yang mengurangi waktu dan upaya implementasi. Organizations yang memilih solusi SaaS harus memastikan bahwa solusi tersebut memenuhi kebutuhan mereka untuk konsumsi data, pemrosesan, dan analitik. Mereka juga harus mengonfirmasi bahwa ia dapat beroperasi dengan layanan cloud lainnya untuk memenuhi kebutuhan ini.

Sebagai alternatif, organisasi dapat membangun solusi data menggunakan data cloud dan layanan analitik. Pendekatan ini adalah yang paling fleksibel. Namun, itu membutuhkan keahlian dan sumber daya. Solusi yang dibuat khusus memberi organisasi kontrol penuh atas penyimpanan dan pemrosesan data. Pendekatan ini juga mengurangi kemungkinan organisasi melampaui strategi data mereka. Membangun solusi data perawatan kesehatan membutuhkan organisasi untuk berinvestasi pada para ahli untuk mengembangkan dan memelihara infrastruktur cloud. Seiring waktu, para ahli ini menjadi aset organisasi utama. Selain itu, konsultan cloud, seperti [Layanan AWS Profesional](#) dan anggota [AWS Partner Network](#), dapat mempercepat kemampuan dan meningkatkan nilai ketika mengembangkan komponen solusi data. Organizations yang membangun strategi data perawatan kesehatan modern juga harus mempertimbangkan pemeliharaan berkelanjutan dari solusi data cloud mereka, yang sering berarti mempekerjakan insinyur operasi cloud.

Organizations juga dapat mempertimbangkan untuk mengadopsi solusi platform as a service (PaaS) untuk data cloud. Solusi ini menyederhanakan alur kerja pemrosesan data umum sehingga organisasi dapat mencurahkan lebih banyak waktu dan sumber daya untuk memperoleh wawasan dari data mereka. Solusi PaaS membantu mengurangi waktu dan upaya yang diperlukan untuk menerapkan dan memelihara solusi data cloud sekaligus memungkinkan organisasi untuk mempertahankan tingkat fleksibilitas dan kontrol yang tinggi. Solusi PaaS membutuhkan insinyur cloud yang dilatih secara khusus dalam pemeliharaan dan penggunaan solusi data, yang meningkatkan kompleksitas perekrutan dan pelatihan insinyur cloud.

Akhirnya, organisasi juga harus mempertimbangkan persyaratan keamanan dan kepatuhan mereka ketika membangun strategi data perawatan kesehatan modern. Saat menggunakan solusi PaaS dan SaaS, organisasi harus bekerja dengan penyedia solusi untuk mengklarifikasi persyaratan dan tanggung jawab ini. Membangun solusi data membutuhkan insinyur yang berpengalaman dalam praktik terbaik keamanan dan kepatuhan untuk cloud. AWS menyediakan sumber daya seperti [Referensi Layanan yang Memenuhi Syarat HIPAA](#). Sumber daya ini membantu memandu dan melatih arsitek dan insinyur cloud dalam mencapai tujuan keamanan dan kepatuhan.

Solusi data yang mendukung strategi data perawatan kesehatan modern harus memungkinkan organisasi memperoleh nilai dari semua aset data mereka. Ini harus dilakukan sambil memberikan lingkungan yang aman, terukur, berkinerja tinggi, berkelanjutan, dan easy-to-use lingkungan untuk mengakses, menganalisis, dan memperoleh wawasan dari data. Fitur utama meliputi:

- Persyaratan keamanan dan kepatuhan ditangani melalui pencatatan, kontrol akses yang halus, serta pemantauan dan peringatan terpusat.
- Dukungan untuk resolusi entitas, anonimisasi PHI dan informasi identitas pribadi (PII), model data yang berpusat pada pasien, dan manajemen persetujuan pasien.
- Penyimpanan data khusus yang dirancang untuk kebutuhan spesifik. Kebutuhan ini dapat mencakup dokumen, log, gambar, pasangan nilai kunci, dan data semi-terstruktur dan tidak terstruktur.
- Manajemen data federasi, dengan penemuan data terpusat, audit, dan tata kelola menggunakan kerangka kerja untuk federasi data.
- Support untuk beragam kasus penggunaan data melalui model data umum, seperti [Observational Medical Outcome Partnership \(OMOP\) Common Data Model](#) dan [Informatics for Integrating Biology and the Bedside](#) (i2b2) framework.
- Interoperabilitas dan berbagi data dengan menggunakan standar seperti berikut:
  - [Health Level Seven Internasional \(HL7\) V2](#)
  - HL7 [Sumber Daya Interoperabilitas Perawatan Kesehatan Cepat \(FHIR\)](#)
  - HL7 [Arsitektur Dokumen Klinis Konsolidasi \(C-CDA\)](#)
  - Saran pengiriman uang EDI 835
  - Dokumen klaim EDI 837

AWS menawarkan rangkaian layanan dan kemampuan yang kuat untuk mengatasi setiap aspek arsitektur data perawatan kesehatan modern. Menerapkan beban kerja pada AWS membawa manfaat berikut:

- Kelincahan — Tim dapat bereksperimen dan berinovasi dengan cepat dan sering, tanpa memengaruhi sistem produksi.
- Elastisitas — Sumber daya dapat ditingkatkan dan diturunkan seiring kebutuhan bisnis berubah.
- Penghematan biaya — Hanya sumber daya yang digunakan yang mengeluarkan biaya.
- Innovation — Organizations dapat fokus pada pembeda bisnis, bukan infrastruktur.
- Keamanan dan kepatuhan - infrastruktur AWS inti dibangun untuk memenuhi persyaratan keamanan untuk organisasi dengan sensitivitas tinggi. Ini didukung oleh seperangkat alat keamanan cloud yang dalam, dengan lebih dari 300 layanan dan fitur keamanan, kepatuhan, dan tata kelola. AWS mendukung 143 standar keamanan dan sertifikasi kepatuhan, termasuk:
  - Standar Keamanan Data Industri Kartu Pembayaran (PCI-DSS)
  - HIPAA dan Undang-Undang Teknologi Informasi Kesehatan untuk Kesehatan Ekonomi dan Klinis (HITECH)
  - Program Manajemen Risiko dan Otorisasi Federal (FedRAMP)
  - Peraturan Perlindungan Data Umum (GDPR)
  - Standar Pemrosesan Informasi Federal (FIPS) 140-2
  - Institut Nasional Standar dan Teknologi (NIST) 800-171

# Kontributor

Kontributor untuk panduan ini meliputi:

- Madhu Bussa, Manajer, Arsitek Solusi, AWS
- Mark Garcia, Manajer Pengembangan Bisnis Utama - Kedokteran Akademik, AWS
- Kas Parthasarathy, Manajer, Arsitek Solusi Kesehatan, AWS
- Rod Tarrago, Manajer Pengembangan Bisnis Utama — Kedokteran Akademik, AWS
- Paul Saxman, Pemimpin Teknis, AWS
- Scott Glasser, Arsitek Solusi Utama, AWS

## Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada panduan ini. Jika Anda ingin diberi tahu tentang pembaruan masa depan, Anda dapat berlangganan umpan [RSS](#).

Perubahan	Deskripsi	Tanggal
<a href="#">Publikasi awal</a>	—	16 November 2023

# AWS Glosarium Panduan Preskriptif

Berikut ini adalah istilah yang umum digunakan dalam strategi, panduan, dan pola yang disediakan oleh Panduan AWS Preskriptif. Untuk menyarankan entri, silakan gunakan tautan Berikan umpan balik di akhir glosarium.

## Nomor

### 7 Rs

Tujuh strategi migrasi umum untuk memindahkan aplikasi ke cloud. Strategi ini dibangun di atas 5 Rs yang diidentifikasi Gartner pada tahun 2011 dan terdiri dari yang berikut:

- Refactor/Re-Architect — Memindahkan aplikasi dan memodifikasi arsitekturnya dengan memanfaatkan sepenuhnya fitur cloud-native untuk meningkatkan kelincahan, kinerja, dan skalabilitas. Ini biasanya melibatkan porting sistem operasi dan database. Contoh: Migrasikan database Oracle lokal Anda ke Amazon Aurora PostgreSQL Compatible Edition.
- Replatform (angkat dan bentuk ulang) — Pindahkan aplikasi ke cloud, dan perkenalkan beberapa tingkat pengoptimalan untuk memanfaatkan kemampuan cloud. Contoh: Memigrasikan database Oracle lokal Anda ke Amazon Relational Database Service (Amazon RDS) untuk Oracle di AWS Cloud
- Pembelian kembali (drop and shop) - Beralih ke produk yang berbeda, biasanya dengan beralih dari lisensi tradisional ke model SaaS. Contoh: Migrasikan sistem manajemen hubungan pelanggan (CRM) Anda ke Salesforce.com.
- Rehost (lift dan shift) — Pindahkan aplikasi ke cloud tanpa membuat perubahan apa pun untuk memanfaatkan kemampuan cloud. Contoh: Migrasikan database Oracle lokal Anda ke Oracle pada instans EC2 di AWS Cloud
- Relokasi (hypervisor-level lift and shift) — Pindahkan infrastruktur ke cloud tanpa membeli perangkat keras baru, menulis ulang aplikasi, atau memodifikasi operasi yang ada. Anda memigrasikan server dari platform lokal ke layanan cloud untuk platform yang sama. Contoh: Migrasikan Microsoft Hyper-V aplikasi ke AWS.
- Pertahankan (kunjungi kembali) - Simpan aplikasi di lingkungan sumber Anda. Ini mungkin termasuk aplikasi yang memerlukan refactoring besar, dan Anda ingin menunda pekerjaan itu sampai nanti, dan aplikasi lama yang ingin Anda pertahankan, karena tidak ada pembenaran bisnis untuk memigrasikannya.

- Pensiun — Menonaktifkan atau menghapus aplikasi yang tidak lagi diperlukan di lingkungan sumber Anda.

## A

### ABAC

Lihat [kontrol akses berbasis atribut](#).

### layanan abstrak

Lihat [layanan terkelola](#).

### ASAM

Lihat [atomisitas, konsistensi, isolasi, daya tahan](#).

### migrasi aktif-aktif

Metode migrasi database di mana database sumber dan target tetap sinkron (dengan menggunakan alat replikasi dua arah atau operasi penulisan ganda), dan kedua database menangani transaksi dari menghubungkan aplikasi selama migrasi. Metode ini mendukung migrasi dalam batch kecil yang terkontrol alih-alih memerlukan pemotongan satu kali. Ini lebih fleksibel tetapi membutuhkan lebih banyak pekerjaan daripada migrasi [aktif-pasif](#).

### migrasi aktif-pasif

Metode migrasi database di mana database sumber dan target disimpan dalam sinkron, tetapi hanya database sumber yang menangani transaksi dari menghubungkan aplikasi sementara data direplikasi ke database target. Basis data target tidak menerima transaksi apa pun selama migrasi.

### fungsi agregat

Fungsi SQL yang beroperasi pada sekelompok baris dan menghitung nilai pengembalian tunggal untuk grup. Contoh fungsi agregat meliputi SUM dan MAX.

## AI

Lihat [kecerdasan buatan](#).

### AIOps

Lihat [operasi kecerdasan buatan](#).

## anonimisasi

Proses menghapus informasi pribadi secara permanen dalam kumpulan data. Anonimisasi dapat membantu melindungi privasi pribadi. Data anonim tidak lagi dianggap sebagai data pribadi.

## anti-pola

Solusi yang sering digunakan untuk masalah berulang di mana solusinya kontra-produktif, tidak efektif, atau kurang efektif daripada alternatif.

## kontrol aplikasi

Pendekatan keamanan yang memungkinkan penggunaan hanya aplikasi yang disetujui untuk membantu melindungi sistem dari malware.

## portofolio aplikasi

Kumpulan informasi rinci tentang setiap aplikasi yang digunakan oleh organisasi, termasuk biaya untuk membangun dan memelihara aplikasi, dan nilai bisnisnya. Informasi ini adalah kunci untuk [penemuan portofolio dan proses analisis dan](#) membantu mengidentifikasi dan memprioritaskan aplikasi yang akan dimigrasi, dimodernisasi, dan dioptimalkan.

## kecerdasan buatan (AI)

Bidang ilmu komputer yang didedikasikan untuk menggunakan teknologi komputasi untuk melakukan fungsi kognitif yang biasanya terkait dengan manusia, seperti belajar, memecahkan masalah, dan mengenali pola. Untuk informasi lebih lanjut, lihat [Apa itu Kecerdasan Buatan?](#)

## operasi kecerdasan buatan (AIOps)

Proses menggunakan teknik pembelajaran mesin untuk memecahkan masalah operasional, mengurangi insiden operasional dan intervensi manusia, dan meningkatkan kualitas layanan. Untuk informasi selengkapnya tentang cara AIOps digunakan dalam strategi AWS migrasi, lihat [panduan integrasi operasi](#).

## enkripsi asimetris

Algoritma enkripsi yang menggunakan sepasang kunci, kunci publik untuk enkripsi dan kunci pribadi untuk dekripsi. Anda dapat berbagi kunci publik karena tidak digunakan untuk dekripsi, tetapi akses ke kunci pribadi harus sangat dibatasi.

## atomisitas, konsistensi, isolasi, daya tahan (ACID)

Satu set properti perangkat lunak yang menjamin validitas data dan keandalan operasional database, bahkan dalam kasus kesalahan, kegagalan daya, atau masalah lainnya.

## kontrol akses berbasis atribut (ABAC)

Praktik membuat izin berbutir halus berdasarkan atribut pengguna, seperti departemen, peran pekerjaan, dan nama tim. Untuk informasi selengkapnya, lihat [ABAC untuk AWS](#) dokumentasi AWS Identity and Access Management (IAM).

## sumber data otoritatif

Lokasi di mana Anda menyimpan versi utama data, yang dianggap sebagai sumber informasi yang paling dapat diandalkan. Anda dapat menyalin data dari sumber data otoritatif ke lokasi lain untuk tujuan pemrosesan atau modifikasi data, seperti menganonimkan, menyunting, atau membuat nama samaran.

## Zona Ketersediaan

Lokasi berbeda di dalam Wilayah AWS yang terisolasi dari kegagalan di Availability Zone lainnya dan menyediakan konektivitas jaringan latensi rendah yang murah ke Availability Zone lainnya di Wilayah yang sama.

## AWS Kerangka Adopsi Cloud (AWS CAF)

Kerangka pedoman dan praktik terbaik AWS untuk membantu organisasi mengembangkan rencana yang efisien dan efektif untuk bergerak dengan sukses ke cloud. AWS CAF mengatur panduan ke dalam enam area fokus yang disebut perspektif: bisnis, orang, tata kelola, platform, keamanan, dan operasi. Perspektif bisnis, orang, dan tata kelola fokus pada keterampilan dan proses bisnis; perspektif platform, keamanan, dan operasi fokus pada keterampilan dan proses teknis. Misalnya, perspektif masyarakat menargetkan pemangku kepentingan yang menangani sumber daya manusia (SDM), fungsi kepegawaian, dan manajemen orang. Untuk perspektif ini, AWS CAF memberikan panduan untuk pengembangan, pelatihan, dan komunikasi orang untuk membantu mempersiapkan organisasi untuk adopsi cloud yang sukses. Untuk informasi lebih lanjut, lihat [situs web AWS CAF dan whitepaper AWS CAF](#).

## AWS Kerangka Kualifikasi Beban Kerja (AWS WQF)

Alat yang mengevaluasi beban kerja migrasi database, merekomendasikan strategi migrasi, dan memberikan perkiraan kerja. AWS WQF disertakan dengan AWS Schema Conversion Tool (AWS SCT). Ini menganalisis skema database dan objek kode, kode aplikasi, dependensi, dan karakteristik kinerja, dan memberikan laporan penilaian.

## B

bot buruk

[Bot](#) yang dimaksudkan untuk mengganggu atau menyebabkan kerugian bagi individu atau organisasi.

BCP

Lihat [perencanaan kontinuitas bisnis](#).

grafik perilaku

Pandangan interaktif yang terpadu tentang perilaku dan interaksi sumber daya dari waktu ke waktu. Anda dapat menggunakan grafik perilaku dengan Amazon Detective untuk memeriksa upaya logon yang gagal, panggilan API yang mencurigakan, dan tindakan serupa. Untuk informasi selengkapnya, lihat [Data dalam grafik perilaku](#) di dokumentasi Detektif.

sistem big-endian

Sistem yang menyimpan byte paling signifikan terlebih dahulu. Lihat juga [endianness](#).

klasifikasi biner

Sebuah proses yang memprediksi hasil biner (salah satu dari dua kelas yang mungkin). Misalnya, model ML Anda mungkin perlu memprediksi masalah seperti “Apakah email ini spam atau bukan spam?” atau “Apakah produk ini buku atau mobil?”

filter mekar

Struktur data probabilistik dan efisien memori yang digunakan untuk menguji apakah suatu elemen adalah anggota dari suatu himpunan.

deployment biru/hijau

Strategi penyebaran tempat Anda membuat dua lingkungan yang terpisah namun identik. Anda menjalankan versi aplikasi saat ini di satu lingkungan (biru) dan versi aplikasi baru di lingkungan lain (hijau). Strategi ini membantu Anda dengan cepat memutar kembali dengan dampak minimal.

bot

Aplikasi perangkat lunak yang menjalankan tugas otomatis melalui internet dan mensimulasikan aktivitas atau interaksi manusia. Beberapa bot berguna atau bermanfaat, seperti perayap web yang mengindeks informasi di internet. Beberapa bot lain, yang dikenal sebagai bot buruk, dimaksudkan untuk mengganggu atau membahayakan individu atau organisasi.

## botnet

Jaringan [bot](#) yang terinfeksi oleh [malware](#) dan berada di bawah kendali satu pihak, yang dikenal sebagai bot herder atau operator bot. Botnet adalah mekanisme paling terkenal untuk skala bot dan dampaknya.

## cabang

Area berisi repositori kode. Cabang pertama yang dibuat dalam repositori adalah cabang utama. Anda dapat membuat cabang baru dari cabang yang ada, dan Anda kemudian dapat mengembangkan fitur atau memperbaiki bug di cabang baru. Cabang yang Anda buat untuk membangun fitur biasanya disebut sebagai cabang fitur. Saat fitur siap dirilis, Anda menggabungkan cabang fitur kembali ke cabang utama. Untuk informasi selengkapnya, lihat [Tentang cabang](#) (GitHub dokumentasi).

## akses break-glass

Dalam keadaan luar biasa dan melalui proses yang disetujui, cara cepat bagi pengguna untuk mendapatkan akses ke Akun AWS yang biasanya tidak memiliki izin untuk mengaksesnya. Untuk informasi lebih lanjut, lihat indikator [Implementasikan prosedur break-glass](#) dalam panduan Well-Architected AWS .

## strategi brownfield

Infrastruktur yang ada di lingkungan Anda. Saat mengadopsi strategi brownfield untuk arsitektur sistem, Anda merancang arsitektur di sekitar kendala sistem dan infrastruktur saat ini. Jika Anda memperluas infrastruktur yang ada, Anda dapat memadukan strategi brownfield dan [greenfield](#).

## cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

## kemampuan bisnis

Apa yang dilakukan bisnis untuk menghasilkan nilai (misalnya, penjualan, layanan pelanggan, atau pemasaran). Arsitektur layanan mikro dan keputusan pengembangan dapat didorong oleh kemampuan bisnis. Untuk informasi selengkapnya, lihat bagian [Terorganisir di sekitar kemampuan bisnis](#) dari [Menjalankan layanan mikro kontainer](#) di whitepaper. AWS

## perencanaan kelangsungan bisnis (BCP)

Rencana yang membahas dampak potensial dari peristiwa yang mengganggu, seperti migrasi skala besar, pada operasi dan memungkinkan bisnis untuk melanjutkan operasi dengan cepat.

## C

### KAFE

Lihat [Kerangka Adopsi AWS Cloud](#).

penyebaran kenari

Rilis versi yang lambat dan bertahap untuk pengguna akhir. Ketika Anda yakin, Anda menyebarkan versi baru dan mengganti versi saat ini secara keseluruhan.

### CCoE

Lihat [Cloud Center of Excellence](#).

### CDC

Lihat [mengubah pengambilan data](#).

ubah pengambilan data (CDC)

Proses melacak perubahan ke sumber data, seperti tabel database, dan merekam metadata tentang perubahan tersebut. Anda dapat menggunakan CDC untuk berbagai tujuan, seperti mengaudit atau mereplikasi perubahan dalam sistem target untuk mempertahankan sinkronisasi.

rekayasa kekacauan

Sengaja memperkenalkan kegagalan atau peristiwa yang mengganggu untuk menguji ketahanan sistem. Anda dapat menggunakan [AWS Fault Injection Service \(AWS FIS\)](#) untuk melakukan eksperimen yang menekankan AWS beban kerja Anda dan mengevaluasi responsnya.

### CI/CD

Lihat [integrasi berkelanjutan dan pengiriman berkelanjutan](#).

klasifikasi

Proses kategorisasi yang membantu menghasilkan prediksi. Model ML untuk masalah klasifikasi memprediksi nilai diskrit. Nilai diskrit selalu berbeda satu sama lain. Misalnya, model mungkin perlu mengevaluasi apakah ada mobil dalam gambar atau tidak.

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

## Pusat Keunggulan Cloud (CCoE)

Tim multi-disiplin yang mendorong upaya adopsi cloud di seluruh organisasi, termasuk mengembangkan praktik terbaik cloud, memobilisasi sumber daya, menetapkan jadwal migrasi, dan memimpin organisasi melalui transformasi skala besar. Untuk informasi selengkapnya, lihat [posting CCo E](#) di Blog Strategi AWS Cloud Perusahaan.

### komputasi cloud

Teknologi cloud yang biasanya digunakan untuk penyimpanan data jarak jauh dan manajemen perangkat IoT. Cloud computing umumnya terhubung ke teknologi [edge computing](#).

### model operasi cloud

Dalam organisasi TI, model operasi yang digunakan untuk membangun, mematangkan, dan mengoptimalkan satu atau lebih lingkungan cloud. Untuk informasi selengkapnya, lihat [Membangun Model Operasi Cloud Anda](#).

### tahap adopsi cloud

Empat fase yang biasanya dilalui organisasi ketika mereka bermigrasi ke AWS Cloud:

- Proyek — Menjalankan beberapa proyek terkait cloud untuk bukti konsep dan tujuan pembelajaran
- Foundation — Melakukan investasi dasar untuk meningkatkan adopsi cloud Anda (misalnya, membuat landing zone, mendefinisikan CCo E, membuat model operasi)
- Migrasi — Migrasi aplikasi individual
- Re-invention — Mengoptimalkan produk dan layanan, dan berinovasi di cloud

Tahapan ini didefinisikan oleh Stephen Orban dalam posting blog [The Journey Toward Cloud-First & the Stages of Adoption](#) di blog Strategi Perusahaan. AWS Cloud Untuk informasi tentang bagaimana kaitannya dengan strategi AWS migrasi, lihat [panduan kesiapan migrasi](#).

### CMDB

Lihat [database manajemen konfigurasi](#).

### repositori kode

Lokasi di mana kode sumber dan aset lainnya, seperti dokumentasi, sampel, dan skrip, disimpan dan diperbarui melalui proses kontrol versi. Repositori cloud umum termasuk GitHub atau Bitbucket Cloud. Setiap versi kode disebut cabang. Dalam struktur layanan mikro, setiap repositori

dikhususkan untuk satu bagian fungsionalitas. Pipa CI/CD tunggal dapat menggunakan beberapa repositori.

#### cache dingin

Cache buffer yang kosong, tidak terisi dengan baik, atau berisi data basi atau tidak relevan. Ini mempengaruhi kinerja karena instance database harus membaca dari memori utama atau disk, yang lebih lambat daripada membaca dari cache buffer.

#### data dingin

Data yang jarang diakses dan biasanya historis. Saat menanyakan jenis data ini, kueri lambat biasanya dapat diterima. Memindahkan data ini ke tingkat penyimpanan atau kelas yang berkinerja lebih rendah dan lebih murah dapat mengurangi biaya.

#### visi komputer (CV)

Bidang [AI](#) yang menggunakan pembelajaran mesin untuk menganalisis dan mengekstrak informasi dari format visual seperti gambar dan video digital. Misalnya, Amazon SageMaker AI menyediakan algoritma pemrosesan gambar untuk CV.

#### konfigurasi drift

Untuk beban kerja, konfigurasi berubah dari status yang diharapkan. Ini dapat menyebabkan beban kerja menjadi tidak patuh, dan biasanya bertahap dan tidak disengaja.

#### database manajemen konfigurasi (CMDB)

Repositori yang menyimpan dan mengelola informasi tentang database dan lingkungan TI, termasuk komponen perangkat keras dan perangkat lunak dan konfigurasinya. Anda biasanya menggunakan data dari CMDB dalam penemuan portofolio dan tahap analisis migrasi.

#### paket kesesuaian

Kumpulan AWS Config aturan dan tindakan remediasi yang dapat Anda kumpulkan untuk menyesuaikan kepatuhan dan pemeriksaan keamanan Anda. Anda dapat menerapkan paket kesesuaian sebagai entitas tunggal di Akun AWS dan Wilayah, atau di seluruh organisasi, dengan menggunakan templat YAMM. Untuk informasi selengkapnya, lihat [Paket kesesuaian dalam dokumentasi](#). AWS Config

#### integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

Proses mengotomatiskan sumber, membangun, menguji, pementasan, dan tahap produksi dari proses rilis perangkat lunak. CI/CD biasanya digambarkan sebagai pipa. CI/CD dapat membantu

Anda mengotomatiskan proses, meningkatkan produktivitas, meningkatkan kualitas kode, dan memberikan lebih cepat. Untuk informasi lebih lanjut, lihat [Manfaat pengiriman berkelanjutan](#). CD juga dapat berarti penerapan berkelanjutan. Untuk informasi selengkapnya, lihat [Continuous Delivery vs Continuous Deployment](#).

## CV

Lihat [visi komputer](#).

## D

### data saat istirahat

Data yang stasioner di jaringan Anda, seperti data yang ada di penyimpanan.

### klasifikasi data

Proses untuk mengidentifikasi dan mengkategorikan data dalam jaringan Anda berdasarkan kekritisannya dan sensitivitasnya. Ini adalah komponen penting dari setiap strategi manajemen risiko keamanan siber karena membantu Anda menentukan perlindungan dan kontrol retensi yang tepat untuk data. Klasifikasi data adalah komponen pilar keamanan dalam AWS Well-Architected Framework. Untuk informasi selengkapnya, lihat [Klasifikasi data](#).

### penyimpangan data

Variasi yang berarti antara data produksi dan data yang digunakan untuk melatih model ML, atau perubahan yang berarti dalam data input dari waktu ke waktu. Penyimpangan data dapat mengurangi kualitas, akurasi, dan keadilan keseluruhan dalam prediksi model ML.

### data dalam transit

Data yang aktif bergerak melalui jaringan Anda, seperti antara sumber daya jaringan.

### jala data

Kerangka arsitektur yang menyediakan kepemilikan data terdistribusi dan terdesentralisasi dengan manajemen dan tata kelola terpusat.

### minimalisasi data

Prinsip pengumpulan dan pemrosesan hanya data yang sangat diperlukan. Mempraktikkan minimalisasi data di dalamnya AWS Cloud dapat mengurangi risiko privasi, biaya, dan jejak karbon analitik Anda.

## perimeter data

Satu set pagar pembatas pencegahan di AWS lingkungan Anda yang membantu memastikan bahwa hanya identitas tepercaya yang mengakses sumber daya tepercaya dari jaringan yang diharapkan. Untuk informasi selengkapnya, lihat [Membangun perimeter data pada AWS](#).

## prapemrosesan data

Untuk mengubah data mentah menjadi format yang mudah diuraikan oleh model ML Anda. Preprocessing data dapat berarti menghapus kolom atau baris tertentu dan menangani nilai yang hilang, tidak konsisten, atau duplikat.

## asal data

Proses melacak asal dan riwayat data sepanjang siklus hidupnya, seperti bagaimana data dihasilkan, ditransmisikan, dan disimpan.

## subjek data

Individu yang datanya dikumpulkan dan diproses.

## gudang data

Sistem manajemen data yang mendukung intelijen bisnis, seperti analitik. Gudang data biasanya berisi sejumlah besar data historis, dan biasanya digunakan untuk kueri dan analisis.

## bahasa definisi database (DDL)

Pernyataan atau perintah untuk membuat atau memodifikasi struktur tabel dan objek dalam database.

## bahasa manipulasi basis data (DHTML)

Pernyataan atau perintah untuk memodifikasi (memasukkan, memperbarui, dan menghapus) informasi dalam database.

## DDL

Lihat [bahasa definisi database](#).

## ansambel yang dalam

Untuk menggabungkan beberapa model pembelajaran mendalam untuk prediksi. Anda dapat menggunakan ansambel dalam untuk mendapatkan prediksi yang lebih akurat atau untuk memperkirakan ketidakpastian dalam prediksi.

## pembelajaran mendalam

Subbidang ML yang menggunakan beberapa lapisan jaringan saraf tiruan untuk mengidentifikasi pemetaan antara data input dan variabel target yang diinginkan.

## defense-in-depth

Pendekatan keamanan informasi di mana serangkaian mekanisme dan kontrol keamanan dilapisi dengan cermat di seluruh jaringan komputer untuk melindungi kerahasiaan, integritas, dan ketersediaan jaringan dan data di dalamnya. Saat Anda mengadopsi strategi ini AWS, Anda menambahkan beberapa kontrol pada lapisan AWS Organizations struktur yang berbeda untuk membantu mengamankan sumber daya. Misalnya, defense-in-depth pendekatan mungkin menggabungkan otentikasi multi-faktor, segmentasi jaringan, dan enkripsi.

## administrator yang didelegasikan

Di AWS Organizations, layanan yang kompatibel dapat mendaftarkan akun AWS anggota untuk mengelola akun organisasi dan mengelola izin untuk layanan tersebut. Akun ini disebut administrator yang didelegasikan untuk layanan itu. Untuk informasi selengkapnya dan daftar layanan yang kompatibel, lihat [Layanan yang berfungsi dengan AWS Organizations](#) AWS Organizations dokumentasi.

## deployment

Proses pembuatan aplikasi, fitur baru, atau perbaikan kode tersedia di lingkungan target. Deployment melibatkan penerapan perubahan dalam basis kode dan kemudian membangun dan menjalankan basis kode itu di lingkungan aplikasi.

## lingkungan pengembangan

Lihat [lingkungan](#).

## kontrol detektif

Kontrol keamanan yang dirancang untuk mendeteksi, mencatat, dan memperingatkan setelah suatu peristiwa terjadi. Kontrol ini adalah garis pertahanan kedua, memperingatkan Anda tentang peristiwa keamanan yang melewati kontrol pencegahan yang ada. Untuk informasi selengkapnya, lihat Kontrol [Detektif dalam Menerapkan kontrol](#) keamanan pada. AWS

## pemetaan aliran nilai pengembangan (DVSM)

Sebuah proses yang digunakan untuk mengidentifikasi dan memprioritaskan kendala yang mempengaruhi kecepatan dan kualitas dalam siklus hidup pengembangan perangkat lunak. DVSM memperluas proses pemetaan aliran nilai yang awalnya dirancang untuk praktik

manufaktur ramping. Ini berfokus pada langkah-langkah dan tim yang diperlukan untuk menciptakan dan memindahkan nilai melalui proses pengembangan perangkat lunak.

## kembar digital

Representasi virtual dari sistem dunia nyata, seperti bangunan, pabrik, peralatan industri, atau jalur produksi. Kembar digital mendukung pemeliharaan prediktif, pemantauan jarak jauh, dan optimalisasi produksi.

## tabel dimensi

Dalam [skema bintang](#), tabel yang lebih kecil yang berisi atribut data tentang data kuantitatif dalam tabel fakta. Atribut tabel dimensi biasanya bidang teks atau angka diskrit yang berperilaku seperti teks. Atribut ini biasanya digunakan untuk pembatasan kueri, pemfilteran, dan pelabelan set hasil.

## musibah

Peristiwa yang mencegah beban kerja atau sistem memenuhi tujuan bisnisnya di lokasi utama yang digunakan. Peristiwa ini dapat berupa bencana alam, kegagalan teknis, atau akibat dari tindakan manusia, seperti kesalahan konfigurasi yang tidak disengaja atau serangan malware.

## pemulihan bencana (DR)

Strategi dan proses yang Anda gunakan untuk meminimalkan downtime dan kehilangan data yang disebabkan oleh [bencana](#). Untuk informasi selengkapnya, lihat [Disaster Recovery of Workloads on AWS: Recovery in the Cloud in the AWS Well-Architected Framework](#).

## DML~

Lihat [bahasa manipulasi basis data](#).

## desain berbasis domain

Pendekatan untuk mengembangkan sistem perangkat lunak yang kompleks dengan menghubungkan komponennya ke domain yang berkembang, atau tujuan bisnis inti, yang dilayani oleh setiap komponen. Konsep ini diperkenalkan oleh Eric Evans dalam bukunya, *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003). Untuk informasi tentang cara menggunakan desain berbasis domain dengan pola gambar pencekik, lihat Memodernisasi layanan web [Microsoft ASP.NET \(ASMX\) lama secara bertahap menggunakan container dan Amazon API Gateway](#).

## DR

Lihat [pemulihan bencana](#).

## deteksi drift

Melacak penyimpangan dari konfigurasi dasar. Misalnya, Anda dapat menggunakan AWS CloudFormation untuk [mendeteksi penyimpangan dalam sumber daya sistem](#), atau Anda dapat menggunakannya AWS Control Tower untuk [mendeteksi perubahan di landing zone](#) yang mungkin memengaruhi kepatuhan terhadap persyaratan tata kelola.

## DVSM

Lihat [pemetaan aliran nilai pengembangan](#).

## E

### EDA

Lihat [analisis data eksplorasi](#).

### EDI

Lihat [pertukaran data elektronik](#).

### komputasi tepi

Teknologi yang meningkatkan daya komputasi untuk perangkat pintar di tepi jaringan IoT. Jika dibandingkan dengan [komputasi awan](#), komputasi tepi dapat mengurangi latensi komunikasi dan meningkatkan waktu respons.

### pertukaran data elektronik (EDI)

Pertukaran otomatis dokumen bisnis antar organisasi. Untuk informasi selengkapnya, lihat [Apa itu Pertukaran Data Elektronik](#).

### enkripsi

Proses komputasi yang mengubah data plaintext, yang dapat dibaca manusia, menjadi ciphertext.

### kunci enkripsi

String kriptografi dari bit acak yang dihasilkan oleh algoritma enkripsi. Panjang kunci dapat bervariasi, dan setiap kunci dirancang agar tidak dapat diprediksi dan unik.

### endianness

Urutan byte disimpan dalam memori komputer. Sistem big-endian menyimpan byte paling signifikan terlebih dahulu. Sistem little-endian menyimpan byte paling tidak signifikan terlebih dahulu.

## titik akhir

Lihat [titik akhir layanan](#).

## layanan endpoint

Layanan yang dapat Anda host di cloud pribadi virtual (VPC) untuk dibagikan dengan pengguna lain. Anda dapat membuat layanan endpoint dengan AWS PrivateLink dan memberikan izin kepada prinsipal lain Akun AWS atau ke AWS Identity and Access Management (IAM). Akun atau prinsipal ini dapat terhubung ke layanan endpoint Anda secara pribadi dengan membuat titik akhir VPC antarmuka. Untuk informasi selengkapnya, lihat [Membuat layanan titik akhir](#) di dokumentasi Amazon Virtual Private Cloud (Amazon VPC).

## perencanaan sumber daya perusahaan (ERP)

Sistem yang mengotomatiskan dan mengelola proses bisnis utama (seperti akuntansi, [MES](#), dan manajemen proyek) untuk suatu perusahaan.

## enkripsi amplop

Proses mengenkripsi kunci enkripsi dengan kunci enkripsi lain. Untuk informasi selengkapnya, lihat [Enkripsi amplop](#) dalam dokumentasi AWS Key Management Service (AWS KMS).

## lingkungan

Sebuah contoh dari aplikasi yang sedang berjalan. Berikut ini adalah jenis lingkungan yang umum dalam komputasi awan:

- **Development Environment** — Sebuah contoh dari aplikasi yang berjalan yang hanya tersedia untuk tim inti yang bertanggung jawab untuk memelihara aplikasi. Lingkungan pengembangan digunakan untuk menguji perubahan sebelum mempromosikannya ke lingkungan atas. Jenis lingkungan ini kadang-kadang disebut sebagai lingkungan pengujian.
- **lingkungan yang lebih rendah** — Semua lingkungan pengembangan untuk aplikasi, seperti yang digunakan untuk build awal dan pengujian.
- **lingkungan produksi** — Sebuah contoh dari aplikasi yang berjalan yang pengguna akhir dapat mengakses. Dalam sebuah CI/CD pipeline, lingkungan produksi adalah lingkungan penyebaran terakhir.
- **lingkungan atas** — Semua lingkungan yang dapat diakses oleh pengguna selain tim pengembangan inti. Ini dapat mencakup lingkungan produksi, lingkungan praproduksi, dan lingkungan untuk pengujian penerimaan pengguna.

## epik

Dalam metodologi tangkas, kategori fungsional yang membantu mengatur dan memprioritaskan pekerjaan Anda. Epik memberikan deskripsi tingkat tinggi tentang persyaratan dan tugas implementasi. Misalnya, epos keamanan AWS CAF mencakup manajemen identitas dan akses, kontrol detektif, keamanan infrastruktur, perlindungan data, dan respons insiden. Untuk informasi selengkapnya tentang epos dalam strategi AWS migrasi, lihat [panduan implementasi program](#).

## ERP

Lihat [perencanaan sumber daya perusahaan](#).

## analisis data eksplorasi (EDA)

Proses menganalisis dataset untuk memahami karakteristik utamanya. Anda mengumpulkan atau mengumpulkan data dan kemudian melakukan penyelidikan awal untuk menemukan pola, mendeteksi anomali, dan memeriksa asumsi. EDA dilakukan dengan menghitung statistik ringkasan dan membuat visualisasi data.

## F

### tabel fakta

Tabel tengah dalam [skema bintang](#). Ini menyimpan data kuantitatif tentang operasi bisnis. Biasanya, tabel fakta berisi dua jenis kolom: kolom yang berisi ukuran dan yang berisi kunci asing ke tabel dimensi.

### gagal cepat

Filosofi yang menggunakan pengujian yang sering dan bertahap untuk mengurangi siklus hidup pengembangan. Ini adalah bagian penting dari pendekatan tangkas.

### batas isolasi kesalahan

Dalam AWS Cloud, batas seperti Availability Zone, Wilayah AWS, control plane, atau data plane yang membatasi efek kegagalan dan membantu meningkatkan ketahanan beban kerja. Untuk informasi selengkapnya, lihat [Batas Isolasi AWS Kesalahan](#).

### cabang fitur

Lihat [cabang](#).

## fitur

Data input yang Anda gunakan untuk membuat prediksi. Misalnya, dalam konteks manufaktur, fitur bisa berupa gambar yang diambil secara berkala dari lini manufaktur.

## pentingnya fitur

Seberapa signifikan fitur untuk prediksi model. Ini biasanya dinyatakan sebagai skor numerik yang dapat dihitung melalui berbagai teknik, seperti Shapley Additive Explanations (SHAP) dan gradien terintegrasi. Untuk informasi lebih lanjut, lihat [Interpretabilitas model pembelajaran mesin](#) dengan AWS

## transformasi fitur

Untuk mengoptimalkan data untuk proses ML, termasuk memperkaya data dengan sumber tambahan, menskalakan nilai, atau mengekstrak beberapa set informasi dari satu bidang data. Hal ini memungkinkan model ML untuk mendapatkan keuntungan dari data. Misalnya, jika Anda memecah tanggal "2021-05-27 00:15:37" menjadi "2021", "Mei", "Kamis", dan "15", Anda dapat membantu algoritme pembelajaran mempelajari pola bernuansa yang terkait dengan komponen data yang berbeda.

## beberapa tembakan mendorong

Menyediakan [LLM](#) dengan sejumlah kecil contoh yang menunjukkan tugas dan output yang diinginkan sebelum memintanya untuk melakukan tugas serupa. Teknik ini adalah aplikasi pembelajaran dalam konteks, di mana model belajar dari contoh (bidikan) yang tertanam dalam petunjuk. Beberapa bidikan dapat efektif untuk tugas-tugas yang memerlukan pemformatan, penalaran, atau pengetahuan domain tertentu. Lihat juga [zero-shot](#) prompting.

## FGAC

Lihat kontrol [akses berbutir halus](#).

## kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

## migrasi flash-cut

Metode migrasi database yang menggunakan replikasi data berkelanjutan melalui [pengambilan data perubahan](#) untuk memigrasikan data dalam waktu sesingkat mungkin, alih-alih menggunakan pendekatan bertahap. Tujuannya adalah untuk menjaga downtime seminimal mungkin.

## FM

Lihat [model pondasi](#).

### model pondasi (FM)

Jaringan saraf pembelajaran mendalam yang besar yang telah melatih kumpulan data besar-besaran data umum dan tidak berlabel. FMs mampu melakukan berbagai tugas umum, seperti memahami bahasa, menghasilkan teks dan gambar, dan berbicara dalam bahasa alami. Untuk informasi selengkapnya, lihat [Apa itu Model Foundation](#).

## G

### AI generatif

Subset model [AI](#) yang telah dilatih pada sejumlah besar data dan yang dapat menggunakan prompt teks sederhana untuk membuat konten dan artefak baru, seperti gambar, video, teks, dan audio. Untuk informasi lebih lanjut, lihat [Apa itu AI Generatif](#).

### pemblokiran geografis

Lihat [pembatasan geografis](#).

### pembatasan geografis (pemblokiran geografis)

Di Amazon CloudFront, opsi untuk mencegah pengguna di negara tertentu mengakses distribusi konten. Anda dapat menggunakan daftar izinkan atau daftar blokir untuk menentukan negara yang disetujui dan dilarang. Untuk informasi selengkapnya, lihat [Membatasi distribusi geografis konten Anda](#) dalam dokumentasi. CloudFront

### Alur kerja Gitflow

Pendekatan di mana lingkungan bawah dan atas menggunakan cabang yang berbeda dalam repositori kode sumber. Alur kerja Gitflow dianggap warisan, dan [alur kerja berbasis batang](#) adalah pendekatan modern yang lebih disukai.

### gambar emas

Sebuah snapshot dari sistem atau perangkat lunak yang digunakan sebagai template untuk menyebarkan instance baru dari sistem atau perangkat lunak itu. Misalnya, di bidang manufaktur, gambar emas dapat digunakan untuk menyediakan perangkat lunak pada beberapa perangkat dan membantu meningkatkan kecepatan, skalabilitas, dan produktivitas dalam operasi manufaktur perangkat.

## strategi greenfield

Tidak adanya infrastruktur yang ada di lingkungan baru. [Saat mengadopsi strategi greenfield untuk arsitektur sistem, Anda dapat memilih semua teknologi baru tanpa batasan kompatibilitas dengan infrastruktur yang ada, juga dikenal sebagai brownfield.](#) Jika Anda memperluas infrastruktur yang ada, Anda dapat memadukan strategi brownfield dan greenfield.

## pagar pembatas

Aturan tingkat tinggi yang membantu mengatur sumber daya, kebijakan, dan kepatuhan di seluruh unit organisasi (OU). Pagar pembatas preventif menegakkan kebijakan untuk memastikan keselarasan dengan standar kepatuhan. Mereka diimplementasikan dengan menggunakan kebijakan kontrol layanan dan batas izin IAM. Detective guardrails mendeteksi pelanggaran kebijakan dan masalah kepatuhan, dan menghasilkan peringatan untuk remediasi. Mereka diimplementasikan dengan menggunakan AWS Config, AWS Security Hub CSPM, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector, dan pemeriksaan khusus AWS Lambda .

# H

## HA

Lihat [ketersediaan tinggi](#).

## migrasi database heterogen

Memigrasi database sumber Anda ke database target yang menggunakan mesin database yang berbeda (misalnya, Oracle ke Amazon Aurora). Migrasi heterogen biasanya merupakan bagian dari upaya arsitektur ulang, dan mengubah skema dapat menjadi tugas yang kompleks. [AWS menyediakan AWS SCT](#) yang membantu dengan konversi skema.

## ketersediaan tinggi (HA)

Kemampuan beban kerja untuk beroperasi terus menerus, tanpa intervensi, jika terjadi tantangan atau bencana. Sistem HA dirancang untuk gagal secara otomatis, secara konsisten memberikan kinerja berkualitas tinggi, dan menangani beban dan kegagalan yang berbeda dengan dampak kinerja minimal.

## modernisasi sejarawan

Pendekatan yang digunakan untuk memodernisasi dan meningkatkan sistem teknologi operasional (OT) untuk melayani kebutuhan industri manufaktur dengan lebih baik. Sejarawan

adalah jenis database yang digunakan untuk mengumpulkan dan menyimpan data dari berbagai sumber di pabrik.

#### data penahanan

Sebagian dari data historis berlabel yang ditahan dari kumpulan data yang digunakan untuk melatih model pembelajaran [mesin](#). Anda dapat menggunakan data penahanan untuk mengevaluasi kinerja model dengan membandingkan prediksi model dengan data penahanan.

#### migrasi database homogen

Memigrasi database sumber Anda ke database target yang berbagi mesin database yang sama (misalnya, Microsoft SQL Server ke Amazon RDS for SQL Server). Migrasi homogen biasanya merupakan bagian dari upaya rehosting atau replatforming. Anda dapat menggunakan utilitas database asli untuk memigrasi skema.

#### data panas

Data yang sering diakses, seperti data real-time atau data translasi terbaru. Data ini biasanya memerlukan tingkat atau kelas penyimpanan berkinerja tinggi untuk memberikan respons kueri yang cepat.

#### perbaikan terbaru

Perbaikan mendesak untuk masalah kritis dalam lingkungan produksi. Karena urgensinya, perbaikan terbaru biasanya dibuat di luar alur kerja DevOps rilis biasa.

#### periode hypercare

Segera setelah cutover, periode waktu ketika tim migrasi mengelola dan memantau aplikasi yang dimigrasi di cloud untuk mengatasi masalah apa pun. Biasanya, periode ini panjangnya 1-4 hari. Pada akhir periode hypercare, tim migrasi biasanya mentransfer tanggung jawab untuk aplikasi ke tim operasi cloud.

|

#### IAC

Lihat [infrastruktur sebagai kode](#).

#### kebijakan berbasis identitas

Kebijakan yang dilampirkan pada satu atau beberapa prinsip IAM yang mendefinisikan izin mereka dalam lingkungan. AWS Cloud

|

## aplikasi idle

Aplikasi yang memiliki penggunaan CPU dan memori rata-rata antara 5 dan 20 persen selama periode 90 hari. Dalam proyek migrasi, adalah umum untuk menghentikan aplikasi ini atau mempertahankannya di tempat.

## IloT

Lihat [Internet of Things industri](#).

## infrastruktur yang tidak dapat diubah

Model yang menyebarkan infrastruktur baru untuk beban kerja produksi alih-alih memperbarui, menambal, atau memodifikasi infrastruktur yang ada. [Infrastruktur yang tidak dapat diubah secara inheren lebih konsisten, andal, dan dapat diprediksi daripada infrastruktur yang dapat berubah](#). Untuk informasi selengkapnya, lihat praktik terbaik [Deploy using immutable infrastructure](#) di AWS Well-Architected Framework.

## masuk (masuknya) VPC

Dalam arsitektur AWS multi-akun, VPC yang menerima, memeriksa, dan merutekan koneksi jaringan dari luar aplikasi. [Arsitektur Referensi AWS Keamanan](#) merekomendasikan pengaturan akun Jaringan Anda dengan inbound, outbound, dan inspeksi VPCs untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

## migrasi inkremental

Strategi cutover di mana Anda memigrasikan aplikasi Anda dalam bagian-bagian kecil alih-alih melakukan satu cutover penuh. Misalnya, Anda mungkin hanya memindahkan beberapa layanan mikro atau pengguna ke sistem baru pada awalnya. Setelah Anda memverifikasi bahwa semuanya berfungsi dengan baik, Anda dapat secara bertahap memindahkan layanan mikro atau pengguna tambahan hingga Anda dapat menonaktifkan sistem lama Anda. Strategi ini mengurangi risiko yang terkait dengan migrasi besar.

## Industri 4.0

Sebuah istilah yang diperkenalkan oleh [Klaus Schwab](#) pada tahun 2016 untuk merujuk pada modernisasi proses manufaktur melalui kemajuan dalam konektivitas, data real-time, otomatisasi, analitik, dan AI/ML.

## infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

## infrastruktur sebagai kode (IAC)

Proses penyediaan dan pengelolaan infrastruktur aplikasi melalui satu set file konfigurasi. IAC dirancang untuk membantu Anda memusatkan manajemen infrastruktur, menstandarisasi sumber daya, dan menskalakan dengan cepat sehingga lingkungan baru dapat diulang, andal, dan konsisten.

## Internet of Things industri (IIoT)

Penggunaan sensor dan perangkat yang terhubung ke internet di sektor industri, seperti manufaktur, energi, otomotif, perawatan kesehatan, ilmu kehidupan, dan pertanian. Untuk informasi lebih lanjut, lihat [Membangun strategi transformasi digital Internet of Things \(IIoT\) industri](#).

## inspeksi VPC

Dalam arsitektur AWS multi-akun, VPC terpusat yang mengelola inspeksi lalu lintas jaringan antara VPCs (dalam yang sama atau berbeda Wilayah AWS), internet, dan jaringan lokal. [Arsitektur Referensi AWS Keamanan](#) merekomendasikan pengaturan akun Jaringan Anda dengan inbound, outbound, dan inspeksi VPCs untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

## Internet of Things (IoT)

Jaringan objek fisik yang terhubung dengan sensor atau prosesor tertanam yang berkomunikasi dengan perangkat dan sistem lain melalui internet atau melalui jaringan komunikasi lokal. Untuk informasi selengkapnya, lihat [Apa itu IoT?](#)

## interpretasi

Karakteristik model pembelajaran mesin yang menggambarkan sejauh mana manusia dapat memahami bagaimana prediksi model bergantung pada inputnya. Untuk informasi lebih lanjut, lihat [Interpretabilitas model pembelajaran mesin](#) dengan AWS

## IoT

Lihat [Internet of Things](#).

## Perpustakaan informasi TI (ITIL)

Serangkaian praktik terbaik untuk memberikan layanan TI dan menyelaraskan layanan ini dengan persyaratan bisnis. ITIL menyediakan dasar untuk ITSM.

## Manajemen layanan TI (ITSM)

Kegiatan yang terkait dengan merancang, menerapkan, mengelola, dan mendukung layanan TI untuk suatu organisasi. Untuk informasi tentang mengintegrasikan operasi cloud dengan alat ITSM, lihat panduan [integrasi operasi](#).

### ITIL

Lihat [perpustakaan informasi TI](#).

### ITSM

Lihat [manajemen layanan TI](#).

## L

### kontrol akses berbasis label (LBAC)

Implementasi kontrol akses wajib (MAC) di mana pengguna dan data itu sendiri masing-masing secara eksplisit diberi nilai label keamanan. Persimpangan antara label keamanan pengguna dan label keamanan data menentukan baris dan kolom mana yang dapat dilihat oleh pengguna.

### landing zone

Landing zone adalah AWS lingkungan multi-akun yang dirancang dengan baik yang dapat diskalakan dan aman. Ini adalah titik awal dari mana organisasi Anda dapat dengan cepat meluncurkan dan menyebarkan beban kerja dan aplikasi dengan percaya diri dalam lingkungan keamanan dan infrastruktur mereka. Untuk informasi selengkapnya tentang zona pendaratan, lihat [Menyiapkan lingkungan multi-akun AWS yang aman dan dapat diskalakan](#).

### model bahasa besar (LLM)

Model [AI](#) pembelajaran mendalam yang dilatih sebelumnya pada sejumlah besar data. LLM dapat melakukan beberapa tugas, seperti menjawab pertanyaan, meringkas dokumen, menerjemahkan teks ke dalam bahasa lain, dan menyelesaikan kalimat. Untuk informasi lebih lanjut, lihat [Apa itu LLMs](#).

### migrasi besar

Migrasi 300 atau lebih server.

### LBAC

Lihat [kontrol akses berbasis label](#).

## hak istimewa paling sedikit

Praktik keamanan terbaik untuk memberikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat [Menerapkan izin hak istimewa terkecil dalam dokumentasi IAM](#).

## angkat dan geser

Lihat [7 Rs](#).

## sistem endian kecil

Sebuah sistem yang menyimpan byte paling tidak signifikan terlebih dahulu. Lihat juga [endianness](#).

## LLM

Lihat [model bahasa besar](#).

## lingkungan yang lebih rendah

Lihat [lingkungan](#).

# M

## pembelajaran mesin (ML)

Jenis kecerdasan buatan yang menggunakan algoritma dan teknik untuk pengenalan pola dan pembelajaran. ML menganalisis dan belajar dari data yang direkam, seperti data Internet of Things (IoT), untuk menghasilkan model statistik berdasarkan pola. Untuk informasi selengkapnya, lihat [Machine Learning](#).

## cabang utama

Lihat [cabang](#).

## malware

Perangkat lunak yang dirancang untuk membahayakan keamanan atau privasi komputer. Malware dapat mengganggu sistem komputer, membocorkan informasi sensitif, atau mendapatkan akses yang tidak sah. Contoh malware termasuk virus, worm, ransomware, Trojan horse, spyware, dan keyloggers.

## layanan terkelola

Layanan AWS yang AWS mengoperasikan lapisan infrastruktur, sistem operasi, dan platform, dan Anda mengakses titik akhir untuk menyimpan dan mengambil data. Amazon Simple Storage Service (Amazon S3) dan Amazon DynamoDB adalah contoh layanan terkelola. Ini juga dikenal sebagai layanan abstrak.

## sistem eksekusi manufaktur (MES)

Sistem perangkat lunak untuk melacak, memantau, mendokumentasikan, dan mengendalikan proses produksi yang mengubah bahan baku menjadi produk jadi di lantai toko.

## PETA

Lihat [Program Percepatan Migrasi](#).

## mekanisme

Proses lengkap di mana Anda membuat alat, mendorong adopsi alat, dan kemudian memeriksa hasilnya untuk melakukan penyesuaian. Mekanisme adalah siklus yang memperkuat dan meningkatkan dirinya sendiri saat beroperasi. Untuk informasi lebih lanjut, lihat [Membangun mekanisme](#) di AWS Well-Architected Framework.

## akun anggota

Semua Akun AWS selain akun manajemen yang merupakan bagian dari organisasi di AWS Organizations. Akun dapat menjadi anggota dari hanya satu organisasi pada suatu waktu.

## MES

Lihat [sistem eksekusi manufaktur](#).

## Transportasi Telemetri Antrian Pesan (MQTT)

[Protokol komunikasi ringan machine-to-machine \(M2M\), berdasarkan pola terbitkan/berlangganan, untuk perangkat IoT yang dibatasi sumber daya.](#)

## layanan mikro

Layanan kecil dan independen yang berkomunikasi dengan jelas APIs dan biasanya dimiliki oleh tim kecil yang mandiri. Misalnya, sistem asuransi mungkin mencakup layanan mikro yang memetakan kemampuan bisnis, seperti penjualan atau pemasaran, atau subdomain, seperti pembelian, klaim, atau analitik. Manfaat layanan mikro termasuk kelincahan, penskalaan yang fleksibel, penyebaran yang mudah, kode yang dapat digunakan kembali, dan ketahanan. Untuk

informasi selengkapnya, lihat [Mengintegrasikan layanan mikro dengan menggunakan layanan tanpa AWS server](#).

## arsitektur microservices

Pendekatan untuk membangun aplikasi dengan komponen independen yang menjalankan setiap proses aplikasi sebagai layanan mikro. Layanan mikro ini berkomunikasi melalui antarmuka yang terdefinisi dengan baik dengan menggunakan ringan. APIs Setiap layanan mikro dalam arsitektur ini dapat diperbarui, digunakan, dan diskalakan untuk memenuhi permintaan fungsi tertentu dari suatu aplikasi. Untuk informasi selengkapnya, lihat [Menerapkan layanan mikro di AWS](#).

## Program Percepatan Migrasi (MAP)

AWS Program yang menyediakan dukungan konsultasi, pelatihan, dan layanan untuk membantu organisasi membangun fondasi operasional yang kuat untuk pindah ke cloud, dan untuk membantu mengimbangi biaya awal migrasi. MAP mencakup metodologi migrasi untuk mengeksekusi migrasi lama dengan cara metodis dan seperangkat alat untuk mengotomatisasi dan mempercepat skenario migrasi umum.

## migrasi dalam skala

Proses memindahkan sebagian besar portofolio aplikasi ke cloud dalam gelombang, dengan lebih banyak aplikasi bergerak pada tingkat yang lebih cepat di setiap gelombang. Fase ini menggunakan praktik dan pelajaran terbaik dari fase sebelumnya untuk mengimplementasikan pabrik migrasi tim, alat, dan proses untuk merampingkan migrasi beban kerja melalui otomatisasi dan pengiriman tangkas. Ini adalah fase ketiga dari [strategi AWS migrasi](#).

## pabrik migrasi

Tim lintas fungsi yang merampingkan migrasi beban kerja melalui pendekatan otomatis dan gesit. Tim pabrik migrasi biasanya mencakup operasi, analis dan pemilik bisnis, insinyur migrasi, pengembang, dan DevOps profesional yang bekerja di sprint. Antara 20 dan 50 persen portofolio aplikasi perusahaan terdiri dari pola berulang yang dapat dioptimalkan dengan pendekatan pabrik. Untuk informasi selengkapnya, lihat [diskusi tentang pabrik migrasi](#) dan [panduan Pabrik Migrasi Cloud](#) di kumpulan konten ini.

## metadata migrasi

Informasi tentang aplikasi dan server yang diperlukan untuk menyelesaikan migrasi. Setiap pola migrasi memerlukan satu set metadata migrasi yang berbeda. Contoh metadata migrasi termasuk subnet target, grup keamanan, dan akun. AWS

## pola migrasi

Tugas migrasi berulang yang merinci strategi migrasi, tujuan migrasi, dan aplikasi atau layanan migrasi yang digunakan. Contoh: Rehost migrasi ke Amazon EC2 AWS dengan Layanan Migrasi Aplikasi.

## Penilaian Portofolio Migrasi (MPA)

Alat online yang menyediakan informasi untuk memvalidasi kasus bisnis untuk bermigrasi ke. AWS Cloud MPA menyediakan penilaian portofolio terperinci (ukuran kanan server, harga, perbandingan TCO, analisis biaya migrasi) serta perencanaan migrasi (analisis data aplikasi dan pengumpulan data, pengelompokan aplikasi, prioritas migrasi, dan perencanaan gelombang). [Alat MPA](#) (memerlukan login) tersedia gratis untuk semua AWS konsultan dan konsultan APN Partner.

## Penilaian Kesiapan Migrasi (MRA)

Proses mendapatkan wawasan tentang status kesiapan cloud organisasi, mengidentifikasi kekuatan dan kelemahan, dan membangun rencana aksi untuk menutup kesenjangan yang diidentifikasi, menggunakan CAF. AWS Untuk informasi selengkapnya, lihat [panduan kesiapan migrasi](#). MRA adalah tahap pertama dari [strategi AWS migrasi](#).

## strategi migrasi

Pendekatan yang digunakan untuk memigrasikan beban kerja ke file. AWS Cloud Untuk informasi lebih lanjut, lihat entri [7 Rs](#) di glosarium ini dan lihat [Memobilisasi organisasi Anda untuk mempercepat](#) migrasi skala besar.

## ML

Lihat [pembelajaran mesin](#).

## modernisasi

Mengubah aplikasi usang (warisan atau monolitik) dan infrastrukturnya menjadi sistem yang gesit, elastis, dan sangat tersedia di cloud untuk mengurangi biaya, mendapatkan efisiensi, dan memanfaatkan inovasi. Untuk informasi selengkapnya, lihat [Strategi untuk memodernisasi aplikasi di](#). AWS Cloud

## penilaian kesiapan modernisasi

Evaluasi yang membantu menentukan kesiapan modernisasi aplikasi organisasi; mengidentifikasi manfaat, risiko, dan dependensi; dan menentukan seberapa baik organisasi dapat mendukung keadaan masa depan aplikasi tersebut. Hasil penilaian adalah cetak biru arsitektur target, peta

jalan yang merinci fase pengembangan dan tonggak untuk proses modernisasi, dan rencana aksi untuk mengatasi kesenjangan yang diidentifikasi. Untuk informasi lebih lanjut, lihat [Mengevaluasi kesiapan modernisasi untuk](#) aplikasi di. AWS Cloud

aplikasi monolitik (monolit)

Aplikasi yang berjalan sebagai layanan tunggal dengan proses yang digabungkan secara ketat. Aplikasi monolitik memiliki beberapa kelemahan. Jika satu fitur aplikasi mengalami lonjakan permintaan, seluruh arsitektur harus diskalakan. Menambahkan atau meningkatkan fitur aplikasi monolitik juga menjadi lebih kompleks ketika basis kode tumbuh. Untuk mengatasi masalah ini, Anda dapat menggunakan arsitektur microservices. Untuk informasi lebih lanjut, lihat [Mengurai monolit](#) menjadi layanan mikro.

MPA

Lihat [Penilaian Portofolio Migrasi](#).

MQTT

Lihat [Transportasi Telemetri Antrian Pesan](#).

klasifikasi multiclass

Sebuah proses yang membantu menghasilkan prediksi untuk beberapa kelas (memprediksi satu dari lebih dari dua hasil). Misalnya, model ML mungkin bertanya “Apakah produk ini buku, mobil, atau telepon?” atau “Kategori produk mana yang paling menarik bagi pelanggan ini?”

infrastruktur yang bisa berubah

Model yang memperbarui dan memodifikasi infrastruktur yang ada untuk beban kerja produksi. Untuk meningkatkan konsistensi, keandalan, dan prediktabilitas, AWS Well-Architected Framework merekomendasikan penggunaan infrastruktur yang [tidak](#) dapat diubah sebagai praktik terbaik.

O

OAC

Lihat [kontrol akses asal](#).

OAI

Lihat [identitas akses asal](#).

## OCM

Lihat [manajemen perubahan organisasi](#).

### migrasi offline

Metode migrasi di mana beban kerja sumber diturunkan selama proses migrasi. Metode ini melibatkan waktu henti yang diperpanjang dan biasanya digunakan untuk beban kerja kecil dan tidak kritis.

## OI

Lihat [integrasi operasi](#).

## OLA

Lihat [perjanjian tingkat operasional](#).

### migrasi online

Metode migrasi di mana beban kerja sumber disalin ke sistem target tanpa diambil offline. Aplikasi yang terhubung ke beban kerja dapat terus berfungsi selama migrasi. Metode ini melibatkan waktu henti nol hingga minimal dan biasanya digunakan untuk beban kerja produksi yang kritis.

## OPC-UA

Lihat [Komunikasi Proses Terbuka - Arsitektur Terpadu](#).

### Komunikasi Proses Terbuka - Arsitektur Terpadu (OPC-UA)

Protokol komunikasi machine-to-machine (M2M) untuk otomasi industri. OPC-UA menyediakan standar interoperabilitas dengan enkripsi data, otentikasi, dan skema otorisasi.

### perjanjian tingkat operasional (OLA)

Perjanjian yang menjelaskan apa yang dijanjikan kelompok TI fungsional untuk diberikan satu sama lain, untuk mendukung perjanjian tingkat layanan (SLA).

### Tinjauan Kesiapan Operasional (ORR)

Daftar pertanyaan dan praktik terbaik terkait yang membantu Anda memahami, mengevaluasi, mencegah, atau mengurangi ruang lingkup insiden dan kemungkinan kegagalan. Untuk informasi lebih lanjut, lihat [Ulasan Kesiapan Operasional \(ORR\)](#) dalam Kerangka Kerja Well-Architected AWS .

## teknologi operasional (OT)

Sistem perangkat keras dan perangkat lunak yang bekerja dengan lingkungan fisik untuk mengendalikan operasi industri, peralatan, dan infrastruktur. Di bidang manufaktur, integrasi sistem OT dan teknologi informasi (TI) adalah fokus utama untuk transformasi [Industri 4.0](#).

## integrasi operasi (OI)

Proses modernisasi operasi di cloud, yang melibatkan perencanaan kesiapan, otomatisasi, dan integrasi. Untuk informasi selengkapnya, lihat [panduan integrasi operasi](#).

## jejak organisasi

Jejak yang dibuat oleh AWS CloudTrail itu mencatat semua peristiwa untuk semua Akun AWS dalam organisasi di AWS Organizations. Jejak ini dibuat di setiap Akun AWS bagian organisasi dan melacak aktivitas di setiap akun. Untuk informasi selengkapnya, lihat [Membuat jejak untuk organisasi](#) dalam CloudTrail dokumentasi.

## manajemen perubahan organisasi (OCM)

Kerangka kerja untuk mengelola transformasi bisnis utama yang mengganggu dari perspektif orang, budaya, dan kepemimpinan. OCM membantu organisasi mempersiapkan, dan transisi ke, sistem dan strategi baru dengan mempercepat adopsi perubahan, mengatasi masalah transisi, dan mendorong perubahan budaya dan organisasi. Dalam strategi AWS migrasi, kerangka kerja ini disebut percepatan orang, karena kecepatan perubahan yang diperlukan dalam proyek adopsi cloud. Untuk informasi lebih lanjut, lihat [panduan OCM](#).

## kontrol akses asal (OAC)

Di CloudFront, opsi yang disempurnakan untuk membatasi akses untuk mengamankan konten Amazon Simple Storage Service (Amazon S3) Anda. OAC mendukung semua bucket S3 di semua Wilayah AWS, enkripsi sisi server dengan AWS KMS (SSE-KMS), dan dinamis dan permintaan ke bucket S3. PUT DELETE

## identitas akses asal (OAI)

Di CloudFront, opsi untuk membatasi akses untuk mengamankan konten Amazon S3 Anda. Saat Anda menggunakan OAI, CloudFront buat prinsipal yang dapat diautentikasi oleh Amazon S3. Prinsipal yang diautentikasi dapat mengakses konten dalam bucket S3 hanya melalui distribusi tertentu. CloudFront Lihat juga [OAC](#), yang menyediakan kontrol akses yang lebih terperinci dan ditingkatkan.

## ORR

Lihat [tinjauan kesiapan operasional](#).

## OT

Lihat [teknologi operasional](#).

### keluar (jalan keluar) VPC

Dalam arsitektur AWS multi-akun, VPC yang menangani koneksi jaringan yang dimulai dari dalam aplikasi. [Arsitektur Referensi AWS Keamanan](#) merekomendasikan pengaturan akun Jaringan Anda dengan inbound, outbound, dan inspeksi VPCs untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

## P

### batas izin

Kebijakan manajemen IAM yang dilampirkan pada prinsipal IAM untuk menetapkan izin maksimum yang dapat dimiliki pengguna atau peran. Untuk informasi selengkapnya, lihat [Batas izin](#) dalam dokumentasi IAM.

### Informasi Identifikasi Pribadi (PII)

Informasi yang, jika dilihat secara langsung atau dipasangkan dengan data terkait lainnya, dapat digunakan untuk menyimpulkan identitas individu secara wajar. Contoh PII termasuk nama, alamat, dan informasi kontak.

### PII

Lihat informasi yang [dapat diidentifikasi secara pribadi](#).

### buku pedoman

Serangkaian langkah yang telah ditentukan sebelumnya yang menangkap pekerjaan yang terkait dengan migrasi, seperti mengirimkan fungsi operasi inti di cloud. Buku pedoman dapat berupa skrip, runbook otomatis, atau ringkasan proses atau langkah-langkah yang diperlukan untuk mengoperasikan lingkungan modern Anda.

### PLC

Lihat [pengontrol logika yang dapat diprogram](#).

### PLM

Lihat [manajemen siklus hidup produk](#).

## kebijakan

Objek yang dapat menentukan izin (lihat kebijakan berbasis identitas), menentukan kondisi akses (lihat kebijakan berbasis sumber daya), atau menentukan izin maksimum untuk semua akun di organisasi (lihat kebijakan kontrol layanan). [AWS Organizations](#)

## persistensi poliglot

Secara independen memilih teknologi penyimpanan data microservice berdasarkan pola akses data dan persyaratan lainnya. Jika layanan mikro Anda memiliki teknologi penyimpanan data yang sama, mereka dapat menghadapi tantangan implementasi atau mengalami kinerja yang buruk. Layanan mikro lebih mudah diimplementasikan dan mencapai kinerja dan skalabilitas yang lebih baik jika mereka menggunakan penyimpanan data yang paling sesuai dengan kebutuhan mereka.

## penilaian portofolio

Proses menemukan, menganalisis, dan memprioritaskan portofolio aplikasi untuk merencanakan migrasi. Untuk informasi selengkapnya, lihat [Mengevaluasi kesiapan migrasi](#).

## predikat

Kondisi kueri yang mengembalikan `true` atau `false`, biasanya terletak di `WHERE` klausa.

## predikat pushdown

Teknik optimasi kueri database yang menyaring data dalam kueri sebelum transfer. Ini mengurangi jumlah data yang harus diambil dan diproses dari database relasional, dan meningkatkan kinerja kueri.

## kontrol preventif

Kontrol keamanan yang dirancang untuk mencegah suatu peristiwa terjadi. Kontrol ini adalah garis pertahanan pertama untuk membantu mencegah akses tidak sah atau perubahan yang tidak diinginkan ke jaringan Anda. Untuk informasi selengkapnya, lihat [Kontrol pencegahan dalam Menerapkan kontrol](#) keamanan pada. AWS

## principal

Entitas AWS yang dapat melakukan tindakan dan mengakses sumber daya. Entitas ini biasanya merupakan pengguna root untuk Akun AWS, peran IAM, atau pengguna. Untuk informasi selengkapnya, lihat Prinsip dalam [istilah dan konsep Peran](#) dalam dokumentasi IAM.

## privasi berdasarkan desain

Pendekatan rekayasa sistem yang memperhitungkan privasi melalui seluruh proses pengembangan.

## zona host pribadi

Container yang menyimpan informasi tentang bagaimana Anda ingin Amazon Route 53 merespons kueri DNS untuk domain dan subdomainnya dalam satu atau lebih VPCs. Untuk informasi selengkapnya, lihat [Bekerja dengan zona yang dihosting pribadi](#) di dokumentasi Route 53.

## kontrol proaktif

[Kontrol keamanan](#) yang dirancang untuk mencegah penyebaran sumber daya yang tidak sesuai. Kontrol ini memindai sumber daya sebelum disediakan. Jika sumber daya tidak sesuai dengan kontrol, maka itu tidak disediakan. Untuk informasi selengkapnya, lihat [panduan referensi Kontrol](#) dalam AWS Control Tower dokumentasi dan lihat [Kontrol proaktif](#) dalam Menerapkan kontrol keamanan pada AWS.

## manajemen siklus hidup produk (PLM)

Manajemen data dan proses untuk suatu produk di seluruh siklus hidupnya, mulai dari desain, pengembangan, dan peluncuran, melalui pertumbuhan dan kematangan, hingga penurunan dan penghapusan.

## lingkungan produksi

Lihat [lingkungan](#).

## pengontrol logika yang dapat diprogram (PLC)

Di bidang manufaktur, komputer yang sangat andal dan mudah beradaptasi yang memantau mesin dan mengotomatiskan proses manufaktur.

## rantai cepat

Menggunakan output dari satu prompt [LLM](#) sebagai input untuk prompt berikutnya untuk menghasilkan respons yang lebih baik. Teknik ini digunakan untuk memecah tugas yang kompleks menjadi subtugas, atau untuk secara iteratif memperbaiki atau memperluas respons awal. Ini membantu meningkatkan akurasi dan relevansi respons model dan memungkinkan hasil yang lebih terperinci dan dipersonalisasi.

## pseudonimisasi

Proses penggantian pengidentifikasi pribadi dalam kumpulan data dengan nilai placeholder. Pseudonimisasi dapat membantu melindungi privasi pribadi. Data pseudonim masih dianggap sebagai data pribadi.

publish/subscribe (pub/sub)

Pola yang memungkinkan komunikasi asinkron antara layanan mikro untuk meningkatkan skalabilitas dan daya tanggap. Misalnya, dalam [MES](#) berbasis layanan mikro, layanan mikro dapat mempublikasikan pesan peristiwa ke saluran yang dapat berlangganan layanan mikro lainnya. Sistem dapat menambahkan layanan mikro baru tanpa mengubah layanan penerbitan.

## Q

rencana kueri

Serangkaian langkah, seperti instruksi, yang digunakan untuk mengakses data dalam sistem database relasional SQL.

regresi rencana kueri

Ketika pengoptimal layanan database memilih rencana yang kurang optimal daripada sebelum perubahan yang diberikan ke lingkungan database. Hal ini dapat disebabkan oleh perubahan statistik, kendala, pengaturan lingkungan, pengikatan parameter kueri, dan pembaruan ke mesin database.

## R

Matriks RACI

Lihat [bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan \(RACI\)](#).

LAP

Lihat [Retrieval Augmented Generation](#).

ransomware

Perangkat lunak berbahaya yang dirancang untuk memblokir akses ke sistem komputer atau data sampai pembayaran dilakukan.

Matriks RASCI

Lihat [bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan \(RACI\)](#).

RCAC

Lihat [kontrol akses baris dan kolom](#).

## replika baca

Salinan database yang digunakan untuk tujuan read-only. Anda dapat merutekan kueri ke replika baca untuk mengurangi beban pada database utama Anda.

## arsitek ulang

Lihat [7 Rs](#).

## tujuan titik pemulihan (RPO)

Jumlah waktu maksimum yang dapat diterima sejak titik pemulihan data terakhir. Ini menentukan apa yang dianggap sebagai kehilangan data yang dapat diterima antara titik pemulihan terakhir dan gangguan layanan.

## tujuan waktu pemulihan (RTO)

Penundaan maksimum yang dapat diterima antara gangguan layanan dan pemulihan layanan.

## refactor

Lihat [7 Rs](#).

## Region

Kumpulan AWS sumber daya di wilayah geografis. Masing-masing Wilayah AWS terisolasi dan independen dari yang lain untuk memberikan toleransi kesalahan, stabilitas, dan ketahanan. Untuk informasi selengkapnya, lihat [Menentukan Wilayah AWS akun yang dapat digunakan](#).

## regresi

Teknik ML yang memprediksi nilai numerik. Misalnya, untuk memecahkan masalah “Berapa harga rumah ini akan dijual?” Model ML dapat menggunakan model regresi linier untuk memprediksi harga jual rumah berdasarkan fakta yang diketahui tentang rumah (misalnya, luas persegi).

## rehost

Lihat [7 Rs](#).

## melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

## memindahkan

Lihat [7 Rs](#).

## memplatform ulang

Lihat [7 Rs](#).

## pembelian kembali

Lihat [7 Rs](#).

## ketahanan

Kemampuan aplikasi untuk melawan atau pulih dari gangguan. [Ketersediaan tinggi](#) dan [pemulihan bencana](#) adalah pertimbangan umum ketika merencanakan ketahanan di AWS Cloud. Untuk informasi lebih lanjut, lihat [AWS Cloud Ketahanan](#).

## kebijakan berbasis sumber daya

Kebijakan yang dilampirkan ke sumber daya, seperti bucket Amazon S3, titik akhir, atau kunci enkripsi. Jenis kebijakan ini menentukan prinsipal mana yang diizinkan mengakses, tindakan yang didukung, dan kondisi lain yang harus dipenuhi.

## matriks yang bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI)

Matriks yang mendefinisikan peran dan tanggung jawab untuk semua pihak yang terlibat dalam kegiatan migrasi dan operasi cloud. Nama matriks berasal dari jenis tanggung jawab yang didefinisikan dalam matriks: bertanggung jawab (R), akuntabel (A), dikonsultasikan (C), dan diinformasikan (I). Jenis dukungan (S) adalah opsional. Jika Anda menyertakan dukungan, matriks disebut matriks RASCI, dan jika Anda mengecualikannya, itu disebut matriks RACI.

## kontrol responsif

Kontrol keamanan yang dirancang untuk mendorong remediasi efek samping atau penyimpangan dari garis dasar keamanan Anda. Untuk informasi selengkapnya, lihat [Kontrol responsif](#) dalam Menerapkan kontrol keamanan pada AWS.

## melestarikan

Lihat [7 Rs](#).

## pensiun

Lihat [7 Rs](#).

## Retrieval Augmented Generation (RAG)

Teknologi [AI generatif](#) di mana [LLM](#) mereferensikan sumber data otoritatif yang berada di luar sumber data pelatihannya sebelum menghasilkan respons. Misalnya, model RAG mungkin

melakukan pencarian semantik dari basis pengetahuan organisasi atau data kustom. Untuk informasi lebih lanjut, lihat [Apa itu RAG](#).

## rotasi

Proses memperbarui [rahasia](#) secara berkala untuk membuatnya lebih sulit bagi penyerang untuk mengakses kredensial.

## kontrol akses baris dan kolom (RCAC)

Penggunaan ekspresi SQL dasar dan fleksibel yang telah menetapkan aturan akses. RCAC terdiri dari izin baris dan topeng kolom.

## RPO

Lihat [tujuan titik pemulihan](#).

## RTO

Lihat [tujuan waktu pemulihan](#).

## buku runbook

Satu set prosedur manual atau otomatis yang diperlukan untuk melakukan tugas tertentu. Ini biasanya dibangun untuk merampingkan operasi berulang atau prosedur dengan tingkat kesalahan yang tinggi.

## D

### SAML 2.0

Standar terbuka yang digunakan oleh banyak penyedia identitas (IdPs). Fitur ini memungkinkan sistem masuk tunggal gabungan (SSO), sehingga pengguna dapat masuk ke Konsol Manajemen AWS atau memanggil operasi AWS API tanpa Anda harus membuat pengguna di IAM untuk semua orang di organisasi Anda. Untuk informasi lebih lanjut tentang federasi berbasis SAMP 2.0, lihat [Tentang federasi berbasis SAMP 2.0](#) dalam dokumentasi IAM.

### PENIPUAN

Lihat [kontrol pengawasan dan akuisisi data](#).

### SCP

Lihat [kebijakan kontrol layanan](#).

## Rahasia

Dalam AWS Secrets Manager, informasi rahasia atau terbatas, seperti kata sandi atau kredensi pengguna, yang Anda simpan dalam bentuk terenkripsi. Ini terdiri dari nilai rahasia dan metadatanya. Nilai rahasia dapat berupa biner, string tunggal, atau beberapa string. Untuk informasi selengkapnya, lihat [Apa yang ada di rahasia Secrets Manager?](#) dalam dokumentasi Secrets Manager.

## keamanan dengan desain

Pendekatan rekayasa sistem yang memperhitungkan keamanan melalui seluruh proses pengembangan.

## kontrol keamanan

Pagar pembatas teknis atau administratif yang mencegah, mendeteksi, atau mengurangi kemampuan pelaku ancaman untuk mengeksploitasi kerentanan keamanan. [Ada empat jenis kontrol keamanan utama: preventif, detektif, responsif, dan proaktif.](#)

## pengerasan keamanan

Proses mengurangi permukaan serangan untuk membuatnya lebih tahan terhadap serangan. Ini dapat mencakup tindakan seperti menghapus sumber daya yang tidak lagi diperlukan, menerapkan praktik keamanan terbaik untuk memberikan hak istimewa paling sedikit, atau menonaktifkan fitur yang tidak perlu dalam file konfigurasi.

## sistem informasi keamanan dan manajemen acara (SIEM)

Alat dan layanan yang menggabungkan sistem manajemen informasi keamanan (SIM) dan manajemen acara keamanan (SEM). Sistem SIEM mengumpulkan, memantau, dan menganalisis data dari server, jaringan, perangkat, dan sumber lain untuk mendeteksi ancaman dan pelanggaran keamanan, dan untuk menghasilkan peringatan.

## otomatisasi respons keamanan

Tindakan yang telah ditentukan dan diprogram yang dirancang untuk secara otomatis merespons atau memulihkan peristiwa keamanan. Otomatisasi ini berfungsi sebagai kontrol keamanan [detektif](#) atau [responsif](#) yang membantu Anda menerapkan praktik terbaik AWS keamanan. Contoh tindakan respons otomatis termasuk memodifikasi grup keamanan VPC, menambal instans Amazon EC2, atau memutar kredensial.

## enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

## kebijakan kontrol layanan (SCP)

Kebijakan yang menyediakan kontrol terpusat atas izin untuk semua akun di organisasi. AWS Organizations SCPs menentukan pagar pembatas atau menetapkan batasan pada tindakan yang dapat didelegasikan oleh administrator kepada pengguna atau peran. Anda dapat menggunakan SCPs daftar izin atau daftar penolakan, untuk menentukan layanan atau tindakan mana yang diizinkan atau dilarang. Untuk informasi selengkapnya, lihat [Kebijakan kontrol layanan](#) dalam AWS Organizations dokumentasi.

## titik akhir layanan

URL titik masuk untuk file Layanan AWS. Anda dapat menggunakan endpoint untuk terhubung secara terprogram ke layanan target. Untuk informasi selengkapnya, lihat [Layanan AWS titik akhir](#) di Referensi Umum AWS.

## perjanjian tingkat layanan (SLA)

Perjanjian yang menjelaskan apa yang dijanjikan tim TI untuk diberikan kepada pelanggan mereka, seperti waktu kerja dan kinerja layanan.

## indikator tingkat layanan (SLI)

Pengukuran aspek kinerja layanan, seperti tingkat kesalahan, ketersediaan, atau throughputnya.

## tujuan tingkat layanan (SLO)

Metrik target yang mewakili kesehatan layanan, yang diukur dengan indikator [tingkat layanan](#).

## model tanggung jawab bersama

Model yang menjelaskan tanggung jawab yang Anda bagikan AWS untuk keamanan dan kepatuhan cloud. AWS bertanggung jawab atas keamanan cloud, sedangkan Anda bertanggung jawab atas keamanan di cloud. Untuk informasi selengkapnya, lihat [Model tanggung jawab bersama](#).

## SIEM

Lihat [informasi keamanan dan sistem manajemen acara](#).

## titik kegagalan tunggal (SPOF)

Kegagalan dalam satu komponen penting dari aplikasi yang dapat mengganggu sistem.

## SLA

Lihat [perjanjian tingkat layanan](#).

## SLI

Lihat [indikator tingkat layanan](#).

## SLO

Lihat [tujuan tingkat layanan](#).

## split-and-seed model

Pola untuk menskalakan dan mempercepat proyek modernisasi. Ketika fitur baru dan rilis produk didefinisikan, tim inti berpisah untuk membuat tim produk baru. Ini membantu meningkatkan kemampuan dan layanan organisasi Anda, meningkatkan produktivitas pengembang, dan mendukung inovasi yang cepat. Untuk informasi lebih lanjut, lihat [Pendekatan bertahap untuk memodernisasi aplikasi](#) di AWS Cloud

## SPOF

Lihat [satu titik kegagalan](#).

## skema bintang

Struktur organisasi database yang menggunakan satu tabel fakta besar untuk menyimpan data transaksional atau terukur dan menggunakan satu atau lebih tabel dimensi yang lebih kecil untuk menyimpan atribut data. Struktur ini dirancang untuk digunakan dalam [gudang data](#) atau untuk tujuan intelijen bisnis.

## pola ara pencekik

Pendekatan untuk memodernisasi sistem monolitik dengan menulis ulang secara bertahap dan mengganti fungsionalitas sistem sampai sistem warisan dapat dinonaktifkan. Pola ini menggunakan analogi pohon ara yang tumbuh menjadi pohon yang sudah mapan dan akhirnya mengatasi dan menggantikan inangnya. Pola ini [diperkenalkan oleh Martin Fowler](#) sebagai cara untuk mengelola risiko saat menulis ulang sistem monolitik. Untuk contoh cara menerapkan pola ini, lihat [Memodernisasi layanan web Microsoft ASP.NET \(ASMX\) lama secara bertahap menggunakan container dan Amazon API Gateway](#).

## subnet

Rentang alamat IP dalam VPC Anda. Subnet harus berada di Availability Zone tunggal.

## kontrol pengawasan dan akuisisi data (SCADA)

Di bidang manufaktur, sistem yang menggunakan perangkat keras dan perangkat lunak untuk memantau aset fisik dan operasi produksi.

## enkripsi simetris

Algoritma enkripsi yang menggunakan kunci yang sama untuk mengenkripsi dan mendekripsi data.

## pengujian sintetis

Menguji sistem dengan cara yang mensimulasikan interaksi pengguna untuk mendeteksi potensi masalah atau untuk memantau kinerja. Anda dapat menggunakan [Amazon CloudWatch Synthetics](#) untuk membuat tes ini.

## sistem prompt

Teknik untuk memberikan konteks, instruksi, atau pedoman ke [LLM](#) untuk mengarahkan perilakunya. Permintaan sistem membantu mengatur konteks dan menetapkan aturan untuk interaksi dengan pengguna.

# T

## tag

Pasangan nilai kunci yang bertindak sebagai metadata untuk mengatur sumber daya Anda. AWS Tanda membantu Anda mengelola, mengidentifikasi, mengatur, dan memfilter sumber daya. Untuk informasi selengkapnya, lihat [Menandai sumber daya AWS](#).

## variabel target

Nilai yang Anda coba prediksi dalam ML yang diawasi. Ini juga disebut sebagai variabel hasil. Misalnya, dalam pengaturan manufaktur, variabel target bisa menjadi cacat produk.

## daftar tugas

Alat yang digunakan untuk melacak kemajuan melalui runbook. Daftar tugas berisi ikhtisar runbook dan daftar tugas umum yang harus diselesaikan. Untuk setiap tugas umum, itu termasuk perkiraan jumlah waktu yang dibutuhkan, pemilik, dan kemajuan.

## lingkungan uji

Lihat [lingkungan](#).

## pelatihan

Untuk menyediakan data bagi model ML Anda untuk dipelajari. Data pelatihan harus berisi jawaban yang benar. Algoritma pembelajaran menemukan pola dalam data pelatihan yang

memetakan atribut data input ke target (jawaban yang ingin Anda prediksi). Ini menghasilkan model ML yang menangkap pola-pola ini. Anda kemudian dapat menggunakan model ML untuk membuat prediksi pada data baru yang Anda tidak tahu targetnya.

### gerbang transit

Hub transit jaringan yang dapat Anda gunakan untuk menghubungkan jaringan Anda VPCs dan lokal. Untuk informasi selengkapnya, lihat [Apa itu gateway transit](#) dalam AWS Transit Gateway dokumentasi.

### alur kerja berbasis batang

Pendekatan di mana pengembang membangun dan menguji fitur secara lokal di cabang fitur dan kemudian menggabungkan perubahan tersebut ke cabang utama. Cabang utama kemudian dibangun untuk pengembangan, praproduksi, dan lingkungan produksi, secara berurutan.

### akses tepercaya

Memberikan izin ke layanan yang Anda tentukan untuk melakukan tugas di organisasi Anda di dalam AWS Organizations dan di akunnya atas nama Anda. Layanan tepercaya menciptakan peran terkait layanan di setiap akun, ketika peran itu diperlukan, untuk melakukan tugas manajemen untuk Anda. Untuk informasi selengkapnya, lihat [Menggunakan AWS Organizations dengan AWS layanan lain](#) dalam AWS Organizations dokumentasi.

### penyetelan

Untuk mengubah aspek proses pelatihan Anda untuk meningkatkan akurasi model ML. Misalnya, Anda dapat melatih model ML dengan membuat set pelabelan, menambahkan label, dan kemudian mengulangi langkah-langkah ini beberapa kali di bawah pengaturan yang berbeda untuk mengoptimalkan model.

### tim dua pizza

Sebuah DevOps tim kecil yang bisa Anda beri makan dengan dua pizza. Ukuran tim dua pizza memastikan peluang terbaik untuk berkolaborasi dalam pengembangan perangkat lunak.

## U

### waswas

Sebuah konsep yang mengacu pada informasi yang tidak tepat, tidak lengkap, atau tidak diketahui yang dapat merusak keandalan model ML prediktif. Ada dua jenis ketidakpastian:

ketidakpastian epistemik disebabkan oleh data yang terbatas dan tidak lengkap, sedangkan ketidakpastian aleatorik disebabkan oleh kebisingan dan keacakan yang melekat dalam data.

tugas yang tidak terdiferensiasi

Juga dikenal sebagai angkat berat, pekerjaan yang diperlukan untuk membuat dan mengoperasikan aplikasi tetapi itu tidak memberikan nilai langsung kepada pengguna akhir atau memberikan keunggulan kompetitif. Contoh tugas yang tidak terdiferensiasi termasuk pengadaan, pemeliharaan, dan perencanaan kapasitas.

lingkungan atas

Lihat [lingkungan](#).

## V

menyedot debu

Operasi pemeliharaan database yang melibatkan pembersihan setelah pembaruan tambahan untuk merebut kembali penyimpanan dan meningkatkan kinerja.

kendali versi

Proses dan alat yang melacak perubahan, seperti perubahan kode sumber dalam repositori.

Peering VPC

Koneksi antara dua VPCs yang memungkinkan Anda untuk merutekan lalu lintas dengan menggunakan alamat IP pribadi. Untuk informasi selengkapnya, lihat [Apa itu peering VPC](#) di dokumentasi VPC Amazon.

kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

## W

cache hangat

Cache buffer yang berisi data terkini dan relevan yang sering diakses. Instance database dapat membaca dari cache buffer, yang lebih cepat daripada membaca dari memori utama atau disk.

## data hangat

Data yang jarang diakses. Saat menanyakan jenis data ini, kueri yang cukup lambat biasanya dapat diterima.

## fungsi jendela

Fungsi SQL yang melakukan perhitungan pada sekelompok baris yang berhubungan dengan catatan saat ini. Fungsi jendela berguna untuk memproses tugas, seperti menghitung rata-rata bergerak atau mengakses nilai baris berdasarkan posisi relatif dari baris saat ini.

## beban kerja

Kumpulan sumber daya dan kode yang memberikan nilai bisnis, seperti aplikasi yang dihadapi pelanggan atau proses backend.

## aliran kerja

Grup fungsional dalam proyek migrasi yang bertanggung jawab atas serangkaian tugas tertentu. Setiap alur kerja independen tetapi mendukung alur kerja lain dalam proyek. Misalnya, alur kerja portofolio bertanggung jawab untuk memprioritaskan aplikasi, perencanaan gelombang, dan mengumpulkan metadata migrasi. Alur kerja portofolio mengirimkan aset ini ke alur kerja migrasi, yang kemudian memigrasikan server dan aplikasi.

## CACING

Lihat [menulis sekali, baca banyak](#).

## WQF

Lihat [AWS Kerangka Kualifikasi Beban Kerja](#).

## tulis sekali, baca banyak (WORM)

Model penyimpanan yang menulis data satu kali dan mencegah data dihapus atau dimodifikasi. Pengguna yang berwenang dapat membaca data sebanyak yang diperlukan, tetapi mereka tidak dapat mengubahnya. Infrastruktur penyimpanan data ini dianggap [tidak dapat diubah](#).

## Z

### eksploitasi zero-day

Serangan, biasanya malware, yang memanfaatkan kerentanan [zero-day](#).

## kerentanan zero-day

Cacat atau kerentanan yang tak tanggung-tanggung dalam sistem produksi. Aktor ancaman dapat menggunakan jenis kerentanan ini untuk menyerang sistem. Pengembang sering menyadari kerentanan sebagai akibat dari serangan tersebut.

## bisikan zero-shot

Memberikan [LLM](#) dengan instruksi untuk melakukan tugas tetapi tidak ada contoh (tembakan) yang dapat membantu membimbingnya. LLM harus menggunakan pengetahuan pra-terlatih untuk menangani tugas. Efektivitas bidikan nol tergantung pada kompleksitas tugas dan kualitas prompt. Lihat juga beberapa [bidikan yang diminta](#).

## aplikasi zombie

Aplikasi yang memiliki CPU rata-rata dan penggunaan memori di bawah 5 persen. Dalam proyek migrasi, adalah umum untuk menghentikan aplikasi ini.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.