

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/382304762>

Konsep Sistem Informasi

Book · October 2023

CITATIONS

0

READS

6,125

1 author:



Zatin Niqotaini
Universitas Pembangunan Nasional Veteran Jakarta

58 PUBLICATIONS 143 CITATIONS

[SEE PROFILE](#)



KONSEP SISTEM INFORMASI



Arif Rizki Marsa, S.Kom., M.Kom., **Rosda Syelly**, M.Kom., **Riska Amelia**, S.Kom., M.Kom.,
Ajang Sopandi, S.Kom., M.Kom., **Suharsono**, S.Kom., M.Kom., **Rifka Dwi Amalia**, S.Pd., M.Kom.,
Ranti Irsa, S.Kom., M.Kom., **Widiyawati**, S.Kom., M.Kom., **Nurhuda Maulana**, M.T.,
Nindy Irzavika, S.Si., M.T., **Musthofa Galih Pradana**, M.Kom.,
Kharisma Wati Gusti, S.T., M.T., **Arif Budiman**, S.Kom., M.Kom., **Zatin Niqotaini**, S.Tr.Kom.,
M.Kom., **Imanaji Hari Sayekti**, S.Pd., M.Pd.

KONSEP SISTEM INFORMASI

Arif Rizki Marsa, S.Kom., M.Kom., Rosda Syelly, S.Kom.,
M.Kom., Riska Amelia, Ajang Sopandi, S.Kom., M.Kom,
Suharsono, S.Kom. M.Kom., Rifka Dwi Amalia, S.Pd.,
M.Kom., Ranti Irsa, Widiyawati, S.Kom , M.Kom., Nurhuda
Maulana, Nindy Irzavika, Musthofa Galih Pradana, M.Kom,
Kharisma Wiaty Gusti, S.T.,M.T., Arif Budiman, S.Kom.,
M.Kom., Zatin Niqotaini, S.Tr.Kom., M.Kom, Imanaji Hari
Sayekti, S.Pd., M.Pd.



Konsep Sistem Informasi

Copyright© PT Penamudamedia, 2023

Penulis:

Arif Rizki Marsa, S.Kom., M.Kom., Rosda Syelly, S.Kom., M.Kom., Riska Amelia, Ajang Sopandi, S.Kom., M.Kom, Suharsono, S.Kom. M.Kom., Rifka Dwi Amalia, S.Pd., M.Kom., Ranti Irsa, Widiyawati, S.Kom , M.Kom., Nurhuda Maulana, Nindy Irzavika, Musthofa Galih Pradana, M.Kom, Kharisma Wati Gusti, S.T.,M.T., Arif Budiman, S.Kom., M.Kom., Zatin Niqotaini, S.Tr.Kom., M.Kom, Imanaji Hari Sayekti, S.Pd., M.Pd.

Editor:

Dr. Dwi Prasetyo, Dipl.Inf, S.Kom, M.Si

ISBN:

978-623-09-5871-7

Desain Sampul:

Tim PT Penamuda Media

Tata Letak:

Enbookdesign

Diterbitkan Oleh

PT Penamuda Media

Casa Sidoarium RT 03 Ngentak, Sidoarium Dodeam Sleman Yogyakarta

HP/Whatsapp	: +6285700592256
Email	: penamudamedia@gmail.com
Web	: www.penamuda.com
Instagram	: @penamudamedia

Cetakan Pertama, Oktober 2023

x + 164, 15x23 cm

*Hak cipta dilindungi oleh undang-undang
Dilarang memperbanyak sebagian atau seluruh isi buku
tanpa izin Penerbit*

Kata Pengantar

Sistem Informasi adalah fondasi dari dunia yang semakin terhubung dan digital di mana kita hidup saat ini. Buku ini, "Konsep Dasar Sistem Informasi," adalah panduan komprehensif yang membahas aspek-aspek kunci dalam pemahaman sistem informasi yang mendalam. Kami menyediakan ringkasan yang padat dan relevan untuk para pembaca yang ingin memahami peran dan prinsip dasar sistem informasi.

Dalam buku ini, kami akan membahas dasar-dasar sistem informasi, mulai dari pengenalan konsep sistem informasi hingga eksplorasi komponen utamanya. Anda akan memahami bagaimana teknologi, data, manusia, dan proses berinteraksi dalam ekosistem sistem informasi yang kompleks. Kami akan membahas peran sistem informasi dalam mendukung operasi organisasi, pengambilan keputusan, dan pencapaian tujuan bisnis.

Selain konsep dasar, buku ini juga akan mengenalkan pembaca pada berbagai jenis sistem informasi, termasuk Sistem Pendukung Keputusan (SPK), Database Management System (DBMS), dan Sistem Enterprise Resource Planning (ERP). Masing-masing sistem ini memiliki peran unik dalam berbagai konteks bisnis dan organisasi.

Selanjutnya, kami akan mengulas isu-isu penting dalam dunia sistem informasi, seperti keamanan informasi dan privasi data. Tantangan-tantangan ini semakin mendalam seiring dengan per-

tumbuhan teknologi informasi, dan pemahaman yang kuat tentang mereka sangat penting. Selain itu membahas peran manusia dalam sistem informasi, dari pengguna akhir hingga pengambil keputusan. Penggunaan teknologi yang efektif dan pemahaman tentang bagaimana orang berinteraksi dengan sistem informasi adalah faktor penting dalam kesuksesan implementasi. Buku ini disusun dengan harapan bahwa akan memberikan pemahaman yang bermanfaat bagi pembaca tentang dunia sistem informasi yang semakin kompleks. Kami berharap buku ini akan menjadi sumber daya yang berharga untuk mahasiswa, profesional, dan siapa pun yang ingin memahami dasar-dasar sistem informasi.

Terakhir, kami ingin mengucapkan terima kasih kepada para ahli dan praktisi yang telah memberikan wawasan mereka dalam pembuatan buku ini. Kontribusi mereka adalah hal yang sangat berharga dan memastikan bahwa buku ini memberikan informasi yang akurat dan relevan dalam dunia yang terus berubah.

Selamat membaca, dan semoga buku ini memberikan Anda pemahaman yang lebih dalam tentang peran penting sistem informasi dalam dunia yang semakin terkoneksi dan digital.

Dr. DWI PRASETYO, Dipl.Inf, S.Kom, M.Si
Editor

Daftar Isi

Kata Pengantar	v
Daftar Isi.....	vii
Bab 1 Konsep Dasar Sistem	1
A. Defenisi Sistem	1
B. Lingkungan Sistem.....	2
C. Karakteristik Sistem.....	3
D. Klasifikasi Sistem	4
E. Pendekatan Sistem.....	6
F. Model dan pembuatan model (modeling) sistem.....	8
Bab 2 Konsep Dasar Informasi.....	10
A. Defenisi Informasi	10
B. Kualitas Informasi.....	12
C. Jenis-jenis Informasi.....	14
Bab 3 Jenis-Jenis Sistem Informasi.....	17
A. Transaction Processing System (TPS)	17
B. Sistem Informasi Manajemen (SIM).....	18
C. Sistem Pendukung Keputusan.....	21
D. Executive Information Systems.....	22
E. Intelligent Business	24
F. Enterprise Resource Planning (ERP):.....	25
G. Customer Relationship Diagram	26

H.	Knowledge Management Systems (KMS)	27
I.	Sistem Informasi Geografis (SIG):.....	29
J.	Expert System dan Artificial Intelligence.....	30
Bab 4 Manajemen Informasi	31	
A.	Apa itu Manajemen Informasi?.....	31
B.	Tujuan Manajemen Infromasi	32
C.	Manajemen Informasi dan Manajemen Data	33
D.	Manajemen Informasi dan Perannya dalam Sebuah Bisnis	34
Bab 5 Sistem Bilangan	37	
A.	Pengertian Sistem Bilangan.....	37
B.	Jenis Sistem Bilangan.....	38
C.	Konversi Sistem Bilangan	41
Bab 6 Pemahaman Bahasa Sistem.....	48	
A.	Pengertian Bahasa Sistem	48
B.	Komponen Utama Bahasa Sistem	50
C.	Bahasa Formal dan Informal dalam Sistem Informasi	53
D.	Peran Bahasa Sistem dalam Pengembangan Sistem Informasi	55
E.	Studi Kasus	56
Bab 7 Basis Data.....	58	
A.	Konsep Dasar Basis Data	58
B.	Sistem berbasis File	59
C.	Definisi Basis Data	60
D.	Komponen Sistem basis data	60
E.	Jenjang Basis Data	61
F.	Karakter dan Manfaat Basis Data.....	63
G.	Operasi Dasar Basis Data.....	65
H.	Bahasa Basis Data.....	65
Bab 8 Sistem Manajemen Basis Data	66	

A. Pengertian Sistem Manajemen Basis Data	66
B. Komponen Sistem Manajemen Basis Data (DBMS)	68
C. Fungsi dan Pelayanan Sistem Manajemen Basis Data (DBMS)	71
D. Kelebihan dan Kekurangan Dalam Penggunaan Sistem Manajemen Basis Data	74
E. Jenis-Jenis Sistem Manajemen Basis Data (DBMS)	76
F. Macam-Macam Aplikasi atau Software Database Management system (DBMS)	77
G. Peranan Sistem Manajemen Basis Data Dalam Perusahaan	78
H. Hambatan Dan Tantangan Sistem Manajemen Basis Data (DBMS)	79
Bab 9 Arsitektur Basis Data	80
A. Apa Itu Arsitektur Sistem Basis Data?	81
B. Arsitektur Satu Tingkat (<i>1-Tier Architecture</i>)	84
C. Arsitektur Dua Tingkat (<i>Two Tier Architecture</i>).....	85
D. Arsitektur Tiga Tingkat (<i>Three Tier Architecture</i>)	87
Bab 10 Model Data	92
A. Jenis-Jenis Model Data	93
B. Siklus Hidup Basis Data.....	97
Bab 11 Normalisasi.....	100
A. Basis Data dan Redudansi Data.....	100
B. Normalisasi Data	102
C. Contoh Kasus Normalisasi Data	103
Bab 12 Data Warehouse	112
A. Konsep Dasar Data Warehouse.....	112
B. Desain Data Warehouse.....	116
C. Implementasi Data Warehouse.....	118
D. Tantangan dan Kesempatan Data Warehouse	119
E. Peran Data Warehouse dalam Sistem Informasi	121

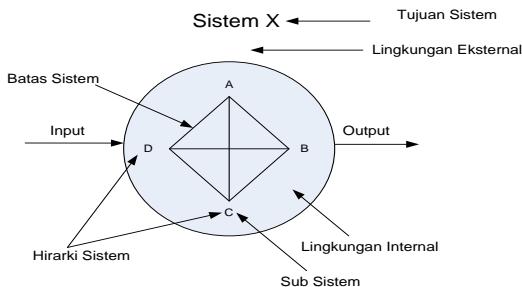
Bab 13 Pengenalan ERD.....	123
A. Komponen Penyusunan ERD	124
B. Merancang ERD	127
Bab 14 Sistem Manajemen Pengetahuan	131
A. Manajemen Pengetahuan (Knowledge Management)	131
B. Sistem Manajemen Pengetahuan, juga dikenal sebagai KMS.....	133
C. Komponen utama dari sistem manajemen pengetahuan.....	134
D. Manfaat Sistem Manajemen Pengetahuan.....	135
Bab 15 Pemodelan Proses Perangkat Lunak.....	138
A. Proses Pemodelan Rekayasa Perangkat Lunak	139
B. Pemodelan pada tahap analisis dan desain	145
Daftar Pustaka	151
Tentang Penulis	157



Konsep Dasar Sistem

A. Defenisi Sistem

Sistem menurut Hall sistem adalah kelompok dari dua atau lebih komponen atau subsistem yang saling berhubungan yang berfungsi dengan tujuan yang sama. Adapun menurut Fat pengertian sistem adalah sebagai berikut : Sistem adalah suatu himpunan benda nyata atau abstrak (a set of thing) yang terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, ketergantungan, saling mendukung, yang secara keseluruhan bersatu dalam satu kesatuan untuk mencapai tujuan tertentu secara efisien dan efektif.



Gambar 1. Ciri sistem

1. Tujuan Sistem : Merupakan target atau sasaran akhir yang ingin dicapai oleh suatu sistem.
2. Batas Sistem : Batas sistem merupakan garis abstraksi yang memisahkan antara sistem dan lingkungannya. Batas sistem ini bagi manusia sangat relatif dan tergantung kepada tingkat pengetahuan dan situasi kondisi yang dirasakan oleh orang yang melihat sistem tersebut.
3. Sub Sistem : Subsistem merupakan komponen atau bagian dari suatu sistem. Subsistem ini bisa fisik atau abstrak
4. Hubungan Sistem : Dalam kebanyakan hal hubungan sistem ini sangat menentukan terhadap keberhasilan suatu sistem karena dalam suatu sistem, misalnya sistem perusahaan, bagaimana tujuan sistem dapat dicapai kalau bagian – bagian sistem yang ada didalamnya tidak saling berhubungan baik atau bekerjasama.
5. I/P/O : Input merupakan segala sesuatu yang masuk ke dalam suatu system. Proses merupakan perubahan dari input menjadi output. Proses ini mungkin dilakukan oleh mesin, orang atau komputer, Output merupakan hasil dari suatu proses yang merupakan tujuan dari keberadaan sistem.

B. Lingkungan Sistem

Lingkungan sistem adalah segala sesuatu yang berada diluar sistem, lingkungan sistem dapat menguntungkan ataupun merugikan. Umumnya, lingkungan yang menguntungkan akan selalu dipertahankan untuk menjaga keberlangsungan sistem. Sedangkan lingkungan sistem yang

merugikan akan diupayakan agar mempunyai pengaruh seminimal mungkin, bahkan jika mungkin ditiadakan

C. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, menurut Gordon,B.D (1974) terdiri dari:

1. Komponen (Component) : Terdiri dari sejumlah komponen yang saling berinteraksi, dan bekerja sama membentuk satu kesatuan. Komponen-komponen dapat terdiri dari beberapa subsistem atau subbagian, dimana setiap subsistem tersebut memiliki fungsi khusus dan akan mempengaruhi proses secara keseluruhan.
2. Batas Sistem (Boundary) : Merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luar . Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu batas system menunjukkan ruang lingkup (scope) dari sistem tersebut.
3. Lingkungan Luar Sistem (Environments)
Adalah apapun diluar batas sistem yang mempengaruhi operasi sistem. Lingkungan luar dapat bersifat menguntungkan dan merugikan. Lingkungan yang menguntungkan harus tetap dijaga dan dipelihara, tetapi lingkungan yang merugikan harus ditahan dan dikendalikan, jika tidak ingin terganggu kelangsungan hidup sistem.
4. Penghubung (Interface) : Merupakan media penghubung antar subsistem, yang memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Keluaran (output) dari satu subsistem akan menjadi masukan (input) untuk subsistem lain melalui

penghubung disamping sebagai untuk mengintegrasikan subsistem-subsistem menjadi satu kesatuan.

5. Masukan (input): Adalah energi yang dimasukan ke dalam sistem , yang dapat berupa masukan perawatan (maintenance input) dan masukan sinyal (signal input). Masukan perawatan adalah energi yang dimasukan supaya sistem dapat beroperasi, sedangkan masukan sinyal adalah energi yang diproses untuk mendapatkan keluaran .
6. Keluaran (output): Adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisanya pembuangan . Keluaran dapat merupakan masukkan untuk subsistem yang lain.
7. Pengolah (proses): Suatu sistem dapat mempunyai suatu bagian pengolah yang akan mengubah masukan menjadi keluaran.
8. Sasaran (objectives) dan Tujuan (goal)
Suatu sistem pasti mempunyai tujuan (goal) atau sasaran (objectives). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada guna/manfaat. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan.

D. Klasifikasi Sistem

Sistem adalah suatu bentuk integrasi antara satu komponen dengan komponen lain karena system memiliki sasaran yang berbeda untuk setiap kasus yang terjadi di dalam system tersebut. Oleh sebab itu system dapat

diklasifikasikan dari beberapa sudut pandang, seperti dibawah ini:

1. Sistem abstrak (abstract system) dan Sistem fisik (physical system).

Sistem abstrak adalah sistem yang berupa pemikiran-pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologia, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik. Misalnya sistem komputer, sistem akutansi, sistem produksi dan lain sebagainya.

2. Sistem alamiah (natural system) dan Sitem buatan manusia (human made system)

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan human-machine system atau ada yang menyebut dengan man-machine sistem. Sistem informasi merupakan contoh manmachine sistem, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Sistem tertentu dan Sistem tak tentu (probabilitas sistem)

Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti. Sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah contoh dari sistem interaksi yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan. Sistem tak tentu adalah

sistem yang kondisi masa depannya tidak dapat diprediksikan karena mengandung unsur probabilitas.

4. Sistem tertutup (close system) dan Sistem terbuka (open system)

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah relatively closedsistem (secara relatif tertutup,tidak benar-benar tertutup). Sistem terbuka adalah system yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya. Karena sistem sifatnya terbuka dan terpengaruh oleh lingkungan luarnya, maka sistem harus mempunyai suatu sistem pengendalian yang baik. Sistem yang baik harus dirancang sedemikian rupa, sehingga secara relatif tertutup karena sistem tertutup akan bekerja secara otomatis dan terbuka hanya untuk pengaruh yang baik saja.

E. Pendekatan Sistem

Merupakan kerangka kerja umum dalam pengambilan keputusan yang berdasarkan kepada empat pikiran utama

1. Pendekatan sistem mengharuskan kita menentukan sistem tersebut dalam bentuk karakteristik
2. Pendekatan sistem mengharuskan kita mempertimbangkan sistem secara keseluruhan

3. Pendekatan sistem berasumsi bahwa selalu ada beberapa alternatif
4. Pendekatan sistem memerlukan penerapan metode ilmiah

Tahapan pendekatan ilmiah :

1. Identifikasi sistem: Pengenalan dan pemahaman tentang sistem yang akan diteliti.
2. Pengumpulan data: Pengumpulan data tentang sistem melalui pengamatan atau pengukuran.
3. Pembuatan model: Pembuatan model sistem berdasarkan data yang telah dikumpulkan.
4. Perumusan hipotesis: Identifikasi hipotesis yang akan diuji mengenai sistem.
5. Eksperimen: Pengujian hipotesis melalui eksperimen pada model sistem.
6. Analisis data: Analisis data dari hasil eksperimen untuk menarik kesimpulan tentang hipotesis.
7. Verifikasi: Verifikasi atas kesimpulan yang didapat melalui pengujian dan analisis.
8. Validasi: Validasi atas kesimpulan untuk memastikan kebenaran dan keandalan hasil.
9. Revisi model: Revisi atas model sistem berdasarkan hasil verifikasi dan validasi.
10. Komunikasi: Memaparkan hasil penelitian dan temuan tentang sistem kepada orang lain untuk peningkatan pemahaman mengenai sistem tersebut..

F. Model dan pembuatan model (modeling) sistem

Model sistem adalah representasi sederhana dari sistem yang diteliti, yang dibangun berdasarkan data dan informasi yang telah dikumpulkan. Modeling dilakukan untuk membantu pemahaman dan analisis lebih lanjut terhadap sistem yang kompleks dan sulit dipahami. Ada beberapa tahapan dalam pembuatan model sistem, antara lain:

1. Identifikasi variabel dan parameter: Variabel adalah faktor-faktor yang berubah dalam sistem, sedangkan parameter adalah ukuran atau nilai tetap dalam sistem. Kedua hal ini harus diidentifikasi secara jelas sebelum memulai pembuatan model.
2. Penetapan skala dan dimensi: Setiap variabel dan parameter harus memiliki skala dan unit yang sama dalam model, dan harus diukur dalam satuan yang konsisten.
3. Formulasi relasi antar variabel: Variabel dalam model harus dihubungkan satu sama lain dengan rumusan matematis yang tepat dan relevan.
4. Verifikasi dan validasi model: Setelah model telah dibuat, perlu dilakukan verifikasi dan validasi untuk memeriksa kebenaran model. Verifikasi melibatkan pembandingan model dengan data empiris, sedangkan validasi melibatkan uji coba model pada kondisi yang berbeda.
5. Penggunaan model: Setelah model telah diverifikasi dan divalidasi, ia dapat digunakan untuk memprediksi dan memahami perilaku sistem dalam kondisi yang berbeda, dan untuk mengembangkan strategi yang lebih baik untuk mengelola atau mengoptimalkan sistem tersebut.

Modeling sistem dapat dilakukan dengan menggunakan berbagai teknik seperti penggunaan persamaan matematika,

diagram aliran atau diagram blok, atau bahkan simulasi komputer. Pembuatan model sistem yang tepat dan relevan sangat penting dalam pendekatan ilmiah, karena dapat membantu memahami serta mengembangkan strategi yang lebih baik dalam mengelola dan memperbaiki sistem tersebut.

Bahasa Sistem merupakan alat untuk berkomunikasi tentang sistem bila kita akan membuat suatu model, agar model tersebut bisa komunikatif maka model tersebut disampaikan dengan menggunakan simbol simbol yang diterima secara umum, contohnya DFD (Data Flow Diagram) dan Flowchart, kedua jenis notasi ini sering digunakan untuk merepresentasikan sistem secara visual dan mudah dipahami. DFD menggunakan simbol-simbol seperti kotak, panah, trapesium atau lingkaran untuk merepresentasikan aktivitas, data, dan penghubung antara keduanya. Sedangkan Flowchart menggunakan simbol seperti kotak, garis-garis, dan simbol lainnya untuk merepresentasikan urutan aktivitas atau proses yang ada dalam sistem.

Kedua notasi ini sangat berguna dalam membantu para pengembang sistem untuk memahami dan mengkomunikasikan ide dan konsep yang ada dalam sistem secara efektif. DFD dan Flowchart dapat menjadi alat yang penting dalam pendekatan ilmiah untuk merancang, membangun, dan mengoperasikan sistem yang efektif dan efisien.



Konsep Dasar Informasi

A. Defenisi Informasi

Untuk memahami konsep dasar informasi, maka kita mesti mengetahui terlebih dahulu defenisi dari informasi. Para ahli banyak menyampaikan defenisi dari informasi diantaranya:

1. The Liang Gie

Informasi atau keterangan adalah rangkaian kata, kalimat, gambar, atau tanda tulis lainnya yang mengandung buah pikiran maupun pengetahuan apa saja yang dapat dipergunakan oleh pemimpin dalam membuat keputusan yang tepat berdasarkan fakta ada.

2. Raymond McLeod

Informasi bermanfaat untuk pengambilan keputusan dari data yang telah diolah menjadi bentuk yang memiliki arti bagi penerimanya

3. Jogiyanto H.M.

Informasi adalah hasil dari pengolahan data menjadi bentuk yang lebih bermanfaat dan berguna bagi pemakainya.

4. **Firmanzah**

Informasi adalah data dan angka yang sudah diberi nilai dan makna

5. **Tata Sutabri**

Informasi adalah data yang telah diklasifikasi, diolah dan diinterpretasikan dengan tujuan untuk dapat digunakan dalam proses pengambilan keputusan.

Dari definisi informasi ini, dapat disimpulkan bahwa informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih bermanfaat bagi pengguna atau pemakainnya. Pada defenisi informasi selalu terhubung dengan kata-kata data.

B. Siklus Informasi

Data merupakan bentuk yang masih baku atau mentah, belum memiliki arti dan makna. Data yang diolah sesuai dengan prosesnya akan menghasilkan informasi. Data ini dapat berupa huruf, angka, symbol, suara, sinyal dan gambar.

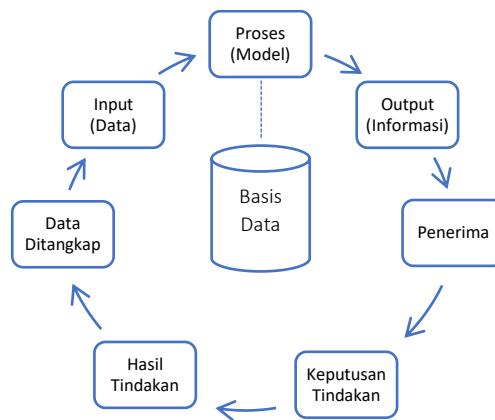


Gambar 1. Data yang Diolah Menjadi Informasi

Dalam kegiatan suatu perguruan tinggi, contohnya dalam proses belajar mengajar. Mahasiswa mengikuti proses perkuliahan dalam satu semester. Setelah mahasiswa mengikuti proses perkuliahan yang terdiri atas kehadiran perkuliahan, tugas, ujian tengah semester dan ujian akhir semester, maka akhirnya setiap mahasiswa akan menerima

laporan hasil semester (LHS). LHS ini bisa menjadi informasi bagi mahasiswa, dosen dan bagian akademik. Dari hasil LHS yang di terima mahasiswa akan menentukan total SKS yang diperoleh untuk semester berikutnya, ini merupakan keputusan tindakan.

Data yang di proses melalui suatu model menghasilkan informasi. Informasi akan di terima oleh penerimanya dan akan menentukan keputusan serta tindakan yang akan di lakukan dari informasi tersebut. Tindakan yang dilakukan akan membuat sejumlah data. Data tersebut akan di tangkap sebagai input. Kemudia akan di proses kembali dan begitu seterusnya sehingga membentuk suatu siklus yang di sebut dengan siklus informasi (*Information Cycle*).



Gambar 2. *Siklus Informasi*

B. Kualitas Informasi

Informasi merupakan hasil dari pengolahan data yang tentunya informasi itu berguna dan bermanfaat bagi si

penerima. Informasi dinyatakan berguna bagi si penerima jika di dukung oleh tiga pilar yaitu:

1. Relevan (*Relevance*)

Informasi yang bermanfaat secara langsung bagi si penerimanya. Suatu informasi yang relevan bagi si penerima belum tentu relevan bagi orang lain.

2. Akurat (*Accurate*)

Informasi yang tepat nilainya atau akurat bermanfaat bagi si penerima.

3. Tepat Waktu (*Timeliness*)

Informasi yang di terima penerimanya tepat pada waktunya atau tidak terlambat. Sehingga penerima informasi bisa menetukan tindakan yang akan di lakukan atas informasi yang diterima. Informasi bisa tidak berguna bagi penerimanya jika terlambat diterima oleh penerima informasi.

Sebuah informasi harus memiliki nilai, dimana nilai informasi itu lebih banyak manfaatnya di bandingkan biaya untuk mendapatkan informasi tersebut.

Beberapa biaya informasi yaitu :

1. Biaya perangkat keras

Biaya ini merupakan biaya tetap atau biaya tertanam dan akan meningkat untuk tingkat-tingkat mekanisme yang lebih tinggi.

2. Biaya untuk analisis, perancangan dan pelaksanaan sistem.

Biaya ini merupakan biaya tertanam dan akan meningkat sesuai dengan tingkat mekanisme yang lebih tinggi. Biaya ini meliputi perumusan suatu metodologi untuk prosedur-prosedur pengolahan data secara

keseluruhan, dan jika penggerjaannya menggunakan komputer bisa digunakan untuk persiapan pembuatan program-program.

3. Biaya untuk tempat dan faktor-faktor Kontrol lingkungan

Biaya ini semivariabel/setengah berubah-ubah. Biasanya biaya ini meningkat sesuai dengan tingkat mekanisme yang lebih tinggi. Contohnya, biaya keamanan, biaya pendingan ruangan.

4. Biaya perubahan

Biaya perubahan ini merupakan biaya tertanam dan meliputi segala perubahan dari suatu metode ke metode lain. Contoh, metode elektromekanis ke metode komputer

5. Biaya Operasi

Biaya ini merupakan biaya variabel dan mencakup biaya bermacam-macam yang berkaitan dengan kegiatan atau operasional. Contoh, Biaya pegawai, biaya pemeliharaan, biaya perlengkapan dan sistem.

Selain nilai yang tak kalah pentingnya adalah umur dari informasi. Umur informasi dilihat dari berapa lama informasi tersebut bermanfaat dan berguna bagi penerimanya.

C. Jenis-jenis Informasi

Para ahli tidak memberikan pendapat yang sama berkaitan dengan jenis-jenis informasi yang beroperasi dalam manajemen untuk. Dari perbedaan pendapat itu diambil kesimpulan bahwa informasi dalam manajemen di klasifikasikan sebagai berikut:

1. Informasi berdasarkan persyaratan

Informasi yang harus memenuhi persyaratan sesuai yang di butuhkan oleh pimpinan dalam menentukan keputusan yang akan di ambil. Berdasarkan persyaratan informasi dalam manajemen di klasifikasikan berikut ini:

a. Informasi yang tepat waktu

Informasi yang tepat waktu disini artinya informasi yang sampai kepada pimpinan sebelum suatu keputusan diambil dan informasi ini akan menjadi bahan pengambilan keputusan. Makna “tepat” disini relatif. Karena bagi satu pimpinan informasi yang sampai sehari sebelum pengambilan keputusan itu mungkin dianggap tepat. Belum tentu dengan pimpinan lain, bisa jadi membutuhkan informasi itu satu minggu sebelum pengambilan keputusan itu.

b. Informasi yang relevan

Sebuah informasi yang disampaikan pimpinan pada bawahnya haruslah relevan, mesti ada kaitannya dengan pihak penerima, sehingga informasi tersebut mendapat perhatian. Informasi dengan pihak penerima kadar relevansinya tidaklah sama.

c. Informasi yang bernilai

Informasi yang bernilai merupakan informasi yang berharga untuk mengambil keputusan bagi pihak penerima informasi.

d. Informasi yang dapat di percaya

Informasi yang dapat dipercaya artinya informasi yang harus betul-betul diyakini kebenarannya dan dapat berguna bagi pihak penerima baik individu maupun organisasi

2. Informasi yang berdasarkan dimensi waktu

Adapun klasifikasi untuk informasi berdasarkan dimensi waktu yaitu:

a. Informasi masa lalu

Informasi ini berkaitan dengan peristiwa lalu yang jarang digunakan, namun tetap disimpan dimedia penyimpanan yang disusun secara teratur dan rapi. Pada negara-negara maju, informasi mengenai peristiwa masa lalu disimpan dalam bentuk mikro film sehingga tidak memerlukan tempat dan ruangan yang besar untuk penyimpanan dan amat mudah memperolehnya.

b. Informasi masa kini

Informasi mengenai kejadian/peristiwa yang terjadi sekarang (*current events*).

3. Informasi yang berdasarkan sasaran

Informasi berdasarkan sasaran ini di klasifikasikan sebagai berikut :

a. Informasi individual (*Individual information*)

Merupakan informasi yang ditujukan pada seseorang yang mempunya fungsi sebagai pembuat kebijakan dan pengambilan keputusan terhadap informasi yang diterimanya. Informasi ini biasanya disampaikan secara tatap muka (*face-to-face*) atau melalui perantara surat. Tergantung informasi yang di sampaikan dan tergantung pada waktu yang di perlukan untuk memperoleh tanggapan.

b. Informasi komunitas

Merupakan informasi yang ditujukan pada suatu kelompok tertentu di masyarakat. Contoh, informasi yang di sampaikan berkaitan dengan pembayaran uang semester yang di tujuhan hanya pada mahasiswa yang aktif kuliah.



Jenis-Jenis Sistem Informasi

Sistem informasi adalah komponen penting dalam organisasi modern, yang memungkinkan mereka mengumpulkan, menyimpan, memproses, dan menyebarkan informasi untuk berbagai tujuan. Sistem informasi dapat dikategorikan ke dalam beberapa jenis berdasarkan fungsi dan sumber informasi yang digunakan. Berikut adalah penjelasan rinci mengenai jenis-jenis sistem informasi dan sumber-sumbernya:

A. Transaction Processing System (TPS)

TPS merupakan inti dari sistem informasi pada sebuah organisasi. TPS memainkan peran kritis dalam pengelolaan dan proses transaksi bisnis sehari-hari. TPS dirancang untuk mengotomatisasi dan menyederhanakan proses operasional yang melibatkan pengumpulan, penyimpanan, pemrosesan, dan pelaporan data transaksi.

TPS memiliki beberapa fungsi diantaranya adalah:

1. Pencatatan Transaksi

TPS mencatat semua transaksi yang terjadi dalam organisasi, seperti penjualan produk, pembelian bahan baku, dan pembayaran gaji karyawan.

2. Pengolahan Data: Sistem ini memproses data transaksi dengan cepat dan akurat, memastikan integritas data dan konsistensi dalam database.

3. Pemantauan Inventaris

TPS membantu dalam pemantauan stok barang, sehingga organisasi dapat menghindari kekurangan persediaan atau penumpukan yang berlebihan.

4. Pemrosesan Pesanan

Dalam lingkungan bisnis e-commerce, TPS dapat memproses pesanan secara otomatis dan memberikan konfirmasi kepada pelanggan.

5. Pelaporan Transaksi

TPS menghasilkan laporan transaksi yang digunakan oleh manajemen untuk pemantauan kinerja dan analisis bisnis.

B. Sistem Informasi Manajemen (SIM)

Sistem Informasi Manajemen adalah sistem komputer yang dirancang dan digunakan oleh organisasi atau perusahaan untuk mengumpulkan, mengelola, mengolah, dan menyediakan informasi yang relevan kepada manajemen dalam rangka mendukung pengambilan keputusan dan pengelolaan operasi bisnis. SIM dirancang untuk mengatasi kebutuhan informasi organisasi dan membantu dalam

meningkatkan efisiensi, produktivitas, dan kualitas pengambilan keputusan.

Berikut ini merupakan tujuan dari SIM:

1. Pengambilan Keputusan yang Lebih Baik

SIM memberikan manajer dengan informasi yang relevan dan akurat yang memungkinkan mereka untuk membuat keputusan yang lebih baik dan lebih terinformasi. Ini termasuk keputusan sehari-hari serta keputusan strategis yang lebih besar.

2. Perencanaan yang Efektif

SIM mendukung perencanaan bisnis dengan menyediakan data historis dan proyeksi masa depan. Manajer dapat menggunakan informasi ini untuk merencanakan langkah-langkah bisnis yang efektif.

3. Pengendalian Operasi

SIM memungkinkan manajer untuk memantau kinerja operasi organisasi secara real-time. Mereka dapat mendeteksi masalah atau deviasi dari rencana dan mengambil tindakan korektif.

4. Pengorganisasian yang Lebih Baik

SIM membantu dalam pengorganisasian sumber daya dan proses bisnis. Ini termasuk manajemen persediaan, manajemen SDM, dan pengelolaan aset organisasi.

5. Peningkatan Efisiensi dan Produktivitas

Dengan menyediakan akses cepat ke informasi yang dibutuhkan, SIM membantu meningkatkan efisiensi operasi organisasi. Ini dapat mengurangi waktu yang

dibutuhkan untuk mengambil keputusan dan menyelesaikan tugas-tugas.

5. Peningkatan Pelayanan Pelanggan

SIM juga dapat digunakan untuk mengumpulkan informasi tentang pelanggan dan preferensi mereka. Ini dapat membantu organisasi dalam menyediakan layanan yang lebih baik kepada pelanggan.

6. Analisis dan Peramalan

SIM memungkinkan analisis data yang mendalam dan peramalan bisnis. Ini membantu organisasi dalam memahami tren pasar, perilaku pelanggan, dan potensi risiko bisnis.

7. Meningkatkan Keamanan Sistem Informasi

Melalui kontrol akses dan pengamanan data, SIM juga membantu melindungi informasi organisasi dari akses yang tidak sah atau ancaman keamanan.

Pemantauan Kinerja Organisasi: SIM membantu manajer dalam melacak dan memahami kinerja organisasi secara keseluruhan. Mereka dapat melihat bagaimana setiap bagian organisasi berkontribusi terhadap tujuan keseluruhan.

8. Mendukung Keputusan Strategis

SIM dapat digunakan untuk analisis jangka panjang dan perencanaan strategis. Ini membantu organisasi dalam merumuskan tujuan jangka panjang dan strategi untuk mencapainya.

C. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) adalah sebuah sistem yang dirancang untuk membantu manajer dan profesional dalam pengambilan keputusan yang lebih baik, terutama dalam konteks keputusan yang kompleks dan tidak terstruktur. SPK mengintegrasikan data, alat analisis, dan model pemecahan masalah untuk menyediakan informasi yang relevan dan wawasan yang mendalam.

Berikut adalah beberapa karakteristik utama dari sistem pendukung keputusan:

1. Data: SPK mengumpulkan, menyimpan, dan mengelola data dari berbagai sumber. Data ini bisa berupa data historis, data saat ini, atau data proyeksi yang digunakan untuk analisis keputusan.
2. Model dan Algoritma: SPK menggunakan model matematika, statistik, dan algoritma untuk menganalisis data dan menghasilkan hasil yang dapat digunakan sebagai dasar untuk pengambilan keputusan.
3. Interface Pengguna: Sebagian besar SPK memiliki interface pengguna yang intuitif yang memungkinkan pengguna untuk mengakses dan menganalisis data dengan mudah. Tampilan antarmuka ini juga memungkinkan pengguna untuk memasukkan parameter, preferensi, dan tujuan mereka.
4. Analisis dan Evaluasi: SPK membantu para pengguna dalam melakukan analisis yang mendalam terhadap berbagai pilihan yang tersedia. Ini bisa meliputi analisis sensitivitas, analisis risiko, dan evaluasi dampak dari setiap pilihan.

5. Rekomendasi: Berdasarkan hasil analisis, SPK dapat memberikan rekomendasi atau alternatif keputusan kepada penggunanya. Rekomendasi ini dapat membantu pengguna dalam pengambilan keputusan dalam memilih pilihan yang paling sesuai dengan tujuan dan kriteria yang telah ditentukan.
6. Interaktif: SPK seringkali bersifat interaktif, yang berarti pengguna dapat mengubah parameter dan kriteria mereka untuk melihat bagaimana itu memengaruhi rekomendasi yang diberikan oleh sistem.
7. Berorientasi Tujuan: SPK umumnya didesain untuk membantu pengguna mencapai tujuan yang disesuaikan dengan kebutuhan pengguna.

D. Executive Information Systems

Sistem Informasi Eksekutif atau umumnya dikenal dengan Executive Information Systems (EIS) adalah jenis sistem informasi yang membantu para eksekutif dan manajer tingkat atas dalam pengambilan keputusan strategis dan pengelolaan organisasi.

Berikut adalah beberapa fungsi utama dari Sistem Informasi Eksekutif:

1. Mengumpulkan, Memproses, dan Menganalisis Data: EIS berfungsi untuk mengumpulkan data dari berbagai sumber internal dan eksternal, termasuk data operasional dan data pasar, kemudian memproses dan menganalisis data ini untuk menghasilkan informasi yang relevan dan bermanfaat bagi para eksekutif.
2. Memberikan Laporan dan Informasi yang Tepat Waktu: EIS berfungsi menyediakan laporan dan informasi secara

tepatis waktu kepada para eksekutif. Informasi ini dapat berupa gambaran tentang kinerja organisasi, tren pasar, dan berbagai metrik kunci yang membantu dalam pengambilan keputusan.

3. Membantu dalam Perencanaan dan Strategi: EIS berfungsi membantu eksekutif dalam merencanakan strategi jangka panjang dan taktik operasional. Dengan cara memberikan akses ke data historis yang memungkinkan para eksekutif dan manager untuk membuat keputusan yang lebih informasional dan berdasarkan fakta.
4. Memfasilitasi Komunikasi dan Kolaborasi: EIS memfasilitasi komunikasi dan kolaborasi antara para eksekutif dan manajer lainnya dalam organisasi. Sehingga memungkinkan pertukaran informasi yang lebih efisien.
5. Pengawasan Kinerja: EIS memungkinkan eksekutif untuk memonitor kinerja organisasi secara kontinu. Mereka dapat melihat indikator kinerja kunci (Key Performance Indicators - KPIs) dan mengidentifikasi masalah atau peluang yang memerlukan tindakan segera.
6. Pemantauan Lingkungan Bisnis: EIS membantu para eksekutif untuk mengawasi perubahan dalam lingkungan bisnis, seperti perubahan dalam regulasi, persaingan pasar, dan tren industri. Ini memungkinkan organisasi untuk beradaptasi dengan cepat dan mengambil tindakan yang tepat.
7. Keputusan Berdasarkan Analisis: Salah satu fungsi utama EIS adalah memberikan dukungan dalam pengambilan keputusan. Hal ini dapat mencakup rekomendasi

berdasarkan pada analisis data atau kemampuan untuk mengakses informasi yang diperlukan untuk mempertimbangkan berbagai pilihan.

8. Keamanan dan Aksesibilitas: SIE harus menjaga keamanan data dan informasi yang digunakan. Hal ini berarti hanya orang yang memiliki izin akses yang dapat mengakses informasi tertentu, dan data harus dilindungi dari berbagai ancaman keamanan data.

E. Inteligent Business

Inteligent business (IB) adalah pendekatan dan teknologi yang digunakan oleh organisasi untuk mengumpulkan, menyimpan, mengolah, dan menganalisis data bisnis untuk mendukung pengambilan keputusan yang lebih baik. IB berfokus pada transformasi data menjadi wawasan yang actionable (dapat dijadikan tindakan) dengan tujuan meningkatkan kinerja dan efisiensi bisnis. Sistem BI mengumpulkan data dari berbagai sumber, termasuk database, gudang data, web scraping, media sosial, dan perangkat IoT.

Fungsi Sistem Intelijen Bisnis (BI):

1. Analisis Data: BI memungkinkan organisasi untuk menggali wawasan dari data bisnis, baik yang bersumber dari internal maupun eksternal.
2. Pelaporan: BI menyediakan alat untuk pembuatan laporan yang mudah dimengerti dan dapat disesuaikan, yang membantu dalam pemantauan kinerja dan pelaporan ke berbagai pemangku kepentingan.

3. Visualisasi Data: SBI menggunakan visualisasi grafis seperti grafik, dashboard, dan infografik untuk menyampaikan informasi dengan cara yang lebih mudah dimengerti.
4. Pemantauan Real-Time: SBI memungkinkan organisasi untuk memantau data secara real-time, sehingga dapat merespons perubahan pasar dengan cepat.
5. Prediksi dan Analisis Prakiraan: BI dapat menggunakan model analisis data untuk membuat prediksi yang membantu dalam perencanaan bisnis.

Komponen Sistem Intelijen Bisnis:

- a. Data Warehouse: IB memanfaatkan data warehouse yang terintegrasi untuk menyimpan data bisnis yang jelas dan terstruktur.
- b. Alat Analisis: mencakup berbagai macam perangkat lunak dan algoritma analisis yang digunakan untuk menggali wawasan dari data.
- c. Laporan dan Dashboard: BI menghasilkan laporan standar dan dashboard yang memudahkan pemantauan dan pelaporan.

F. Enterprise Resource Planning (ERP):

Enterprise Resource Planning (ERP) adalah perangkat lunak manajemen yang digunakan oleh organisasi untuk mengintegrasikan dan mengelola proses bisnis mereka dengan lebih efisien. ERP memungkinkan berbagai departemen atau bagian untuk berbagi data dan mengintegrasikan operasi mereka dalam satu sistem terpadu.

Peran Enterprise Resource Planning (ERP):

1. Integrasi Proses Bisnis: ERP mengintegrasikan berbagai fungsi dan proses bisnis, termasuk keuangan, manufaktur, rantai pasokan, sumber daya manusia, dan penjualan, dalam satu platform terpadu.
2. Peningkatan Efisiensi: ERP meminimalkan pengulangan pekerjaan dan mengurangi kesalahan manusia dengan otomatisasi tugas rutin.
3. Pemantauan Kinerja: ERP memberikan visibilitas ke dalam kinerja organisasi secara real-time melalui laporan dan dashboard.
4. Pengambilan Keputusan: Data yang dihasilkan oleh ERP digunakan oleh manajemen untuk pengambilan keputusan yang lebih baik dan berdasarkan fakta.
5. Manajemen Rantai Pasokan: ERP membantu mengelola rantai pasokan dengan mengintegrasikan pesanan, inventaris, dan produksi.

G. Customer Relationship Diagram

Customer Relationship Diagram atau biasa disingkat dengan (CRM) adalah pendekatan dan teknologi yang digunakan oleh organisasi untuk mengelola dan meningkatkan hubungan mereka dengan pelanggan. CRM fokus pada pemahaman mendalam mengenai pelanggan, mengumpulkan data tentang interaksi pelanggan dengan perusahaan, dan memanfaatkan pengetahuan ini untuk meningkatkan customer experience (perasaan pelanggan terhadap suatu merk dari waktu ke waktu).

Peran Sistem Manajemen Hubungan Pelanggan (CRM):

1. Penyimpanan Data Pelanggan: CRM mengumpulkan dan menyimpan data pelanggan, seperti kontak, riwayat pembelian, preferensi, dan interaksi sebelumnya.
2. Analisis Data Pelanggan: CRM menganalisis data pelanggan untuk mengidentifikasi tren, preferensi, dan peluang penjualan.
3. Pelayanan Pelanggan: CRM memungkinkan organisasi memberikan pelayanan yang lebih baik dengan memberikan akses cepat ke informasi pelanggan saat berinteraksi dengan mereka.
4. Manajemen Kampanye: CRM membantu dalam merencanakan dan melaksanakan kampanye pemasaran yang lebih efektif dengan memahami pelanggan dan segmentasi pasar.
5. Peningkatan Retensi Pelanggan: CRM membantu dalam mempertahankan pelanggan yang ada dengan memberikan layanan yang lebih baik dan menyesuaikan tawaran produk.

H. Knowledge Management Systems (KMS)

Sistem perangkat lunak atau platform yang dirancang untuk mengumpulkan, menyimpan, mengelola, dan menyebarluaskan pengetahuan dan informasi di dalam suatu organisasi. Tujuan utama dari Knowledge Management Systems adalah untuk memfasilitasi proses pengelolaan pengetahuan agar informasi yang berharga dapat diakses dengan mudah oleh karyawan, digunakan secara efektif, dan berbagi dengan anggota organisasi yang membutuhkannya. Berikut beberapa fitur dan tujuan utama Knowledge Management Systems:

1. Penyimpanan dan Organisasi Pengetahuan: Sistem ini menyediakan tempat untuk menyimpan semua jenis pengetahuan, termasuk dokumen, data, catatan, pengalaman, dan informasi lainnya. Informasi ini biasanya berupa basis data atau repositori digital.
2. Pencarian dan Akses: Knowledge Management Systems menyediakan fasilitas pencarian yang kuat sehingga pengguna dapat dengan cepat menemukan informasi yang mereka butuhkan. Hal ini memungkinkan akses yang mudah ke pengetahuan yang tersedia dalam organisasi.
3. Kolaborasi: Sistem ini mendukung kolaborasi antara karyawan dengan cara memungkinkan mereka untuk berbagi pengetahuan, berdiskusi, dan bekerja sama pada proyek-proyek. Kolaborasi ini mencakup alat komunikasi internal, forum diskusi, atau fitur kolaborasi lainnya.
4. Pengelolaan Dokumen: Knowledge Management Systems biasanya memiliki fitur pengelolaan dokumen yang memungkinkan penyimpanan, versi kontrol, dan pembagian hak akses untuk dokumen-dokumen organisasi.
5. Pengidentifikasi Ahli: Sistem ini dapat membantu mengidentifikasi individu yang ahli dalam bidang tertentu berdasarkan kontribusi mereka dalam berbagi pengetahuan atau partisipasi dalam diskusi.
6. Pembelajaran dan Pengembangan: Dengan menyimpan pengetahuan dan pengalaman yang ada, Knowledge Management Systems dapat digunakan untuk memberikan pelatihan dan pembelajaran yang terkait dengan pekerjaan kepada karyawan baru atau yang ada.

7. Manajemen Proses: Beberapa sistem memiliki fitur untuk menghubungkan pengetahuan dengan proses bisnis organisasi, memastikan bahwa pengetahuan digunakan secara efektif dalam operasi sehari-hari.
8. Keamanan dan Hak Akses: memastikan bahwa pengetahuan yang bersifat sensitif atau rahasia dilindungi dengan baik dengan mengatur hak akses dan mengimplementasikan kontrol keamanan.
9. Analisis Pengetahuan: Beberapa Knowledge Management Systems dapat menganalisis data mengidentifikasi pengetahuan mengenai tren, kesenjangan dalam pengetahuan, atau kebutuhan pelatihan yang dibutuhkan.
10. Kebijakan dan Pedoman: Sistem ini dapat digunakan untuk menyimpan dan mengakses kebijakan, pedoman, dan praktik terbaik dalam organisasi

I. Sistem Informasi Geografis (SIG):

Sistem Informasi Geografis (GIS) adalah sistem yang dirancang untuk mengumpulkan, menyimpan, mengelola, menganalisis, dan memvisualisasikan data geografis. GIS memiliki banyak contoh penerapan aplikasi dalam berbagai industri, termasuk ilmu lingkungan, perencanaan kota, ilmu sosial, dan banyak lagi. GIS menggabungkan informasi spasial (lokasi geografis) dengan data atribut (informasi yang terkait dengan lokasi tersebut) untuk membantu pemahaman lebih baik tentang dunia kita.

Dibawah ini merupakan fungsi dari GIS:

1. Pemetaan dan Visualisasi: GIS memungkinkan pemetaan dan visualisasi data geografis, seperti peta, grafik, dan model 3D.
2. Analisis Spasial: GIS memungkinkan analisis berbagai masalah berdasarkan lokasi, seperti pemodelan iklim, perencanaan tata kota, dan pemantauan bencana alam.
3. Manajemen Aset: Organisasi dapat menggunakan GIS untuk melacak dan mengelola aset fisik, seperti jaringan jalan, utilitas, dan tanah.
4. Pengambilan Keputusan: GIS membantu dalam pengambilan keputusan berdasarkan pemahaman yang lebih baik tentang aspek-aspek spasial dari data.
5. Pemantauan Lingkungan: GIS digunakan dalam pemantauan dan pemahaman perubahan lingkungan, termasuk deforestasi, polusi air, dan perubahan iklim.

J. Expert System dan Artificial Intelligence

Expert Systems (ES) merupakan cabang penting dalam bidang Kecerdasan Buatan (Artificial Intelligence - AI). ES adalah program komputer yang dirancang untuk meniru pengetahuan dan keahlian manusia dalam suatu domain tertentu. Mereka digunakan untuk mengambil keputusan kompleks, memberikan rekomendasi, dan memberikan solusi berdasarkan basis pengetahuan yang ada. ES dapat diterapkan diberbagai bidang, yakni diagnostik medis, manajemen perbankan, inspeksi kualitas, dan lainnya.

ES dan Kecerdasan Buatan dapat digunakan bersama-sama atau secara terpisah, hal ini tergantung dari tujuan dan kebutuhan aplikasi yang dibutuhkan oleh pengguna.



Manajemen Informasi

Manajemen informasi merupakan aspek penting dari bisnis berbasis data yang memungkinkan pelaku bisnis membuat keputusan yang lebih baik dalam mencapai tujuan. Memahami tujuan manajemen informasi dapat membantu dalam meningkatkan penggunaan, perlindungan, dan penyimpanan informasi agar bisnis berjalan lebih efektif.

A. Apa itu Manajemen Informasi?

Manajemen informasi atau *information management* adalah pengumpulan, penyimpanan, pengelolaan dan pemeliharaan data dan jenis informasi lainnya. Ini melibatkan pengumpulan, penyebaran, pengarsipan dan penghancuran informasi dalam segala bentuknya.

Information management mencakup prosedur dan pedoman yang diadopsi organisasi untuk mengelola dan mengkomunikasikan informasi di antara individu, departemen, dan pemangku kepentingan yang berbeda.

Manajemen ini berfokus pada tingkat kontrol yang dimiliki organisasi atas informasi yang dihasilkannya. Hal ini membutuhkan pembangunan sistem manajemen informasi khusus yang dirancang untuk membantu suatu organisasi

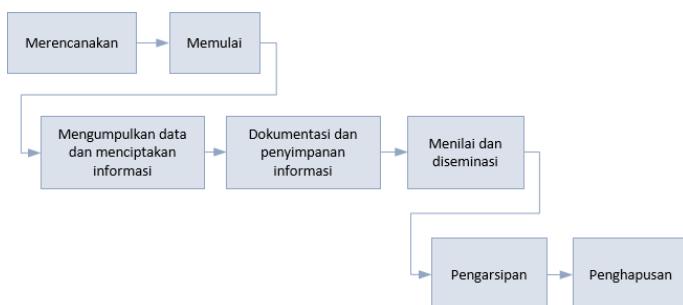
menggunakan sumber dayanya untuk mendukung proses bisnis.

Manajemen informasi juga berkaitan dengan bagaimana organisasi berbagi dan memberikan informasi kepada penerima yang beragam. Ini termasuk format seperti informasi digital dan fisik, serta media seperti komputer, server, situs web, media sosial, perangkat seluler, dan aplikasi.

B. Tujuan Manajemen Infromasi

1. Mendapatkan data secara akurat dan konsisten;
2. Menghasilkan informasi yang berguna dari data mentah;
3. Menyimpan informasi secara aman dan dapat diakses selama masa manfaatnya;
4. Mendukung komunikasi dan mengambil keputusan secara efektif.

Sejumlah besar data akan dikumpulkan selama proyek, program, atau portofolio. Tim manajemen perlu mengambil data mentah dan menghasilkan informasi melalui analisis dan interpretasi. Prosedur manajemen informasi khusus ditunjukkan di bawah ini:



Gambar 1. Prosedur Manajemen Informasi Khusus

Pembuatan data dimulai segera setelah mandat dikeluarkan sehingga prosedur dan tanggung jawab manajemen informasi perlu didefinisikan pada awal siklus hidup suatu organisasi. Ini biasanya sesuai dengan standar organisasi, tetapi mungkin perlu menyesuaikan dengan konteks tertentu seperti persyaratan peraturan atau keamanan. Standar tersebut akan ditetapkan dalam rencana manajemen informasi.

Langkah awal akan memastikan sumber daya yang diperlukan sehingga dapat dimobiliasi dan tersedianya infrasturture TI.

Pada tahap awal pengumpulan data siklus hidup akan menitikberatkan pada manajemen persyaratan dan pengembangan solusi. Kemudian beralih ke pembuatan rencana manajemen dan pengiriman yang menunjukkan bagaimana penyelesaian akan disampaikan. Seiring berjalannya waktu, data kinerja akan dikumpulkan untuk mendukung pengendalian.

C. Manajemen Informasi dan Manajemen Data

Manajemen informasi dimulai dengan pembuatan dan pengumpulan data yang diolah dan dianalisis menjadi informasi. Langkah selanjutnya adalah penyimpanan, berbagi, dan pengarsipan ketika lebih banyak data saat ini tersedia. Proses ini berakhir dengan penghancuran informasi karena berbagai alasan, termasuk keamanan, kerahasiaan, biaya, dan kendala penyimpanan.

Manajemen data adalah bagian dari manajemen informasi dan berkaitan dengan kebijakan, rencana, dan program yang melindungi dan mengontrol aset data dan

informasi melalui siklus hidupnya. Manajemen data memberi organisasi kendali atas data bisnis mereka, yang dapat meminimalkan pelanggaran keamanan dan risiko lainnya bila dilakukan dengan benar.

D. **Manajemen Informasi dan Perannya dalam Sebuah Bisnis**

Manajemen informasi memungkinkan suatu organisasi untuk mencapai berbagai tujuan. Ini meningkatkan kepatuhan, mengurangi risiko, dan mengontrol akses ke informasi bisnis penting.

Inilah mengapa manajemen informasi penting dalam suatu organisasi:

1. Mengontrol pembuatan catatan

Sistem information management yang efektif dapat membantu organisasi mengontrol pembuatan dan pertumbuhan arsip. Tanpa strategi yang pasti untuk membuat dan merekam informasi, tempat kerja dapat menghasilkan catatan kertas dan tanpa kertas yang berlebihan.

Hal ini dapat meningkatkan waktu untuk mengambil catatan dan meningkatkan biaya pengelolaan sumber daya informasi. Untuk mencegah hal ini, protokol manajemen informasi menetapkan batasan untuk membuat dan menghancurkan informasi untuk meningkatkan produktivitas dan efisiensi.

2. Memastikan kepatuhan terhadap peraturan

Banyak organisasi harus bekerja dalam peraturan tentang bagaimana mereka menangani data klien dan

bisnis. Sistem information management yang efektif memberikan pedoman yang menegakkan kepatuhan terhadap undang-undang dan peraturan, yang memungkinkan perusahaan untuk menghindari hukuman hukum dan keuangan yang dapat diakibatkan oleh pelanggaran yang tidak disengaja.

3. Mengurangi biaya operasi

Suatu organisasi membutuhkan sistem manajemen informasi yang efisien untuk mengurangi biaya pencatatan. Pengumpulan data, analisis, penyimpanan informasi, berbagi dan penghancuran adalah kegiatan padat modal, terutama untuk organisasi besar. Manajemen Informasi memprioritaskan catatan yang paling penting, mengurangi biaya di seluruh siklus hidup informasi.

4. Mengadopsi teknologi baru

Manajemen informasi menyediakan kemampuan untuk mengadopsi teknologi yang lebih baru dan lebih efisien untuk mengelola informasi. Ini bisa berupa otomatisasi, solusi perusahaan, kecerdasan buatan, atau produk atau layanan teknologi apa pun yang akan membantu perusahaan memperoleh lebih banyak manfaat dari informasinya.

5. Meningkatkan produktivitas dan efisiensi

Sistem information management yang hebat dapat meningkatkan cara karyawan menyimpan dan mengambil informasi yang diperlukan selama aktivitas sehari-hari mereka. Ini juga dapat mempermudah penyebaran informasi ke berbagai penerima melalui berbagai saluran, memungkinkan tim untuk

berkolaborasi dan berkomunikasi dengan mudah melintasi zona waktu dan lokasi. Sistem manajemen informasi yang efektif dapat membantu organisasi mengekstrak wawasan yang dapat ditindaklanjuti dari catatannya untuk memandu pengambilan keputusan.

6. Mengurangi risiko

Fungsi penting dari manajemen informasi adalah untuk mengurangi risiko hukuman hukum dan keuangan terhadap organisasi. Ini mencapai ini dengan protokol yang terdefinisi dengan baik untuk merekam, menyimpan, menyebarkan dan menghancurkan data. Ini mengurangi kemungkinan pelanggaran dan meningkatkan kepatuhan terhadap peraturan.

7. Melindungi informasi hak milik dan mempertahankan memori perusahaan

Organisasi membutuhkan proses untuk melindungi informasi penting mereka dari pesaing dan akses yang tidak sah. Manajemen informasi menyediakan sistem untuk melindungi informasi hak milik dari penyusup, kegagalan sistem, dan bencana alam. Ini membantu melindungi kerahasiaan dan integritas aset informasi penting, memungkinkan pemilik memperoleh manfaat maksimal dari rahasia dagang mereka.

8. Pertahankan dan tingkatkan

Evaluasi akan menunjukkan bagaimana meningkatkan efektivitas sistem dan juga memberikan kesempatan untuk meningkatkan infrastruktur dan melatih kembali staf. Perbaikan terus-menerus dapat memberikan kontribusi positif bagi kemampuan perusahaan untuk mencapai tujuan jangka pendek dan jangka panjang.



Sistem Bilangan

A. Pengertian Sistem Bilangan

Sistem bilangan adalah sebuah simbol atau kumpulan dari simbol yang merepresentasikan sebuah bilangan. Sistem bilangan yang banyak dipergunakan oleh manusia adalah sistem bilangan desimal, yaitu sistem bilangan yang menggunakan 10 macam simbol untuk mewakili suatu besaran. Sistem sering digunakan karena memiliki kesamaan dengan perhitungan manusia mempunyai yaitu sistem sepuluh jari sebagai alat membantu perhitungan. Berbeda halnya dengan cara kerja komputer, dimana logika di komputer diwakili oleh bentuk elemen dengan dua keadaan yaitu tidak ada arus dengan status *off* dan ada arus dengan status *on*. Konsep ini digunakan dalam sistem bilangan yang mempunyai dua jenis nilai yang mewakili suatu besaran nilai. Selain sistem bilangan biner, komputer juga menggunakan sistem bilangan oktal dan heksadesimal.

Sistem bilangan merupakan sekumpulan simbol khusus yang dipakai dalam membangun sebuah bilangan. Sistem bilangan yang umum dipakai manusia adalah desimal yang

terdiri dari sepuluh simbol yaitu 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 [1]. Sistem bilangan desimal biasanya disebut sistem bilangan berbasis sepuluh karena jumlah bilangan desimal adalah sepuluh digit. Penulisan basis sistem bilangan biasanya diakhiri dengan angka yang diperkecil/*subscript*, contoh sistem bilangan desimal tidak dituliskan : 200_{10} .

B. Jenis Sistem Bilangan

Sistem bilangan yang digunakan di dalam komputer adalah :

1. Sistem Bilangan Biner.
2. Sistem Bilangan Oktal.
3. Sistem Bilangan Desimal.
4. Sistem Bilangan Heksadesimal

1. Sistem Bilangan Desimal (*Decimal*)

Sistem ini menggunakan 10 macam simbol yaitu 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Sistem ini menggunakan basis 10. Bentuk nilai ini dapat berupa *integer* desimal atau pecahan.

Integer decimal adalah nilai desimal yang bulat, misalnya 8598 dapat diartikan :

$$8 \times 10^3 = 8000$$

$$5 \times 10^2 = 500$$

$$9 \times 10^1 = 90$$

$$8 \times 10^0 = 8 +$$

8598	<i>Positif value</i>
	<i>Absolute value</i>

Absolute value merupakan nilai untuk masing-masing digit bilangan, sedangkan *position value* adalah merupakan penimbang atau bobot dari masing-masing digit tergantung dari letak posisinya, yaitu bernilai basis dipangkatkan dengan urutan posisinya.

Pecahan desimal adalah nilai desimal yang mengandung nilai pecahan dibelakang koma, misalnya nilai 183,75 adalah pecahan desimal yang dapat diartikan :

$$\begin{array}{rcl}
 1 \times 10^2 & = & 100 \\
 8 \times 10^1 & = & 80 \\
 3 \times 10^0 & = & 3 \\
 7 \times 10^{-1} & = & 0,7 \\
 \hline
 5 \times 10^{-2} & = & 0,05 + \\
 & & 183,75
 \end{array}$$

2. Sistem Bilangan Biner (*Binary*)

Sistem bilangan *binary* menggunakan dua macam simbol bilangan berbasis dua *digit* angka, yaitu 0 dan 1. Sistem biner menggunakan dua simbol khusus, yaitu 0 dan 1. Disebut juga sistem bilangan berbasis dua. Bilangan biner adalah bilangan dasar yang digunakan dalam sistem komputer digital. Penulisan bilangan *biner* didalam komputer yaitu dengan mengelompokkan empat digit bilangan biner, misalnya : 1010 0001. Contoh bilangan *biner* dan bilangan desimal.

- a. $0010_2 = 2_{10}$
- b. $1010_2 = 10_{10}$

Bilangan *biner* 1100 sama dengan 12 bilangan desimal.

$$\begin{array}{r}
 1 \ 1 \ 0 \ 0 \\
 | \quad | \quad | \quad | \\
 0 \times 2^0 = 0 \\
 0 \times 2^1 = 0 \\
 1 \times 2^2 = 4 \\
 \underline{1 \times 2^3 = 8 +} \\
 = 12_{10}
 \end{array}$$

3. Sistem Bilangan Oktal (*Octal*)

Sistem ini menggunakan delapan simbol khusus, yaitu 0, 1, 2, 3, 4, 5, 6, 7. Disebut juga sistem bilangan berbasis delapan. Nilai positif atau *position value* pada sistem bilangan *octal* adalah perpangkatan dari nilai delapan. Contoh bilangan *octal* dan bilangan desimal:

- a. $2_8 = 2_{10}$
- b. $10_8 = 8_{10}$

Bilangan oktal 12 sama dengan 10 bilangan desimal.

$$\begin{array}{r}
 1 \ 2 \\
 | \quad | \\
 2 \times 8^0 = 2 \\
 \underline{1 \times 8^1 = 8 +} \\
 = 10_{10}
 \end{array}$$

4. Sistem Bilangan Heksadesimal (*Hexadecimal*)

Sistem bilangan heksadesimal terdiri dari 16 digit terdiri dari simbol bilangan berbasis 10 digit angka, yaitu 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, dan F dimana A = 10, B = 11, C = 12, D = 13, E = 14 dan F = 15 *position value* sistem bilangan heksadesimal adalah perpangkatan dari

nilai 16. Contoh bilangan heksadesimal dan bilangan desimal :

- a. $8_{16} = 2_{10}$
- b. $A_{16} = 10_{10}$
- c. $1A_{16} = 26_{10}$

Bilangan oktal C7 sama dengan 199 bilangan desimal.

$$C7_{16} = 199_{10}$$

	$\begin{array}{r} 7 \times 16^0 = 7 \\ C(12) \times 16^1 = 192 + \\ \hline 199 \end{array}$
--	---

C. Konversi Sistem Bilangan

Konversi sistem bilangan adalah sebuah proses dimana satu sistem bilangan dengan basis tertentu akan diubah menjadi bilangan dengan basis yang lain [2]. Konversi bilangan mengenal istilah basis yang diambil dari bilangan itu sendiri. Adapun basis konversi bilangan antara lain : bilangan biner menggunakan basis dua, bilangan desimal menggunakan basis sepuluh, bilangan oktal menggunakan basis delapan dan bilangan heksadesimal menggunakan basis enam belas.

1. Konversi Sistem Bilangan Desimal (*Decimal*)

- a. Mengkonversi bilangan desimal ke *biner*

Dalam mengubah angka desimal menjadi angka *biner* dipergunakan sebuah metode pembagian dengan angka 2 dengan memperhatikan sisanya yaitu 0 atau 1.

Sedangkan untuk menuliskan hasil konversinya adalah dari bawah keatas.

$$205_{10} = 1001101_2$$

205 : 2	= 102	sisa	1
102 : 2	= 51	sisa	0
51 : 2	= 25	sisa	1
25 : 2	= 12	sisa	1
12 : 2	= 6	sisa	0
6 : 2	= 3	sisa	0
3 : 2	= 1	sisa	1

sebagai sisa akhir 1, maka penulisan notasi binernya adalah 1001101.

b. Mengkonversi bilangan desimal ke *octal*

Dalam mengubah angka desimal menjadi bilangan oktal dipergunakan sebuah metode pembagian dengan angka 8 dengan memperhatikan sisanya. Sedangkan untuk menuliskan hasil konversinya adalah dari bawah keatas.

$$237_{10} = 355_8$$

237 : 8	= 29	sisa	5
29 : 8	= 3	sisa	5

3 sebagai sisa akhir 3, maka penulisan hasil konversi bilangan oktalnya adalah 325.

c. Mengkonversi bilangan decimal ke heksadesimal

Dalam mengubah angka desimal menjadi bilangan heksadesimal dipergunakan sebuah metode pembagian dengan angka 16 dengan memperhatikan sisanya. Sedangkan untuk menuliskan hasil konversinya adalah dari bawah keatas.

$$2839_{10} = 117_{16}$$

$$2839 : 16 = 177 \text{ sisa } 7$$

$$177 : 16 = 11 \text{ sisa } 1$$

1 sebagai sisa akhir 1, maka penulisan hasil konversi bilangan heksadesimalnya adalah 117.

2. Konversi Sistem Bilangan Binari (*Biner*)

a. Mengonversi bilangan *biner* ke desimal

Cara mengkonversi bilangan *biner* ke desimal menggunakan tabel konversi pada Tabel 5.1 konversi biner ke desimal berikut ini.

Tabel 5.1 Konversi Biner ke Desimal

Pangkat		2^4	2^3	2^2	2^1	2^0
Nilai Pangkat		16	8	4	2	1
<i>Biner</i>	11011	1	1	0	1	1
Desimal	27	16	8	0	2	1

Contoh bilangan *biner* 11011 yang dikonversi ke bilangan decimal adalah 27.

$$11011_2 = 27_{10}$$

Langkah mengkonversi menggunakan tabel konversi *biner* ke desimal.

- 1) Masukkan angka biner kedalam tabel konversi.
 - 2) Kalikan nilai pangkat dengan angka *biner* dan masukkan hasilnya ke kolom desimal.
 - 3) Jumlahkan angka yang ada pada kolom desimal.
- b. Mengonversi bilangan *Biner* ke *octal*

Cara mengkonversi bilangan *biner* ke desimal menggunakan tabel konversi pada Tabel 5.1 konversi biner ke octal berikut ini.

Tabel 5.2 Konversi *biner* ke *octal*

Pangkat		2^2	2^1	2^0	2^2	2^1	2^0
Nilai Pangkat		4	2	1	4	2	1
Biner	101011	1	0	1	0	1	1
Desimal		4	0	1	0	2	1
Oktal	53	5			3		

Langkah mengkonversi menggunakan tabel konversi *biner* ke oktal.

- 1) Buatlah pengelompokan pada table konversi, dimana dalam 1 kelompok terdiri dari 3 digit biner.
- 2) Masukkan angka biner yang sudah dikelompokkan kedalam table konversi.
- 3) Kalikan nilai pangkat dengan angka *biner* dan masukkan hasilnya ke kolom desimal.

- 4) Jumlahkan angka yang ada pada masing-masing kolom desimal.
 - 5) Tuliskan bilangan oktal sesuai dengan digit yang ada pada kelompoknya.
- c. Mengonversi bilangan Biner ke heksadesimal

Cara mengkonversi bilangan *biner* ke desimal menggunakan tabel konversi pada Tabel 5.3 konversi biner ke heksadesimal berikut ini.

Tabel 5.3 Konversi *biner* ke *octal*

Pangkat	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
Nilai	8	4	2	1	8	4	2	1
Biner	11010101	1	1	0	1	0	1	0
Desimal		8	4	0	1	0	4	0
Heksa desimal	D5	13=D				5		

Langkah mengkonversi menggunakan tabel konversi *biner* ke heksadesimal.

- 1) Buatlah pengelompokan pada Tabel 5.3 konversi, dimana dalam 1 kelompok terdiri dari 4 digit biner.
- 2) Masukkan angka *biner* yang sudah dikelompokkan kedalam tabel konversi.
- 3) Kalikan nilai pangkat dengan angka *biner* dan masukkan hasilnya ke kolom desimal.
- 4) Jumlahkan angka yang ada pada masing-masing kolom desimal.

- 5) Tuliskan bilangan heksadesimal sesuai dengan digit yang ada pada kelompoknya.
- 3. Konversi Sistem Bilangan Oktal (*Octal*)**

a. Mengonversi bilangan *octal* ke *decimal*

Cara mengkonversi bilangan *octal* ke desimal menggunakan tabel konversi pada Tabel 5.4 konversi *octal* ke *decimal* berikut ini.

Tabel 5.4 Konversi Bilangan Oktal

Pangkat		8^3	8^2	8^1	8^0
Nilai Pangkat		512	64	8	1
Oktal	1501	1	5	0	1
Desimal	833	512	128	0	1

Berikut ini contoh konversi bilangan *octal* 1501_8 di konversi menjadi bilangan desimal 833_{10} .

Langkah mengkonversi menggunakan Tabel 5.4 konversi *octal* ke desimal.

- 1) Masukkan bilangan *octal* kedalam Tabel 5.4 konversi.
- 2) Kalikan nilai pangkat dengan angka desimal dan masukkan hasilnya ke kolom desimal.
- 3) Jumlahkan angka yang ada pada masing-masing kolom desimal.

4. Konversi Sistem Bilangan Heksadesimal (*Hexadecimal*)

- a. Mengkonversi bilangan *hexadesimal* ke *decimal*

Cara mengkonversi bilangan *octal* ke desimal menggunakan tabel konversi pada Tabel 5.5 konversi *octal* ke *decimal* berikut ini.

Tabel 5.5 Konversi Bilangan Heksadesimal

Pangkat		16^2	16 1	16 0
Nilai		256	16	1
Heksadesimal	A1F	A	1	F
Desimal		10	1	15
Desimal	2591	2560	16	15

Berikut ini contoh konversi bilangan heksadesimal A1F₁₆ di konversi menjadi bilangan desimal 2591₁₀.

Langkah mengkonversi menggunakan tabel konversi heksadesimal ke desimal.

- 1) Masukkan bilangan heksadesimal kedalam Tabel 5.5 konversi bilangan heksadesimal.
- 2) Kalikan nilai pangkat dengan angka desimal dan masukkan hasilnya ke kolom desimal.
- 3) Jumlahkan angka yang ada pada masing-masing kolom desimal.



Pemahaman Bahasa Sistem

Pada bab ini, kita akan membahas dasar-dasar bahasa sistem yang sangat penting dalam konteks pengembangan sistem informasi. Memahami bahasa sistem adalah fondasi utama untuk mengkomunikasikan ide, spesifikasi, dan instruksi dengan jelas dan tepat di antara anggota tim pengembangan.

A. Pengertian Bahasa Sistem

1. Definisi Bahasa Sistem

Bahasa sistem adalah bahasa atau notasi yang digunakan untuk menggambarkan, merancang, dan berkomunikasi tentang sistem informasi. Ini mencakup penggunaan simbol, notasi, dan tata bahasa khusus yang memungkinkan pemangku kepentingan, seperti pengembang perangkat lunak, analis sistem, dan pemilik proyek, untuk berbicara dan berinteraksi secara efektif dalam konteks pengembangan sistem informasi. Bahasa sistem membantu dalam menyusun panduan, spesifikasi, dan dokumentasi yang diperlukan untuk memahami dan mengimplementasikan sistem informasi dengan benar (Pressman, 2015).

2. Pentingnya Memahami Bahasa Sistem

Memahami bahasa sistem memiliki dampak yang signifikan dalam pengembangan sistem informasi. Pemahaman yang kuat terhadap bahasa sistem memiliki beberapa implikasi penting, yaitu :

- a. **Komunikasi yang Jelas:** Bahasa sistem yang dipahami oleh semua anggota tim memungkinkan komunikasi yang jelas dan tepat. Ini mengurangi risiko kesalahpahaman dan kesalahan dalam pengembangan sistem.
- b. **Pengembangan yang Tepat Sasaran:** Bahasa sistem membantu dalam merinci persyaratan sistem dengan cara yang dapat diimplementasikan oleh tim pengembang. Ini menghindari kesenjangan antara apa yang diinginkan oleh pemilik proyek dan apa yang sebenarnya dihasilkan.
- c. **Dokumentasi yang Efektif:** Pemahaman bahasa sistem memungkinkan untuk menyusun dokumentasi sistem yang efektif dan berkelanjutan. Ini membantu dalam pemeliharaan sistem dan pelacakan perubahan.
- d. **Peningkatan Kolaborasi:** Anggota tim yang memiliki pemahaman yang kuat tentang bahasa sistem dapat berkolaborasi dengan lebih baik dan dengan mudah berbagi ide, pemikiran, dan spesifikasi (Sommerville, 2019).

B. Komponen Utama Bahasa Sistem

Dalam sub-bab ini, kita akan membahas dua komponen utama dari bahasa sistem, yaitu Simbol dan Notasi. Kedua elemen ini membentuk fondasi dari bahasa sistem, memungkinkan komunikasi yang tepat dalam konteks pengembangan sistem informasi.

1. Simbol dan Notasi

Simbol dan notasi adalah elemen penting dalam bahasa sistem yang memungkinkan kita untuk mengungkapkan ide, spesifikasi, dan instruksi dengan jelas. Di bawah ini akan dijelaskan berbagai simbol matematika yang umum digunakan dan notasi khusus dalam konteks sistem informasi (Kurnia, 2023).

a. Simbol Matematika Umum

Simbol matematika adalah representasi grafis atau simbolis dari konsep matematika atau logika. Beberapa simbol matematika umum yang digunakan dalam bahasa sistem meliputi :

- 1) $+, -, *, /$: Simbol untuk operasi matematika dasar seperti penjumlahan, pengurangan, perkalian, dan pembagian.
- 2) $=, \neq, >, <, \geq, \leq$: Simbol untuk mengekspresikan hubungan dan perbandingan antara angka atau variabel.
- 3) Σ, Π : Simbol untuk penjumlahan dan perkalian berulang.

- 4) $\sqrt{ }, ^n, \log$: Simbol untuk operasi matematika khusus seperti akar kuadrat, pangkat, dan logaritma.
- b. Notasi khusus dalam konteks sistem informasi

Notasi khusus adalah simbol atau konvensi yang digunakan secara khusus dalam bahasa sistem untuk menggambarkan elemen-elemen sistem informasi (Harahap, 2022). Contoh notasi khusus termasuk :

- 1) DFD (*Data Flow Diagram*): Diagram yang menggambarkan aliran data di dalam sistem.
- 2) ERD (*Entity-Relationship Diagram*): Diagram yang menggambarkan hubungan antara entitas dalam basis data.
- 3) UML (*Unified Modeling Language*): Bahasa visual untuk mendokumentasikan, merancang, dan memodelkan sistem perangkat lunak.

2. Struktur Kalimat dan Gramatika

Tata bahasa adalah aturan yang mengatur cara kata-kata dan simbol disusun dalam kalimat. Dalam konteks bahasa sistem, kita akan membahas tata bahasa dasar dan tata bahasa khusus yang digunakan dalam spesifikasi sistem informasi.

a. Tata Bahasa Dasar

Tata bahasa dasar mencakup aturan-aturan umum dalam membentuk kalimat, termasuk penggunaan subjek, predikat, objek, dan elemen-elemen bahasa lainnya. Dalam bahasa sistem, ini berarti menulis instruksi dan spesifikasi dengan cara yang dapat

dipahami dan diinterpretasikan oleh semua anggota tim pengembangan.

Contoh Tata Bahasa Dasar:

"Sebuah sistem informasi harus dapat mengelola data pengguna."

Dalam kalimat ini, terdapat subjek ("sebuah sistem informasi"), predikat ("harus dapat mengelola"), dan objek ("data pengguna"). Tata bahasa dasar memastikan kalimat tersebut memiliki struktur yang benar dan mudah dimengerti.

b. Tata Bahasa Khusus dalam Sistem Informasi

Dalam bahasa sistem, terdapat tata bahasa khusus yang digunakan untuk menggambarkan elemen-elemen sistem informasi seperti entitas, atribut, relasi, dan lain sebagainya. Misalnya, dalam Entity-Relationship Diagram (ERD), terdapat aturan khusus untuk menghubungkan entitas dan menentukan sifat-sifatnya.

Contoh Tata Bahasa Khusus :

Dalam ERD, kita menggunakan simbol seperti kotak untuk entitas, garis untuk hubungan, dan elips untuk atribut. Aturan-aturan ini membantu dalam menggambarkan struktur dan hubungan data dengan jelas.

3. Semantik dan Makna

Pemahaman bahasa sistem tidak hanya terbatas pada tata bahasa dan struktur kalimat, tetapi juga mencakup interpretasi makna yang terkandung dalam teks. Sub-bab ini membahas peran penting semantik dalam memastikan bahwa spesifikasi sistem diartikan dengan benar oleh semua anggota tim.

a. Interpretasi Makna dalam Bahasa Sistem

Interpretasi makna adalah kemampuan untuk memahami pesan atau instruksi yang terkandung dalam teks. Dalam konteks bahasa sistem, ini berarti memahami persyaratan, instruksi, dan spesifikasi dengan cara yang sesuai dengan tujuan sistem informasi yang dikembangkan.

Contoh Interpretasi Makna :

Jika sebuah spesifikasi menyatakan "Sistem harus mengirimkan notifikasi pada setiap perubahan status," maka pemahaman semantik memastikan bahwa setiap perubahan status harus menyebabkan sistem mengirimkan notifikasi kepada pihak terkait.

b. Peran Semantik dalam Kejelasan Spesifikasi

Ketika semantik dipahami dengan benar, maka spesifikasi sistem akan lebih jelas dan terhindar dari penafsiran yang salah. Hal ini mengurangi risiko terjadinya kesalahan implementasi atau pemahaman yang salah.

C. Bahasa Formal dan Informal dalam Sistem Informasi

Bahasa formal dan informal adalah dua jenis bahasa yang dapat digunakan dalam konteks Sistem Informasi. Masing-masing memiliki karakteristik dan kegunaannya sendiri (Wijaya, 2021).

1. Perbedaan Antara Bahasa Formal dan Informal

Bahasa formal adalah jenis bahasa yang memiliki aturan dan struktur yang ketat. Biasanya, bahasa formal digunakan dalam situasi-situasi resmi atau profesional di mana kejelasan dan presisi sangat penting. Contohnya

termasuk laporan resmi, dokumen kontrak, atau presentasi akademis. Bahasa formal cenderung menghindari penggunaan ungkapan atau frasa informal, dan fokus pada penyampaian informasi dengan jelas dan tepat.

Di sisi lain, bahasa informal lebih santai dan kurang terikat pada aturan baku. Biasanya, bahasa informal digunakan dalam situasi yang lebih akrab dan tidak resmi, seperti percakapan sehari-hari antar kolega atau komunikasi informal melalui media sosial. Bahasa informal dapat mencakup ungkapan atau kata-kata yang mungkin tidak cocok untuk penggunaan dalam konteks formal.

2. Kelebihan dan Kekurangan Bahasa Formal dan Informal

Setiap jenis bahasa memiliki kelebihan dan kekurangannya masing-masing. Bahasa formal, dengan ketatnya aturan dan kejelasannya, sangat cocok untuk komunikasi yang membutuhkan tingkat presisi yang tinggi. Namun, terkadang penggunaannya dapat terasa kaku atau kurang akrab, terutama dalam situasi yang membutuhkan interaksi yang lebih santai.

Di sisi lain, bahasa informal dapat membantu menciptakan atmosfer yang lebih akrab dan ramah. Ini sangat berguna dalam situasi di mana keterlibatan emosional atau hubungan personal memiliki peran penting. Namun, bahasa informal dapat menjadi kurang tepat atau kurang jelas dalam konteks formal atau profesional.

D. Peran Bahasa Sistem dalam Pengembangan Sistem Informasi

Dalam konteks pengembangan Sistem Informasi, bahasa sistem memegang peran yang penting dalam beberapa aspek kunci.

1. Pemilihan Bahasa dalam Spesifikasi Sistem

Pemilihan bahasa untuk spesifikasi sistem adalah langkah krusial dalam pengembangan sistem informasi. Faktor-faktor seperti jenis sistem yang akan dikembangkan, kebutuhan pengguna, dan kemampuan tim pengembang harus dipertimbangkan dengan cermat. Pemilihan bahasa yang tepat dapat mempengaruhi kinerja sistem, kecepatan pengembangan, dan kemudahan pemeliharaan. Referensi yang relevan dalam bahasa Indonesia dapat ditemukan dalam literatur mengenai pemilihan bahasa pemrograman dalam pengembangan perangkat lunak (Pressman, 2015).

2. Dokumentasi dengan Bahasa Sistem

Bagian ini membahas cara efektif mendokumentasikan sistem informasi menggunakan bahasa sistem. Dokumentasi yang baik memainkan peran kunci dalam memastikan pemahaman yang tepat oleh semua anggota tim. Dokumen harus disusun dengan jelas dan dapat diakses dengan mudah. Referensi yang dapat mendukung bagian ini termasuk buku-buku atau artikel mengenai praktik dokumentasi yang baik dalam pengembangan perangkat lunak (Sommerville, 2019).

3. Komunikasi Efektif dalam Tim Pengembangan

Bahasa sistem juga memainkan peran penting dalam komunikasi di antara anggota tim pengembangan. Komunikasi yang jelas dan efektif adalah kunci

kesuksesan proyek. Bagian ini akan memberikan tips dan praktik terbaik untuk memastikan bahwa semua orang memahami persyaratan dan instruksi dengan jelas. Referensi yang relevan untuk bagian ini dapat mencakup buku-buku atau artikel tentang manajemen proyek dan komunikasi tim yang efektif (Schwalbe, 2018).

E. Studi Kasus

Studi Kasus: Sistem Informasi Manajemen Inventaris Toko "Jaya Baru"

Dalam rangka mengilustrasikan pentingnya pemahaman bahasa sistem dalam pengembangan Sistem Informasi, kita akan mengeksplorasi studi kasus pada proyek pengembangan Sistem Informasi Manajemen Inventaris untuk toko retail "Jaya Baru". Proyek ini dipaparkan dalam buku "Konsep Sistem Informasi" pada Bab "Pemahaman Bahasa Sistem".

1. Deskripsi Proyek

"Toko Jaya Baru" merupakan sebuah toko retail yang menyediakan berbagai produk elektronik, mulai dari perangkat telekomunikasi hingga elektronik rumah tangga. Dalam rangka meningkatkan efisiensi pengelolaan inventaris, pemilik toko, Bapak Agus, memutuskan untuk mengembangkan Sistem Informasi Manajemen Inventaris yang terintegrasi.

2. Persyaratan Bahasa Sistem

Dalam fase perencanaan proyek, tim pengembang Sistem Informasi harus mengidentifikasi dan mendefinisikan persyaratan bahasa sistem dengan jelas. Persyaratan ini meliputi kemampuan untuk mencatat dan memonitor stok produk, mengelola pemesanan,

memperbarui harga, dan menghasilkan laporan penjualan secara berkala.

3. Implementasi dalam Konteks Situasi Nyata

Tim pengembang mulai membangun sistem dengan memilih bahasa pemrograman yang sesuai dengan kebutuhan proyek. Mereka memastikan bahwa sistem mampu berinteraksi dengan basis data inventaris toko secara efektif. Setelah pengujian menyeluruh, Sistem Informasi Manajemen Inventaris diintegrasikan ke dalam operasi sehari-hari "Toko Jaya Baru".

Hasilnya, Bapak Agus dapat mengamati peningkatan signifikan dalam efisiensi manajemen inventaris. Proses pemesanan, pemantauan stok, dan pengelolaan harga menjadi lebih lancar dan terotomatisasi. Ini membuktikan bagaimana pemahaman bahasa sistem dapat memberikan dampak yang nyata dalam meningkatkan kinerja operasional suatu bisnis.

Studi kasus ini adalah contoh konkret tentang bagaimana teori bahasa sistem diterapkan dan menghasilkan manfaat yang signifikan dalam dunia nyata.



Basis Data

A. Konsep Dasar Basis Data

Pada era transformasi digital kebutuhan informasi semakin meningkat. Berbagai instansi dan organisasi berusaha untuk meningkatkan pelayanan dan kinerja dengan membuat berbagai aplikasi yang memudahkan pengguna untuk mengakses data sesuai dengan kebutuhan informasi serta hak akses yang diberikan. Untuk menghasilkan informasi yang baik dimana memiliki data yang benar, bisa berada di tangan orang tepat, dan bisa disajikan diwaktu yang cepat , salah satu komponen yang dibutuhkan sebuah sistem informasi adalah basis data. Secara umum Basis data merupakan sebuah pangkalan data yang menyimpan berbagai data yang telah disimpan secara terstruktur dan saling teringrasi sehingga dapat menghasilkan informasi sesuai kebutuhan pengguna.

Dalam kehidupan di zaman digital ini kita tak lepas dari basis data. Banyak aktifitas sehari hari saat ini menggunakan basis data. Beberapa contohnya adalah saat kita melakukan transaksi banking seperti mentransfer uang melalui aplikasi e-banking, atau mengambil uang di ATM, saat itu kita ter-

hubung dengan basis data Bank dengan akses dari rekening atau akun kita. . Saat kita mendaftar berobat ke rumah sakit kita sudah tercatat di basis data rekam medis rumah sakit hanya dengan memberikan nomor pasien . Waktu kita mendaftar sekolah kita mempunyai nomor induk siswa yang langsung teringrasi pada basis data penerimaan siswa. Seiring perkembangan teknologi informasi saat ini kita tak susah lagi membeli tiket, memesan hotel atau berbagai akomodasi untuk perjalanan kita karena sudah tersedia berbagai aplikasi yang membantu kita untuk menyelesaikan permasalahan, dan semua didukung oleh basis data.

B. Sistem berbasis File

Sebelum diperkenalkan sistem basis data digunakanlah sistem berbasis file dimana pada sistem ini informasi diambil dalam suatu file (Watt and Eng, 2012). Perusahaan membangun sejumlah aplikasi dengan sistem berbasis file dimana masing masing dirancang untuk dapat menghasilkan data, tetapi sistem ini memiliki kekurangan sebagai berikut :

1. Redudansi data, dimana terjadi pengulangan data karena masing masing aplikasi memiliki satu file yang terpisah.
2. Isolasi data, dimana adanya kesulitan untuk mengambil data yang sesuai karena disimpan diberbagai file
3. Masalah integritas data, tidak ada jaminan untuk memastikan bahwa data dalam basis data benar dan konsisten
4. Masalah keamanan, pada sistem berbasis file ada kendala dalam mengatur hak akses .
5. Masalah akses konkurensi yaitu masalah pengaturan yang memungkinkan banyak pengguna yang bisa mengakses file.

C. Definisi Basis Data

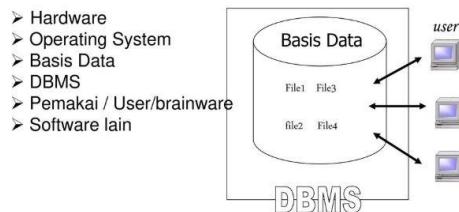
Secara umum Basis data adalah himpunan data yang disimpan dalam media penyimpanan elektronik yang telah disusun sedemikian rupa dan saling terintegrasi sehingga dapat menyajikan informasi sesuai dengan yang diinginkan.

Basis data dikelola oleh sebuah Data Base manajemen Sistem (DBMS) atau disebut sistem manajemen basis data. DBMS merupakan perangkat lunak untuk mengelola basis data, dimana dapat dilakukan operasi dasar basis data. Dalam pengembangan sebuah sistem informasi, program aplikasi dihubungkan dengan basis data yang dikelola oleh DBMS.

Pada saat ini berbagai instansi dan Perusahaan menggunakan sistem informasi untuk memudahkan pekerjaannya seperti perbankan, pemerintahan, rumah sakit, pariwisata, Pendidikan, organisasi Masyarakat dan lainnya. Untuk memperoleh informasi yang handal dan memudahkan pengelolaan data maka diperlukan basis data .

D. Komponen Sistem basis data

Komponen pendukung sistem basis data seperti terlihat pada gambar 1 terdiri dari 6 komponen utama yaitu perangkat keras, DBMS, sistem operasi, aplikasi , data, dan user. Perangkat keras seperti computer, perangkat telekomunikasi, merupakan komponen utama untuk mengelola basis. Sistem operasi, program aplikasi dan DBMS merupakan perangkat lunak yang bekerja untuk menunjang pemrosesan basi data. Sementara tanpa user dan data, basis data tidak dapat dioperasikan .

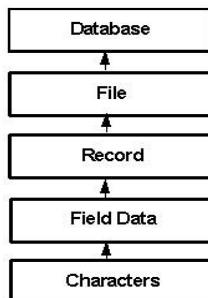


Gambar 1. Komponen sistem basis data

E. Jenjang Basis Data

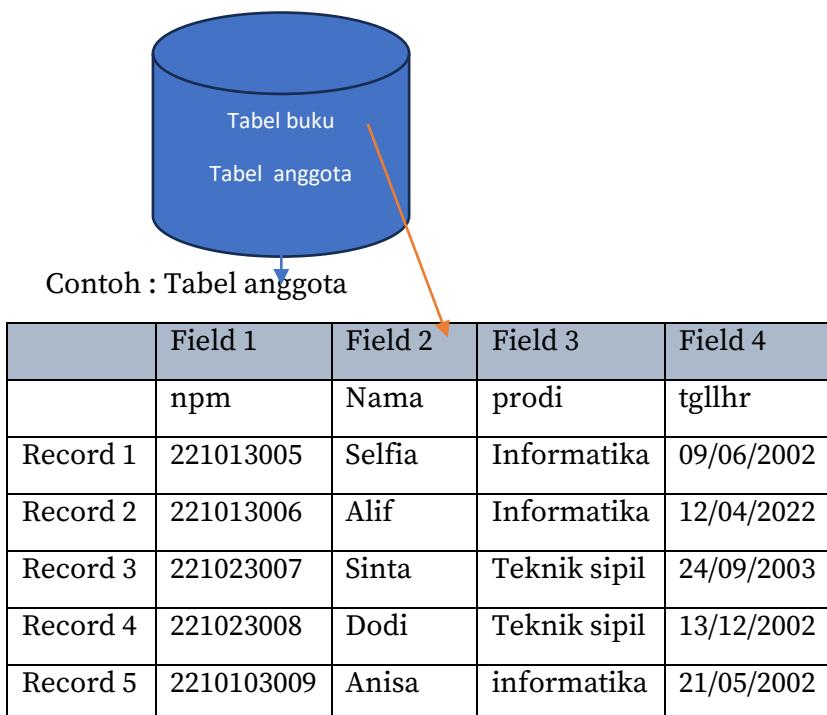
Jenjang basis data menggambarkan tingkatan basis data sehingga lebih mudah untuk dipahami . Jenjang basis data dapat dilihat pada gambar 2 . Jenjang basis data terdiri dari :

1. Karakter , bisa berupa huruf, angka atau simbol yang membentuk sebuah item data
2. Field , disebut sebagai atribut yang menunjukkan item dari data .
3. Record/ rekaman/tupel merupakan Kumpulan dari field yang membentuk suatu record atau rekaman.
4. File/tabel merupakan Kumpulan dari record yang menggambarkan satu kesatuan data yang sejenis .
5. Basis data merupakan Kumpulan file



Gambar 2. Jenjang Basis Data

Dari gambar 3 dapat kita lihat sebuah contoh basis data sederhana yakni basis data pustaka yang terdiri dari 4 (empat) tabel yaitu tabel buku, tabel anggota, tabel penerbit, tabel peminjaman . Untuk Tabel Anggota terdiri dari 4 buah *field* atau atribut yaitu *field* npm, nama, prodi, tglhr dan 5 buah record (baris). Masing masing field dirancang dengan menggunakan Batasan yang terdiri dari type data dan property pendukung lainnya sehingga data mudah di input dan ditampilkan.



Gambar 3. Contoh representasi tabel pada basis data .

F. Karakter dan Manfaat Basis Data

Untuk menyelesaikan permasalahan pengelolaan data pada sistem berbasis File, dikembangkanlah sistem basis data. Adapun karakter dan manfaat dari sistem basis data adalah sebagai berikut (Watt and Eng, 2012) :

1. Mendeskripsikan data .

Pemisahan data dan informasi membuat sistem ini berbeda dengan sistem berbasis file

2. Isolasi antara program dan data

Pada basis data struktur disimpan dalam katalog sistem dan bukan dalam program. Oleh karena itu, hanya satu perubahan yang diperlukan untuk mengubah struktur file. Isolasi antar program dan data ini disebut kemandirian program-data.

3. Mendukung banyak tampilan data.

Tampilan adalah bagian dari basis data yang didefinisikan dan didedikasikan untuk pengguna sistem.

4. Mendukung multi user

Sistem basis data dapat dirancang untuk banyak pengguna. Artinya, mereka mengizinkan banyak pengguna untuk mengakses basis data yang sama pada waktu yang sama.

5. Mengendalikan redundancy (kerangkapan) data.

Dalam beberapa kasus , redundansi data masih ada untuk meningkatkan kinerja sistem , namun sebaiknya tetap diminimumkan.

6. Berbagi Data

Basis data memberikan kemungkinan berbagi data sehingga pengguna dapat melakukan pertukaran data, dan akan menghasilkan lebih banyak informasi.

7. Penegakan batasan integritas

Batasan integritas basis data adalah batasan atau aturan yang menentukan apa yang dapat dimasukkan dalam sebuah field. Menentukan tipe data pada sebuah field berguna menentukan jenis data yang diperbolehkan dalam suatu field, misalnya data hanya bisa berbentuk angka atau karakter. Keunikan data seperti kunci utama memastikan tidak ada duplikat yang dimasukkan. Batasan bisa sederhana (berbasis lapangan) atau kompleks (pemrograman).

8. Pembatasan akses yang tidak sah

Tidak semua pengguna sistem database memiliki hak akses yang sama. Misalnya, nasabah bank hanya bisa melihat akun rekeningnya saja atau melakukan transaksi yang ditentukan .

9. Data *independent*.

Keuntungan lain dari sistem manajemen basis data adalah memungkinkan independensi data. Maksudnya adalah data terpisah dengan program aplikasi deskripsi data .

10. Pemrosesan transaksi

Pada basis data diharapkan data tetap konsisten dan valid selama pemrosesan transaksi meskipun beberapa pengguna sedang memperbarui data yang sama.

11. Fasilitas pencadangan dan pemulihan

Pencadangan dan pemulihan adalah metode yang memungkinkan untuk melindungi data dari kehilangan.

12. Penegakan batasan integritas

Pada basis data disediakan Batasan integritas untuk meminimalisir kesalahan data.

G. Operasi Dasar Basis Data

Dengan adanya basis data kita bisa mengelola bagaimana data itu di simpan, mulai dari pembuatan Basis data hingga menampilkan data yang diinginkan. Adapun operasi dasar Basis data adalah membuat basis data, menghapus basis data, membuat tabel, menghapus tabel, mengubah tabel, mengisi tabel, mengambil/menampilkan data serta menghapus data .

H. Bahasa Basis Data

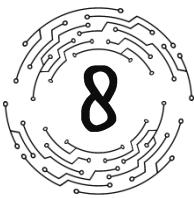
Untuk melakukan pendefinisian basis data dan juga memanipulasi basis data sistem basis data menyediakan bahasa basis data yang terdiri dari *Data Definition language* dan *data manipulation language* . Keduanya merupakan bagian dari Bahasa basis data Tunggal yang disebut SQL (*Struktur Query Language*) yang merupakan Bahasa basis data standar untuk basis data bertipe relasional. (Nugroho, 2004)

1. *Data Definition Language*

Data definition language (DDL) digunakan untuk mendefinisikan skema basis data . DDL mendefinisikan basis data, membuat struktur tabel sesuai dengan Batasan yang diinginkan. Serta melakukan penghapusan tabel.

2. *Data Manipulation Language* (DML)

DML adalah bahasa yang memungkinkan pengguna untuk melakukan pemanggilan informasi yang tersimpan pada basis data (query), menambah informasi baru pada basis data, penghapusan informasi dan juga memperbarui informasi.



Sistem Manajemen Basis Data

A. Pengertian Sistem Manajemen Basis Data

Dengan Perkembangan Teknologi yang cukup pesat pada saat ini sangat lah di perlukan adanya system managemen basis data, yang mengatur atau mengelola semua database dalam Perusahaan atau pun organisasi. Basis data sangat penting digunakan untuk aktivitas pemrosesan yang di dukung oleh system computer. Pengertian database management system menurut Connolly,T.M,et al (2010), database management system (DBMS) adalah system perangkat lunak yang memungkinkan pengguna untuk mendefinisikan , membuat, memelihara dan mengontrol akses ke database. Ada pula menurut Laudon, K.C, & Laudon, Jane P, (2012) database management system (DBMS) adalah perangkat lunak yang memungkinkan sebuah organisasi untuk memusatkan data, mengelola secara efisien dan memberikan akses ke data yang di simpan oleh program aplikasi.

Sistem Manajemen Basis Data atau Database Management Systems (DBMS) adalah suatu perangkat lunak yang mengorganisasikan , mengategorikan , menyimpan dan menampilkan serta memelihara data dalam basis data dengan menggunakan beberapa aplikasi perangkat program yang terdapat dalam database management system antara lain Microsoft SQL, MySQL, Oracle , Ms Access dan lain sebagainya . Sistem manajemen basis data lebih berfokus pada pengelolaan data dalam jumlah cukup besar, selain itu system ini dapat di gunakan sebagai perantara antar pengguna dan database.

Dalam penerapannya DBMS atau system manajemen basis data memanfaatkan database yang dapat di terjemahkan secara langsung menjadi suatu perintah. Terdapat 2 jenis Bahasa database diantara nya :

1. Definition Language (DDL) mencerminkan desain database secara menyeluruh agar dapat membuat tabel baru, memuat indeks tertentu dan seluruh kegiatan pengolahan database secara umum.
2. Data Manipulation Language (DML) berguna dalam memanipulasi , penghimpunan data yang terdapat didalam database .

Menurut Connolly,T.M,et al,(2010) secara khusus DBMS menyediakan fasilitas sebagai berikut :

1. Memungkinkan pengguna untuk menentukan database biasanya melalui Data Definition Language (DDL). DDL memungkinkan pengguna untuk dapat menentukan tipe data dan struktur dan kendala pada data yang akan disimpan dalam database.
2. Memungkinkan pengguna untuk melakukan operasi insert, update,mengedetele, dan mengambil data dari

database dan biasanya melalui Data Manipulation Language (DML).

3. Menyediakan control akses ke database yang meliputi
 - a. Sistem ke amanan, mencegah pengguna yang tidak sah mengakses database .
 - b. Sistem integritas, Mempertahankan konsistensi data yang di simpan
 - c. Sistem control Concurrency, memungkinkan akses Bersama database
 - d. Sistem control recovery , yang mengembalikan database ke keadaan yang konsisten sebelumnya setelah kegagalan hardware atau software
 - e. User-accessible catalog , berisi deskripsi dari data dalam database.

B. Komponen Sistem Manajemen Basis Data (DBMS)

Terdapat 5 komponen Utama pada Sistem Manajemen Basis Data

1. Hardware(Perangkat Keras)

Sistem Manajemen Basis Data (DBMS) dan aplikasi membutuhkan perangkat keras untuk dapat berjalan. Perangkat kerasnya dapat berupa satu personal computer, satu mainframe, maupun jaringan yang terdiri dari banyak computer. Perangkat keras yang dibutuhkan bergantung dari permintaan dari organisasi dan DBMS yang digunakan.

2. Software (Perangkat Lunak)

Komponen dari perangkat lunak terdiri dari perangkat lunak DBMS itu sendiri dan program aplikasi, Bersama dengan sistem aplikasi, termasuk perangkat lunak jaringan jika DBMS digunakan melalui jaringan.

3. Data

Mungkin komponen yang terpenting pada DBMS, terutama dari sudut pandang pengguna, adalah data. Data berperan sebagai jembatan antara komponen mesin (hardware dan software) dan komponen manusia (prosedur dan manusia). Database berisi baik data, maupun meta data, yaitu data tentang data. Struktur dari database disebut skema.

4. Prosedur

Menunjuk pada instruksi dan aturan yang mempengaruhi desain dan penggunaan dari database. Para pengguna sistem dan para staf yang mengatur dokumen prosedur database yang dibutuhkan dan bagaimana cara menggunakan atau menjalankan sistem.

- a. Bagaimana cara memasuki DBMS
- b. Bagaimana menggunakan fasilitas DBMS atau program aplikasi
- c. Bagaimana memulai dan mengakhiri DBMS
- d. Bagaimana membuat Salinan dari database
- e. Bagaimana mengatasi kegagalan perangkat keras dan perangkat lunak. Hal ini termasuk prosedur tentang bagaimana mengidentifikasi komponen yang gagal, bagaimana memperbaikinya, dan mengikuti perbaikan dari kesalahan, bagaimana memulihkan database

5. Manusia

Orang-orang yang berhubungan dengan sistem antaralain:

- a. Database Designer

Ada dua tipe dari database designer, yaitu :

- 1) Logical database designer, tugasnya berhubungan dengan mengidentifikasi data, relasi antar data,

dan Batasan pada data yang akan disimpan di database.

- 2) Physical database designer, bertugas untuk memutuskan bagaimana desain logical database direalisasikan. Hal ini termasuk memetakan desain logical database kepada seperangkat table dan Batasan integritas, memilih struktur penyimpanan yang spesifik dan metode akses data untuk mencapai performa yang baik, dan mendesain aturan keamanan yang dibutuhkan

b. Application Developers

Ketika database diimplementasikan, program aplikasi yang menyediakan fungsi yang dibutuhkan oleh pengguna harus diimplementasikan juga. Ini adalah tanggung jawab dari application developers. Biasanya, application developers bekerja dari spesifikasi yang diproduksi oleh system analys. Setiap program mengandung kalimat yang meminta DBMS untuk melakukan beberapa operasi pada data base. Hal ini termasuk mengambil data, memasukkan, mengubah, dan menghapus data.

c. *End-users*

Para pengguna dapat diklasifikasikan berdasarkan bagaimana mereka menggunakan sistem, yaitu :

1) Pengguna dasar

Pengguna dasar adalah pengguna yang terlatih menggunakan DBMS secara awam. Mereka mengakses database melalui program aplikasi tertulis khusus yang diusahakan untuk membuat operasi sesederhana mungkin. Mereka menggunakan operasi database dengan memasukan perintah sederhana atau memilih

pilihan dari menu. Hal ini berarti mereka tidak perlu mendalami topik khusus mengenai database atau DBMS.

2) Pengguna berpengalaman

Pengguna berpengalaman biasanya sudah mengenal struktur dari database dan asilitas yang ditawarkan oleh DBMS. Pengguna berpengalaman mungkin menggunakan Bahasa query yang dengan tingkat tinggi seperti SQL untuk melakukan operasi yang dibutuhkan.

C. Fungsi dan Pelayanan Sistem Manajemen Basis Data (DBMS)

1. **Data Storage Manajeman Penyimpanan Data** : DBMS Menciptakan struktur kompleks yang diperlukan untuk menyimpan data dalam database fisik. Ini menyediakan mekanisme untuk pengelolaan penyimpanan permanen data.
2. **Manajeman Transaksi** : Transaksi adalah serangkaian operasi Database, yang dilakukan oleh program aplikasi, yang mengakses atau mengubah isi Database. Oleh karena itu, DBMS harus menyediakan mekanisme untuk memastikan bahwa semua pembaruan yang terkait dengan transaksi dilakukan atau tidak ada satu pun yang dibuat.
3. **Layanan Integritas** : Integritas database menga pada kebenaran dan konsistensi data yang disimpan dan sangat penting dalam sistem database berorientasi transaksi. Oleh karena itu, DBMS menyediakan untuk memastikan bahwa baik data dalam database dan perubahan mengikuti aturan tertentu. Ini meminimalkan

redundansi data dan konsistensi data. Hubungan data yang disimpan dalam kamus data digunakan untuk menegakkan integritas data. Berbagai jenis mekanisme dan Batasan integritas dapat didukung untuk membantu memastikan bahwa nilai data dalam database valid, bahwa operasi yang dilakukan pada nilai tersebut valid dan database tetap dalam keadaan konsisten.

4. **Manajemen Pencadangan dan Pemulihan** : DBMS menyediakan mekanisme untuk berbagai jenis kegagalan. Ini mencegah hilangnya data. Mekanisme pemulihan DBMS, memastikan bahwa database dikembalikan ke keadaan yang konsisten setelah transaksi gagal atau dibatalkan karena sistem crash, kegagalan media, kesalahan perangkat keras atau perangkat lunak, kegagalan daya, dan sebagainya.
5. **Concurrency Control Service** : Karena DBMS mendukung berbagai data di antara banyak pengguna, mereka harus menyediakan mekanisme untuk mengelola akses bersamaan ke database.
6. **Data Manipulation Management** : DBMS melengkapi pengguna dengan kemampuan untuk mengambil, memperbarui dan menghapus data yang ada dalam database atau untuk menambahkan data baru ke database. Ini termasuk komponen prosesor DML untuk menangani Bahasa manipulasi data (DML).
7. **Data Dictionary/Manajeman Katalog Sistem**: DBMS menyediakan kamus data atau fungsi katalog sistem dimana deskripsi item data disimpan dan dapat diakses oleh pengguna. Katalog sistem atau kamus data adalah Database sistem, yang merupakan tempat penyimpanan informasi yang menjelaskan data dalam Database. Ini adalah data tentang data metadata.

Misalnya, DBMS akan berkonsultasi dengan katalog sistem untuk memverifikasi bahwa tabel yang diminta ada dan pengguna yang mengeluarkan permintaan memiliki hak akses yang diperlukan.

8. **Otorisasi/Manajemen Keamanan** : DBMS melindungi database terhadap akses yang tidak sah, baik di sengaja atau tidak disengaja. Ini melengkapi mekanisme untuk memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses database. Ini menciptakan sistem keamanan yang memberlakukan keamanan pengguna dan privasi data dalam database. Aturan keamanan menentukan pengguna mana yang dapat mengakses database, item data mana yang dapat diakses setiap pengguna, dan operasi data mana (menambah, menghapus dan memodifikasi) yang dapat dilakukan pengguna.
9. **Utility Service/Layanan Utilitas** : DBMS menyediakan satu set layanan utilitas yang digunakan oleh DBA dan perancang Database untuk membuat, mengimplementasikan, memantau, dan memelihara Database. Layanan utilitas ini membantu DBA untuk mengelola database secara efektif.
10. **Akses Database dan Antarmuka Pemrograman Aplikasi** : Semua DBMS menyediakan antarmuka untuk memungkinkan aplikasi menggunakan layanan DBMS. Mereka menyediakan akses data melalui Bahasa kueri terstruktur (SQL). Bahasa query DBMS berisi dua komponen:
 - a. Bahasa definisi data/ data definition language (DDL)
 - b. Bahasa manipulasi data/data manipulation language (DML)

Mendefinisikan struktur di mana disimpan dan DML memungkinkan pengguna akhir untuk mengestrak data dari database. DBMS juga menyediakan akses data ke pemrogram aplikasi melalui Bahasa procedural seperti C, C++, Java dan lain-lain.

11. **Data Independence Service/Layanan Independensi Data :** DBMS harus mendukung independensi program dari struktur database yang sebenarnya.
12. **Data Definition Service/Layanan Definisi Data :** DBMS menerima definisi data seperti skema eksternal, skema konseptual, skema internal, dan semua pemetaan terkait dalam bentuk sumber. Ini mengubahnya ke bentuk objek yang sesuai menggunakan komponen prosesor DDL untuk masing-masing dari berbagai Bahasa definisi data (DDL).

D. Kelebihan dan Kekurangan Dalam Penggunaan Sistem Manajemen Basis Data

1. Kelebihan dari Sistem manajemen basis data (DBMS)

Penyimpanan data dalam bentuk DBMS mempunyai banyak manfaat dan kelebihan di bandingkan dengan penyimpanan dalam bentuk flat file atau spreadsheet , diantaranya adalah :

- a. Performa yang dapat di dapat dari penyimpanan bentuk DBMS cukup besar , sangat berbeda dengan sistem flat file ,di samping itu memiliki unjuk kerja yang lebih baik ,serta lebih efisiensi dalam penggunaan media penyimpanan dan memori .
- b. Integritas data lebih terjamin dengan penggunaan DBMS, masalah redundansi (redundansi adalah

kejadian berulang nya data , yang akan mengakibatkan pemborosan) sering terjadi pada sistem flat file tetapi untuk DBMS itu tidak terjadi .

- c. Independensi perubahan struktur file database di mungkinkan terjadi tanpa harus mengubah aplikasi yang mengaksesnya sehingga pembuatan antarmuka (interface) kedalam data akan lebih mudah dengan penggunaan DBMS
 - d. Sentralisasi data yang terpusat akan mempermudah pengelolaan database, kemudahan yang di berikan untuk pengguna dalam hal melakukan bagi pakai dengan DBMS dan kekonsistenan data yang di akses secara Bersama-sama akan lebih terjamin dari pada data yang di simpan dalam bentuk file
 - e. Keamanan DBMS memiliki sistem keamanan yang lebih fleksibel dari pada pengamanan pada bentuk file ,sistem operasi keamanan dalam DBMS akan memberikan keluwesan dalam pemberian Hak
 - f. Layanan back up dan recovery data yang cukup baik
 - g. Memudahkan pengguna untuk membuat laporan dan report
2. **Kekurangan pada Sistem Manajemen Basis Data (DBMS)**
- Kekurangan pada sistem DBMS adalah sebagai berikut :
- a. Komplesitas, DBMS adalah suatu perangkat lunak yang cukup kompleks, yang di butuhkan atau keahlian tertentu untuk memanajemen dan di perlukan keahlian khusus untuk menjadi Database Administrator, Database developer,Database Designers , dan End-user yang harus mengerti fungsionalitasnya.

- b. Ukuran , karena besarnya kompleksitas pada DBMS bila membuat DBMS membutuhkan kapasitas penyimpanan yang besar agar dapat menjalankan aplikasi nya , jadi akan memerlukan storage baik eksternal mau pun internal untuk dapat bekerja secara cepat, tepat dan efisiensi .
- c. Biaya dalam DBMS , biaya pembuatan DBMS sangat bervariasi dan itu tergantung dari lingkungan dan fungsionalitas yang di inginkan . Harga database management system termasuk relative mahal
- d. Biaya tambahan dari Hardware. Membutuhkan biaya tambahan untuk kapasitas penyimpanan dan untuk tercapainya kinerja yang baik ,yang di inginkan sehingga membutuhkan mesin yang mempunyai kapasitas yang cukup besar
- e. Biaya konversi , Biaya konversi relative lebih kecil, biaya yang meliputi pelatihan atau training kepada staf untuk menggunakan sistem yang baru dan memungkinkan staf dengan keahlian khusus .

E. Jenis-Jenis Sistem Manajemen Basis Data (DBMS)

Terdapat 4 (empat) jenis sistem Manajemen Basis Data Diantara sebagai berikut yaitu :

1. Database Management system Hierarchical

Jenis yang pertama ini adalah Hierarchical DBMS yang datanya model nya tersusun dalam struktur seperti pohon, dimana datanya di simpan dalam format secara hierarchical atau topdown dan bottom up ,data di representasikan menggunakan hubungan parent-child. Dalam pengertiannya parent berarti banyak tetapi child hanya memiliki satu parent.

2. Network Model

Jenis network model ini adalah model DBMS jaringan yang memungkinkan setiap child atau anak memiliki banyak parent atau orang tua . Jenis model jaringan ini dapat membantu untuk mengatasi kebutuhan memodelkan hubungan yang lebih kompleks seperti hubungan data Many-to-Many , entitas di atur dalam grafik yang dapat di akses melalui beberapa jalur .

3. Relational DBMS

Relational DBMS merupakan model DBMS yang paling banyak digunakan karena paling mudah. Model ini didasarkan pada normalisasi data di bariskan dan kolom tabel, model relational ini di simpan dalam struktur tetap dan di manipulasi menggunakan SQL

4. Object-Oriented Model

Dalam object-oriented atau di kenal dengan model DBMS ber orientasi object .Model ini data di simpan dalam bentuk object, ada struktur yang di sebut kelas yang menampilkan data didalamnya. Ini didefinisikan database sebagai Kumpulan object yang menyimpan nilai-nilai anggota data dan operasinya.

F. Macam-Macam Aplikasi atau Software Database Management system (DBMS)

Adapun beberapa macam software dari DBMS sebagai berikut :

1. MySQL

Software atau perangkat lunak DBMS ini merupakan perangkat lunak yang paling banyak di gunakan karena tersedia open source atau gratis, sehingga perangkat

lunak MY SQL ini cocok di gunakan oleh berbagai pengguna .

2. ORACLE

ORACLE merupakan perangkat lunak DBMS yang mempunyai beragam macam fitur yang dapat memenuhi tuntutan kebutuhan , fleksibilitas dan di gunakan oleh Perusahaan besar , perangkat lunak ini berbayar.

3. MICROSOFT SQL SERVER

Perangkat lunak DBMS ini untuk di terapkan dalam sistem Perusahaan-perusahaan yang cukup besar ,karena akan terhubung dan mempunyai kemampuan mengelola data dengan jumlah cukup .

4. FIREBRID

Perangkat lunak DBMS ini merupakan sistem manajemen basis data relational. Perangkat lunak yang menawarkan fitur yang sesuai dengan standar dari SQL 2003 dan ANSI SQL-99, FIREBRID dapat bekerja dengan menggunakan OS WINDOWS dan LINUX.

G. Peranan Sistem Manajemen Basis Data Dalam Perusahaan

Peranan sistem manajemen berbasis data secara metode dan prosedur kegiatan Perusahaan antara lain :

1. Dilihat Metode Kerja

- a. Menciptakan metode kerja yang mengarah pada pencapaian tujuan yang efektif dan efisien.
- b. Menambahkan efisiensi kerja kantor
- c. Membantu manajemen dalam menilai pekerjaan kantor
- d. Mengadakan menghematan waktu dan biaya

- e. Memeriksa pengeluaran yang sifatnya memboroskan penggunaan pegawai dan catatan-catatan yang tidak perlu .
- 2. **Dilihat dari prosedur kerja**
 - a. Memudahkan pelaksanaan kegiatan sehingga berjalan dengan baik secara efisien dalam waktu dan tenaga
 - b. Memperlancar kegiatan kerja sehingga menghasilkan pekerjaan yang efektif

H. Hambatan Dan Tantangan Sistem Manajemen Basis Data (DBMS)

Hambatan dan tantangan DBMS saat ini sebagai berikut :

- 1. Memperoleh perangkat lunak dan perangkat keras yang Mahal
- 2. DBMS mainframe masih sangat mahal harga nya , DBMS berbasis computer mikro walaupun biayanya hanya ratusan dollar dapat merupakan pengeluaran yang besar bagi Perusahaan yang kecil .
- 3. Kurangnya se orang Ahli Database
- 4. DBMS untuk mengelolanya memerlukan pengetahuan khusus agar dapat memanfaatkan kemampuan secara penuh . Pengetahuan khusus ini paling baik di sediakan oleh para pengelola database (DBA)
- 5. Biaya pemrosesan data sangat tinggi.



Arsitektur Basis Data

Dalam era digital yang terus berkembang, sistem informasi telah menjadi fondasi bagi berbagai organisasi dan bisnis di seluruh dunia. Sistem ini tidak hanya memungkinkan pengumpulan, penyimpanan, dan pengambilan data yang efisien, tetapi juga memberikan landasan bagi pengambilan keputusan yang tepat waktu dan terinformasi. Untuk mendukung fungsi-fungsi kritis ini, arsitektur basis data menjadi elemen yang sangat penting dalam perancangan sistem informasi.

Arsitektur basis data merujuk pada struktur keseluruhan yang melibatkan perencanaan, pengorganisasian, dan pengelolaan basis data. Ini mencakup komponen seperti sistem manajemen basis data (DBMS), data dictionary, metode penyimpanan data, dan berbagai strategi pengelolaan data. Pengembangan dan pemahaman arsitektur basis data yang baik akan membantu organisasi untuk mengoptimalkan penggunaan data, meningkatkan efisiensi, dan memastikan keamanan informasi yang penting.

Dalam Bab ini, kita akan memperdalam pemahaman tentang peran DBMS dalam arsitektur basis data dan bagaimana DBMS dapat menjadi elemen kunci dalam pengembangan sistem informasi yang sukses. Kami juga akan menggali hubungan DBMS dengan komponen lain dalam arsitektur basis data serta

pentingnya pemahaman tentang keamanan data. Referensi ini akan menjadi panduan berharga bagi para profesional dan pengembang sistem informasi untuk memahami dan mengimplementasikan arsitektur basis data yang efisien dan aman.

A. Apa Itu Arsitektur Sistem Basis Data?

Arsitektur Sistem Basis Data mengacu pada desain dan organisasi keseluruhan dari Sistem Manajemen Basis Data (DBMS), termasuk komponen-komponen perangkat keras, perangkat lunak, data, dan personel yang terlibat. Arsitektur ini menentukan bagaimana data disimpan, diproses, dan diambil dari basis data. Hal ini mencakup desain fisik dan logis basis data, serta interaksi antara berbagai komponen dalam sistem.

Jenis-jenis arsitektur basis data bervariasi, seperti arsitektur satu tingkat (*single-tier*), dua tingkat (*two-tier*), tiga tingkat (*three-tier*), dan n tingkat (*n-tier*). Pemilihan arsitektur ini bergantung pada faktor-faktor seperti ukuran basis data, jumlah pengguna, jenis aplikasi yang akan mengakses basis data, serta tingkat keamanan dan skalabilitas yang diperlukan(Silberschatz et al., 2020).

Arsitektur basis data dapat dilihat sebagai model logis atau model fisik:

1. Model Logis: Model ini menggambarkan data dan hubungannya dalam basis data dalam bentuk entitas, atribut, dan relasi. Ini memberikan pandangan tingkat tinggi tentang data dan hubungan antara elemen data dan biasanya direpresentasikan dengan *Entity Relational Diagram* (ERD).

2. Model Fisik: Model ini menggambarkan desain fisik dan implementasi basis data. Ini mencakup informasi tentang struktur penyimpanan, indeks, dan jalur akses yang digunakan untuk menyimpan dan mengambil data. Model fisik juga mencakup detail tentang perangkat keras dan perangkat lunak yang digunakan untuk mendukung basis data.

Kedua model, baik logis maupun fisik, penting untuk dipertimbangkan saat merancang Sistem Manajemen Basis Data. Model logis digunakan untuk menentukan struktur data dan hubungan antara elemen data, sementara model fisik digunakan untuk menentukan bagaimana data akan disimpan dan diambil secara fisik.

Arsitektur basis data dalam RDBMS didasarkan pada model relasional, yang menggunakan tabel untuk menyimpan dan mengelola data. Arsitektur basis data RDBMS terdiri dari komponen-komponen berikut:

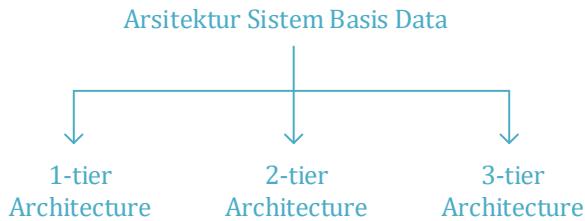
1. Basis Data: Kumpulan data terkait yang disimpan dalam tabel-tabel.
2. Tabel: Kumpulan catatan (baris) dan kolom (atribut) yang mewakili data dalam basis data.
3. Atribut: Kolom-kolom dalam sebuah tabel yang mewakili karakteristik atau properti tertentu dari data.
4. Catatan: Baris-baris dalam sebuah tabel yang mewakili elemen data individu, dan terdiri dari nilai-nilai untuk setiap atribut.
5. Kunci Primer: Setiap catatan dalam sebuah tabel memiliki identifikasi unik.

6. Kunci Asing: Kunci yang menghubungkan catatan dalam satu tabel dengan catatan dalam tabel lain, mewakili hubungan antar data.
7. Indeks: Struktur yang digunakan untuk mempercepat pengambilan data dengan memungkinkan basis data untuk dengan cepat menemukan catatan khusus dalam sebuah tabel.
8. Bahasa Kueri: Bahasa yang digunakan untuk mengambil data dari basis data, seperti SQL (*Structured Query Language*).
9. Manajemen Transaksi: Komponen yang memastikan konsistensi dan keandalan data dengan mengelola perubahan dalam basis data sebagai satu unit kerja tunggal.

Dalam arsitektur Sistem Manajemen Basis Data, pada RDBMS, basis data dan komponennya disimpan pada disk dan dikelola oleh perangkat lunak RDBMS, yang menyediakan fasilitas untuk menambah, mengubah, dan menghapus data, serta mencari dan mengambil data berdasarkan kondisi tertentu.

Jenis-jenis Arsitektur Sistem Manajemen Basis Data (DBMS):

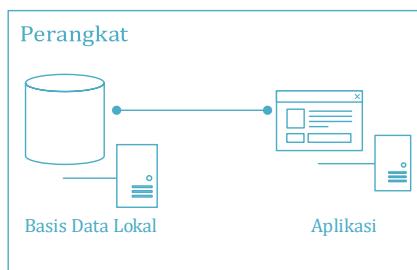
1. Arsitektur Satu Tingkat (*Single Tier Architecture*)
2. Arsitektur Dua Tingkat (*Two Tier Architecture*)
3. Arsitektur Tiga Tingkat (*Three Tier Architecture*)



B. Arsitektur Satu Tingkat (*1-Tier Architecture*)

Arsitektur DBMS satu tingkat adalah desain DBMS yang paling dasar. Semua komponen DBMS ditempatkan pada satu server atau platform tunggal, yang berarti bahwa basis data dapat diakses langsung oleh pengguna akhir. Karena koneksi langsung ini, DBMS memberikan respons cepat, itulah mengapa para programmer sering menggunakan desain ini untuk meningkatkan aplikasi lokal.

Setiap perubahan yang dilakukan oleh klien akan langsung tercermin dalam basis data dalam struktur ini, dan semua pemrosesan dilakukan pada satu server tunggal. Selain itu, tidak diperlukan koneksi jaringan untuk menjalankan tugas-tugas basis data. Nama lain untuk sistem manajemen basis data ini adalah sistem basis data lokal.



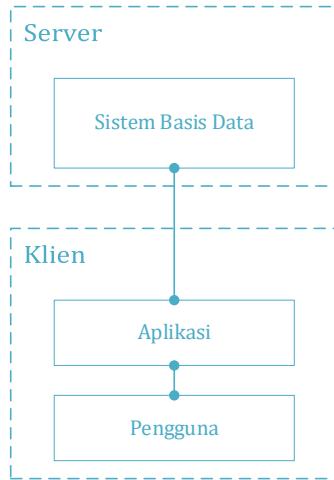
Arsitektur satu tingkat cocok digunakan ketika:

1. Informasi tidak sering diperbarui.

2. Sistem basis data tidak dikunjungi oleh banyak pengguna.
3. Kebutuhan yang sederhana dan mudah untuk mengedit atau mengakses basis data untuk pengembangan aplikasi.
4. Beberapa keuntungan dari arsitektur satu tingkat adalah:
5. Kesederhanaan: Arsitektur satu tingkat mudah untuk diatur dan dipelihara, karena semua komponen berada pada satu mesin.
6. Biaya Rendah: Arsitektur satu tingkat tidak memerlukan perangkat keras atau perangkat lunak tambahan, sehingga menjadi solusi yang hemat biaya untuk basis data kecil atau pribadi.
7. Implementasi Mudah: Arsitektur satu tingkat dapat diimplementasikan dengan cepat dan mudah, menjadikannya pilihan yang baik untuk proyek-proyek kecil.

C. Arsitektur Dua Tingkat (*Two Tier Architecture*)

Arsitektur dua tingkat dalam DBMS mengacu pada arsitektur berbasis klien-server di mana antarmuka pengguna dan logika aplikasi dipisahkan menjadi dua komponen terpisah. Komponen klien biasanya adalah antarmuka pengguna, sedangkan komponen server bertanggung jawab untuk mengelola data dan logika bisnis. Dalam arsitektur ini, komponen klien berkomunikasi langsung dengan komponen server untuk meminta data dan melakukan tindakan.



Beberapa contoh arsitektur dua tingkat adalah:

1. Arsitektur File-Server: Klien mengakses file server bersama untuk penyimpanan dan pengambilan data.
2. Arsitektur Klien-Server: Klien berkomunikasi langsung dengan server basis data untuk penyimpanan dan pengambilan data.
3. Arsitektur Terminal Services: Klien berkomunikasi langsung dengan server terminal, yang kemudian berkomunikasi dengan server basis data untuk penyimpanan dan pengambilan data.
4. Arsitektur Berbasis Web: Peramban web berkomunikasi langsung dengan server web, yang berkomunikasi dengan server basis data untuk penyimpanan dan pengambilan data.
5. Arsitektur *Remote Procedure Call* (RPC): Klien berkomunikasi langsung dengan server basis data menggunakan panggilan prosedur jarak jauh untuk penyimpanan dan pengambilan data.

Beberapa keuntungan dari arsitektur dua tingkat dalam DBMS adalah:

1. Kesederhanaan: Arsitektur dua tingkat sederhana dan mudah dimengerti, karena melibatkan hanya dua komponen, yaitu klien dan server.
2. Hemat Biaya: Arsitektur dua tingkat lebih murah untuk diimplementasikan dan dipelihara dibandingkan dengan arsitektur tiga tingkat atau multi-tingkat.
3. Kemudahan Implementasi: Perangkat lunak klien dapat diimplementasikan pada workstation individu, sehingga lebih mudah dikelola dan diperbarui.
4. Akses Langsung ke Basis Data: Klien memiliki akses langsung ke basis data, memungkinkan pengambilan data dan pembaruan data yang cepat.
5. Pemrosesan di Sisi Klien: Beberapa pemrosesan dapat dilakukan di sisi klien, mengurangi beban pada server dan meningkatkan kinerja.
6. Skalabilitas: Arsitektur ini dapat dengan mudah ditingkatkan dengan menambahkan lebih banyak klien atau meningkatkan perangkat keras server.
7. Independensi: Klien dan server dapat dikembangkan dan diimplementasikan secara independen, yang dapat meningkatkan proses pengembangan secara keseluruhan.

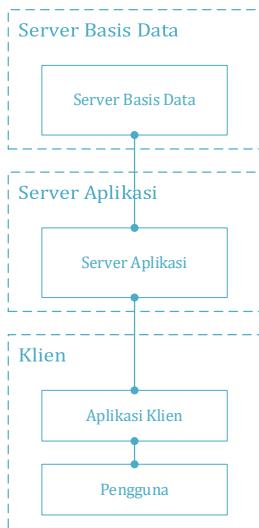
D. Arsitektur Tiga Tingkat (*Three Tier Architecture*)

Arsitektur tiga tingkat merupakan perluasan dari arsitektur dua tingkat yang mencakup lapisan perantara,

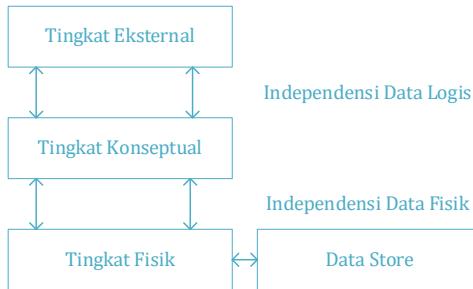
yaitu lapisan Aplikasi, antara klien dan server. Klien tidak dapat berkomunikasi langsung dengan server dengan desain ini.

Di sisi klien, program berkomunikasi dengan server aplikasi, yang kemudian berkomunikasi dengan sistem basis data. Di luar server aplikasi, pengguna akhir tidak memiliki pengetahuan tentang keberadaan basis data. Selain aplikasi, basis data tidak memiliki pengetahuan tentang pengguna lain.

Dalam aplikasi web yang luas, desain tiga tingkat ini sangat umum digunakan.



Arsitektur Tiga Tingkat dalam DBMS dapat dibagi dengan abstraksi data tingkat *internal*, *logical*, dan *view*.



Arsitektur tiga tingkat membagi sistem lengkap menjadi tiga tingkat seperti yang ditunjukkan di bawah ini:

1. Tingkat Fisik (*Internal*): Informasi tentang lokasi objek basis data dalam penyimpanan data fisik dijaga pada tingkat fisik. Beberapa pengguna DBMS tidak tahu di mana objek-objek ini berada. Dalam kata-kata sederhana, tingkat fisik basis data menentukan bagaimana data disimpan dalam perangkat penyimpanan sekunder seperti disk dan pita.
2. Tingkat Konseptual (*Logical*): Pada tingkat konseptual, data diwakili oleh beberapa tabel basis data. Ini menentukan jenis data yang akan disimpan dalam basis data dan juga dikenal sebagai skema logis.
3. Tingkat Eksternal (*View*): Tingkat eksternal mendefinisikan tampilan data dalam hal tabel tingkat konseptual. Setiap tampilan tingkat eksternal dirancang untuk memenuhi kebutuhan kelompok pengguna tertentu.

Dengan demikian, tampilan terpisah dapat dibuat untuk pengguna yang berbeda. Fokus utama tingkat eksternal adalah abstraksi data.

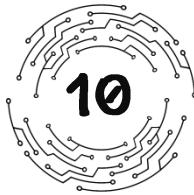
Beberapa keuntungan dari arsitektur tiga tingkat adalah:

1. Skalabilitas: Arsitektur ini dapat dengan mudah ditingkatkan dengan menambahkan lebih banyak server atau meningkatkan perangkat keras server yang ada, meningkatkan kinerja dan memastikan ketersediaan tinggi.
2. Keamanan yang Ditingkatkan: Arsitektur tiga tingkat menyediakan pemisahan tugas yang jelas, yang memudahkan pengamanan data dan pengendalian akses ke basis data.
3. Pemeliharaan yang Ditingkatkan: Arsitektur ini memudahkan pemeliharaan dan peningkatan sistem, karena perubahan dapat dilakukan pada satu lapisan tanpa memengaruhi yang lain.
4. Reusabilitas: Logika bisnis dapat dipusatkan pada lapisan aplikasi, memudahkan untuk digunakan kembali dan dibagikan di berbagai aplikasi.

Perbedaan kunci antara arsitektur dua tingkat dan tiga tingkat adalah:

1. Jumlah Lapisan: Arsitektur dua tingkat memiliki dua lapisan (klien dan server), sedangkan arsitektur tiga tingkat memiliki tiga lapisan (presentasi, aplikasi, dan basis data).
2. *Middleware*: Arsitektur dua tingkat tidak memiliki komponen *middleware* terpisah, sedangkan arsitektur tiga tingkat mencakup lapisan aplikasi sebagai komponen *middleware* antara klien dan server.

3. Skalabilitas: Arsitektur dua tingkat kurang dapat ditingkatkan dibandingkan dengan arsitektur tiga tingkat karena server aplikasi dalam arsitektur tiga tingkat dapat menangani lebih banyak klien dan pemrosesan yang kompleks.
4. Pemeliharaan: Arsitektur dua tingkat lebih mudah untuk dipelihara dibandingkan dengan arsitektur tiga tingkat karena memiliki lebih sedikit komponen, tetapi juga dapat menjadi *bottleneck* ketika jumlah klien dan kompleksitas sistem meningkat.
5. Keamanan: Arsitektur tiga tingkat memberikan keamanan yang lebih baik karena komponen middleware dapat menerapkan kebijakan keamanan dan menyembunyikan informasi sensitif dari klien.



Model Data

Model data merupakan alat yang sangat *powerful* untuk menggambarkan kebutuhan dan kapabilitas sistem informasi (Simsion dan Witt, 2005). Karena waktu yang dibutukan untuk meninjau model data lebih sedikit dibandingkan dengan meninjau spesifikasi kebutuhan fungsional sistem secara keseluruhan.

Pemodelan data adalah teknik yang digunakan untuk mendefinisikan dan mengelola proses bisnis. Model data memungkinkan organisasi untuk membuat deskripsi visual dari bisnis dengan melakukan analisis, memahami, dan mengklarifikasi kebutuhan data untuk mendukung proses bisnis. Pemodelan data yang baik dapat membuat pemrograman menjadi lebih sederhana dan lebih murah, sedangkan pengelolaan data yang buruk dapat menyebabkan biaya perbaikan yang mahal.

Model data memiliki peranan yang sangat besar dalam menjaga kualitas data. Perubahan pada struktur basis data akan memberikan dampak yang besar pada sistem yang menggunakananya, sehingga perlu dibuat model data yang akurat dan stabil. Model data mungkin hanya bagian kecil dalam pengembangan sistem informasi, tetapi memiliki dampak yang besar terhadap kualitas dari sistem yang dibangun.

A. Jenis-Jenis Model Data

1. Model Data Konseptual

Model data konseptual merupakan tahap awal dalam pemodelan data. Model ini merupakan representasi abstrak dari data dan relasi antar data dalam sistem.

Model data konseptual pertama kali diperkenalkan oleh Charles Bachman pada tahun 1960-an. Dia menggunakan persegi untuk menunjukkan tipe *record* dan panah yang menghubungkan antar *record*. Kemudian pada tahun 1976, Peter Chen memperkenalkan pendekatan *entity-relationship* (ER) (Teorey, Lightstone dan Nadeau, 2006).

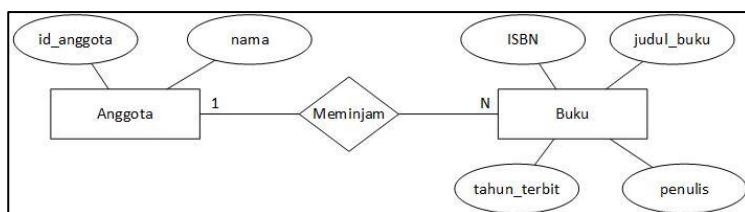
Komponen utama dalam model data konseptual adalah:

- a. Entitas – objek, orang, tempat, konsep, atau peristiwa dari dunia nyata. Contoh dalam sistem perpustakaan, entitasnya adalah **buku**, **anggota**, dan **peminjaman buku**.
- b. Atribut – karakteristik dari entitas yang menyimpan informasi tertentu. Contoh atribut dari entitas buku adalah **judul buku**, **penulis**, **tahun terbit**, dan **ISBN**.
- c. Relasi – hubungan yang menggambarkan interaksi antar entitas. Contoh relasinya adalah anggota meminjam buku.
- d. Kardinalitas – menggambarkan jumlah entitas yang terlibat dalam sebuah hubungan. kardinalitas dapat berupa *one-to-one*, *one-to-many*, dan *many-to-many*. Contohnya relasi antara anggota dan buku, kardinalitasnya adalah *one-to-many*, yang berarti satu anggota dapat meminjam banyak buku, tetapi

satu buku hanya dapat dipinjam oleh satu anggota pada satu waktu.

- e. *Primary key* - atribut yang digunakan untuk mengidentifikasi secara unik setiap entitas dalam model data konseptual. Contohnya pada entitas buku atribut yang menjadi *primary key* adalah **ISBN**.

Contoh sederhana model data konseptual dapat dilihat pada Gambar 1. Model data konseptual merupakan model data yang menggambarkan keadaan sebenarnya. Model data ini berguna sebagai alat komunikasi antara perancang dengan *end-user* dan sebagai alat verifikasi asumsi yang muncul dalam proses pembuatan model.



Gambar 1 Model data konseptual

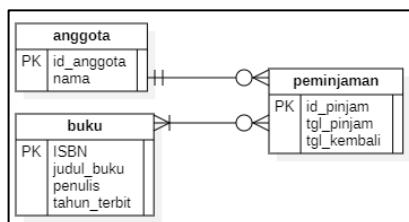
2. Model Data Logis

Model data logis merupakan tahap kedua dalam pemodelan data. Tahap ini menggambarkan cara data akan diorganisasikan, disimpan, dan diakses dalam basis data. Tahap ini mengubah model data konseptual menjadi model data logis. Model data logis biasanya diimplementasikan pada DBMS (*Database Management Systems*). Model data logis merupakan struktur ideal yang menggambarkan kebutuhan bisnis dan atribut data serta ketergantungan fungsionalnya (*functional dependency*).

Komponen utama dalam model data logis adalah sebagai berikut:

- a. Skema data – struktur formal yang menggambarkan bagaimana data akan diorganisasikan dalam basis data.
- b. Tabel – komponen utama dalam skema data. Setiap tabel mewakili entitas dalam model data konseptual. Misalnya, jika terdapat entitas **buku** pada model data konseptual, maka pada model data logis terdapat tabel **buku**.
- c. Kolom – merepresentasikan atribut dari entitas. Setiap kolom memiliki tipe data, seperti teks, angka, atau tanggal.
- d. *Primary key* – kolom yang digunakan untuk mengidentifikasi setiap baris secara unik.
- e. *Foreign key* – kolom yang digunakan untuk merujuk kunci primer dari tabel lain.
- f. Normalisasi – proses mengorganisasi data dalam tabel untuk mengurangi data yang berulang (*redundant*) dan menjaga integritas data. Proses ini memisahkan data menjadi beberapa tabel yang lebih kecil dan saling terkait.

Contoh model data logis dapat dilihat pada Gambar 2. Model data logis merupakan model data yang digunakan untuk mementukan bagaimana sistem harus dilaksanakan.



Gambar 2 Model data logis

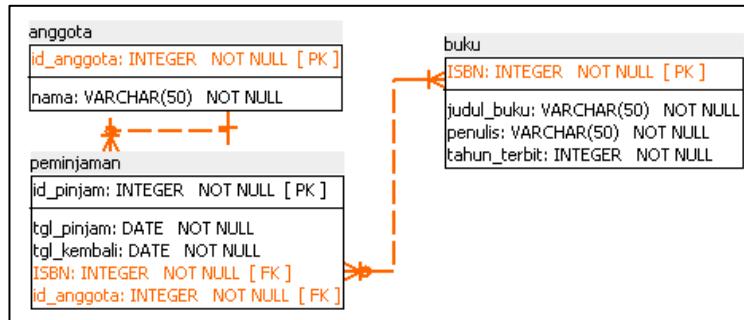
3. Model Data Fisik

Model data fisik adalah tahap terakhir dalam pemodelan data. Model data ini menggambarkan implementasi konkret dari struktur data dalam basis data. Model data fisik merupakan model data yang digunakan untuk menggambarkan bagaimana proses sistem informasi diimplementasikan menggunakan DBMS tertentu.

Komponen utama dalam model data fisik adalah:

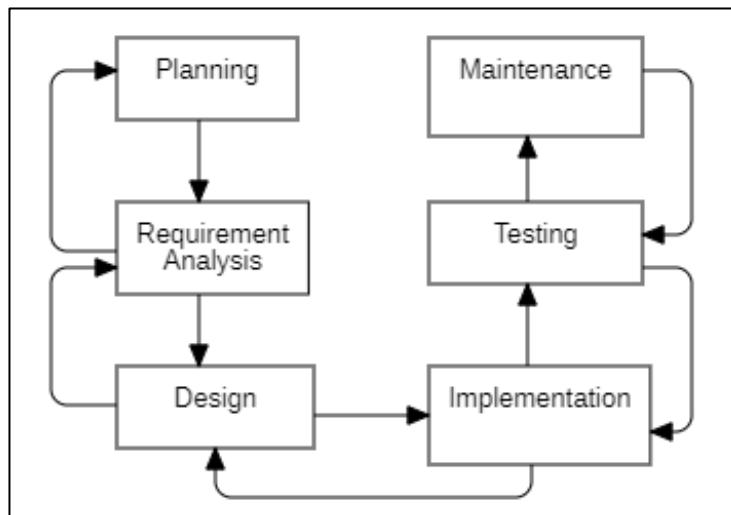
- a. Struktur tabel – mendefinisikan tabel yang akan digunakan dalam basis data.
- b. Konstrain – mendefinisikan *primary key*, *foreign key*, *unique*, dan lain-lain, untuk memastikan integritas data dapat terjaga.
- c. Indeks – menentukan indeks pada tabel. Indeks digunakan untuk mempercepat pencarian dan pengambilan data.
- d. Kardinalitas – menggambarkan hubungan antara tabel. Sama seperti pada model data konseptual, kardinalitas dapat berupa *one-to-one*, *one-to-many*, dan *many-to-many*.

Contoh sederhana model data fisik dapat dilihat pada Gambar 3. Model data fisik sangat bergantung pada DBMS yang akan digunakan, karena setiap DBMS memiliki karakteristik yang berbeda-beda. Implementasi model data fisik dapat bervariasi, tergantung kepada perangkat keras dan perangkat lunak yang digunakan. Hal ini juga akan mempengaruhi bagaimana kinerja basis data.

*Gambar 3. Model data fisik*

B. Siklus Hidup Basis Data

Siklus hidup basis data menggabungkan langkah dasar yang terlibat dalam merancang skema global dari basis data, mengalokasikan data melalui jaringan komputer, dan mendefinisikan skema spesifik dari DBMS (Teorey, Lightstone dan Nadeau, 2006). Siklus hidup basis data dapat diilustrasikan seperti pada Gambar 4.

*Gambar 4. Siklus hidup basis data*

1. Perencanaan (*Planning*)

Tahap perencanaan adalah awal dari siklus hidup basis data. Pada tahap ini dilakukan identifikasi tujuan dan kebutuhan bisnis terhadap basis data.

2. Analisis kebutuhan (*Requirement Analysis*)

Kebutuhan basis data ditentukan dengan menggali semua informasi dan proses bisnis dari semua *stakeholder* yang terlibat sehingga menghasilkan spesifikasi kebutuhan fungsional dan non-fungsional.

3. Perancangan (*Design*)

Pada tahap ini dilakukan Perancangan struktur basis data seperti entitas, atribut, relas, dan skema tabel. Tahap ini mencakup pemodelan data konseptual, pemodelan data logis, dan pemodelan data fisik.

4. Implementasi (*Implementation*)

Basis data dirancang berdasarkan hasil desain pada tahap sebelumnya dengan menggunakan *data definition language (DDL)* serta *data manipulation language (DML)* untuk manipulasi atau mengubah data.

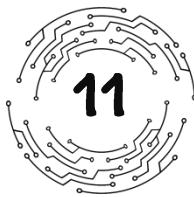
5. Pengujian (*Testing*)

Pengujian dilakukan untuk memastikan kinerja basis data sesuai dengan yang diharapkan dan memenuhi semua aspek kebutuhan organisasi.

6. Pemeliharaan (*Maintenance*)

Ketika basis data berhasil diimplementasikan perlu dilakukan pemantauan untuk memastikan apakah basis data sudah sesuai dengan kebutuhan organisasi, jika belum maka perlu dilakukan modifikasi untuk dapat

memenuhi semua kebutuhan organisasi dan meningkatkan kinerja basis data yang lebih baik.



Normalisasi

A. Basis Data dan Redudansi Data

Posisi dan peranan data saat ini sangat penting dan memiliki kedudukan yang diperhitungkan. Oleh karena data menjadi salah satu hal yang penting dan memiliki kebermanfaatan yang baik, maka proses penyimpanan dan pengumpulan data menjadi satu yang ditinjau dari sisi aksesibilitas, efektifitas mudah didapatkan diperlukanlah *database* (basis data). Basis data adalah kumpulan data yang disimpan secara elektronik yang terorganisir dengan baik. Organisasi yang baik menjadi kunci dari sebuah basis data, dengan kemudahan dan juga kelebihan dari sebuah basis data. Basis data memiliki komponen penyusun, adapun komponen penyusun basis data adalah tabel, kolom, baris, kunci dan indeks. Array adalah unit dasar yang digunakan untuk menyimpan data, sedangkan kolom mewakili atribut atau bidang tabel. Baris tabel berisi data tertentu dan kunci digunakan untuk mengidentifikasi setiap baris secara unik. Indeks digunakan untuk mempercepat pengambilan data dari tabel.

Pemanfaatan dan penggunaan basis data dalam penyimpanan data tentu memiliki banyak manfaat dan nilai lebih, akan tetapi sebuah basis data tentu tidak akan luput dari yang nama kekurangan, seperti salah satu permasalahan klasik tentang data yakni mengenai redudansi data. Redudansi data adalah kondisi dimana ada pengulangan atau penyimpanan dalam sebuah database untuk data yang sama. Hal ini bisa terjadi karena beberapa kemungkinan seperti misalkan pada kasus supermarket, seorang pelanggan bisa saja melakukan transaksi yang lebih dari satu kali dalam kurun waktu tertentu, sehingga *input* data ke dalam basis data menjadi berulang. Redudansi data ini dapat menyebabkan berbagai jenis masalah seperti : (Elmasri, 2015)

1. Pemborosan ruang penyimpanan.

Proses redudansi data tentu dapat menyebabkan pemborosan ruang penyimpanan database yang memiliki keterbatasan kapasitas untuk melakukan penyimpanan suatu data atau hal yang sama, ini menjadikan tidak efektif.

2. Inkonsistensi data

Proses redudansi juga dapat membuat data menjadi inkonsisten misalkan pembaruan dilakukan di suatu tabel, namun pada tabel lain data tidak dilakukan pembaruan. Ini dapat berakibat terhadap integritas data.

3. Kesulitan pemeliharaan

Proses redudansi juga memiliki implikasi langsung terhadap pemeliharaan yang biasa dilakukan pada sistem atau yang berhubungan dengan basis data. Hal ini disebabkan ketika data perlu diperbarui, perlu diubah di banyak tempat dan ini rentan terjadi kesalahan atau *human error* (Hills, 2016).

Berdasarkan hal-hal tersebut menjadi penting untuk menghindari redudansi data. Salah satu cara yang dapat dilakukan untuk menghindari redudansi data adalah dengan cara menormalkan basis data atau sering disebut dengan normalisasi data.

B. Normalisasi Data

Prosedur normalisasi memerlukan pemeriksaan struktur relasional untuk memastikan bahwa skema database sesuai dengan bentuk normal tertentu. Tahapan proses dilakukan secara sistematis, dengan setiap hubungan dievaluasi berdasarkan persyaratan bentuk normal. Melalui analisis, pendekatan ini dianggap sebagai bagian dari desain relasional. Codd pertama kali mendalilkan tiga bentuk normal, yang ia sebut sebagai bentuk normal pertama, kedua, dan ketiga. Belakangan, Boyce dan Codd menyajikan definisi yang lebih tepat untuk bentuk normal ketiga, yang mereka juluki sebagai bentuk normal Boyce-Codd (BCNF). Semua bentuk didasarkan pada pemeriksaan hubungan fungsional antar atribut hubungan. Selanjutnya, berdasarkan konsep ketergantungan bernilai ganda dan ketergantungan kompleks, diusulkan bentuk normal keempat (4NF) dan kelima (5NF) (Erickson, 2014).

Normalisasi juga dapat disebut sebagai proses pengorganisasian properti data menjadi entitas yang sederhana, tidak berlebihan. Jika rancangan basis data sudah memenuhi kaidah normalisasi, maka data yang dikembangkan bisa memiliki kualitas yang baik. Tujuan normalisasi database adalah untuk menghindari dan mengurangi redundansi data sekaligus menjaga ketergantungan data (data dalam tabel yang sesuai). Jika data dalam database tidak

dinormalisasi, dapat terjadi tiga hal yang merugikan sistem secara keseluruhan. Yakni penyimpangan, kesalahan, atau inkonsistensi data yang muncul selama proses input, penghapusan atau perubahan data yang sering didefinisikan sebagai anomali (Blaha, M. Rumbaugh, 2005).

Dalam prosesnya, ada beberapa tahapan yang harus dilakukan untuk melakukan normalisasi. Adapun tahapan Normalisasi Database:

1. Unnormalized Form (UNF) / Bentuk Tidak Normal
2. First Normal Form (1NF)
3. Second Normal Form (2NF)
4. Third Normal Form (3NF)
5. Boyce Code Normal Form (BCNF)
6. Fifth Normal Form (5NF)

C. Contoh Kasus Normalisasi Data

Pada sub-bab ini akan membahas proses ke proses normalisasi data yang berawal dari kondisi data un-normal. Contoh kasus pada normalisasi data kali ini adalah menggunakan kasus pada sistem basis data di poliklinik.

1. Unnormalized Form (UNF)

Basis data di poliklinik tentu menyimpan banyak sekali *record* kunjungan dari pasien. Dalam proses pencatatannya terdapat pencatatan data sebagai berikut :

Tanggal	NIK	Nama	Dokter	Spesialisasi	Diagnosis
01-08-23	1111	Andi	Panca	Umm	Demam

	1112	Asti	Bety	Gigi	Impaksi
02-08-23	1113	Abi	Ronald	Umu m	Flu
	1118	Dhika	Satria	Anak	Demam
03-08-23	1120	Dita	Desti	Kulit	Luka Bakar

Data diatas masih un-normal karena pada normal pertama (1NF) tidak boleh ada *Multivalue* pada kolom. Tahap yang perlu dilakukan adalah dengan menormalkan data pada tahap 1NF.

2. First Normal Form (1NF)

Suatu tabel dikatakan 1NF jika dan hanya jika setiap atribut dari data tersebut hanya memiliki nilai tunggal dalam satu baris. Maka, dari data yang masih Un-Normal tersebut, dipecah menjadi satu nilai satu baris berikut ini:

Tanggal	NIK	Nama	Dokter	Spesi alisasi	Diagnosi s
01-08-23	1111	Andi	Panca	Umu m	Demam
01-08-23	1112	Asti	Bety	Gigi	Impaksi
02-08-23	1113	Abi	Ronald	Umu m	Flu
02-08-23	1118	Dhika	Satria	Anak	Demam
03-08-23	1120	Dita	Desti	Kulit	Luka Bakar

Setelah dilakukan proses pemecahan baris data, data kini sudah ada pada bentuk pertama (First Normal Form). Tahapan berikutnya adalah memproses ke dalam bentuk normal kedua

3. Second Normal Form (2NF)

Syarat 2NF adalah tidak diperkenankan adanya partial “*functional dependency*” kepada primary key dalam sebuah tabel. *Functional dependency* adalah setiap atribut yang bukan kunci (*non key*) bergantung secara fungsional terhadap primary key. Sebelum melakukan proses normalisasi kedua (2NF) ada beberapa hal yang perlu diperhatikan terlebih dahulu. Salah satunya terkait posisi “*Key*” dalam basis data yakni terkait dengan *Candidate Key*, *Primary Key*, *Unique Key* dan *Foreign Key*.

- a. *Candidate Key* adalah satu atau kombinasi beberapa kolom unik yang menjadi penanda bahwa data tersebut merupakan satu kesatuan data yang mewakili kolom-kolom lainnya.
- b. *Primary Key* adalah kolom unik yang menjadi pembeda antar row yang bisa dijadikan referensi dari data tersebut dan tidak boleh null.
- c. *Unique Key* adalah kolom yang nilainya unik. Bedanya dengan Primary Key adalah Unique Key membolehkan nilainya nullable maupun non-nullable.
- d. *Foreign Key* adalah kolom yang menjadi penghubung relasi antar tabel sebagai referensi data dari satu tabel ke tabel lainnya.

Kondisi untuk memenuhi kriteria 2NF memenuhi persyaratan 1NF dan tidak memiliki ketergantungan parsial/fungsional. Ketergantungan parsial/fungsional terjadi

ketika sebuah tabel memiliki lebih dari satu kunci kandidat dan beberapa kolom hanya bergantung pada satu kunci kandidat. Sehingga didapatkan bentuk pada normal kedua sebagai berikut :

Tabel Transaksi

ID_Transaksi	Tanggal	NIK	Nama	ID_Dokter	Diagnosis
T0001	01-08-23	1111	Andi	0001	Demam
T0002	01-08-23	1112	Asti	0002	Impaksi
T0003	02-08-23	1113	Abi	0003	Flu
T0004	02-08-23	1118	Dhika	0004	Demam
T0005	03-08-23	1120	Dita	0005	Luka Bakar

Tabel Dokter

ID_Dokter	Nama	Spesialisasi
0001	Panca	Umum
0002	Bety	Gigi
0003	Ronald	Umum
0004	Satria	Anak
0005	Desti	Kulit

Pada kasus ini masalah yang akan pengulangan data pada kolom *Non-Candidate Key* jika dokter menangani lebih dari satu pasien dalam satu hari, dan kemungkinan untuk terjadi itu sangat tinggi, sehingga akan menyulitkan proses *maintenance*. Prosesnya akan menjadi update lebih dari satu row. Oleh karena itu kolom-kolom yang memiliki Partial/

Functional Dependency dipisah menjadi tabel sendiri agar memenuhi syarat 2NF.

Proses pada tabel yang terbentuk menjadi 2 tabel yakni tabel transaksi dan tabel dokter, sehingga tidak ada lagi pengulangan data pada kolom Non-Candidate Key pada tabel yang baru terbentuk. Pemisahan juga dilakukan pada Primary Key-nya menggunakan auto-incremented, sedangkan Candidate Key menggunakan constraint Unique Key. Kedua tabel tersebut dihubungkan oleh kolom ID_Dokter di tabel Transaksi.

4. Third Normal Form (3NF)

Kriteria yang memenuhi 3NF adalah sudah memenuhi 2NF dan tidak ada Transitive Dependency. Transitive Dependency adalah ketika kolom A bergantung pada kolom B yang bukan Candidate Key, dan kolom B bergantung pada kolom C yang menjadi Candidate Key, tetapi kolom C tidak bergantung pada kolom B. Jadi, ada kolom yang bergantung pada kolom lain yang bukan Candidate Key.

Tabel Dokter

ID_Dokter	Nama	Spesialisasi	Reg_Code	Region
D0001	Panca	Umum	JKT	Jakarta
D0002	Bety	Gigi	BDG	Bandung
D0003	Ronald	Umum	SBY	Surabaya
D0004	Satria	Anak	MDN	Medan
D0005	Desti	Kulit	YOG	Yogyakarta

Table Dokter

ID_Dokter	Nama	Spesialisasi	ID_Region
D0001	Panca	Umum	RG0001
D0002	Bety	Gigi	RG0002
D0003	Ronald	Umum	RG0003
D0004	Satria	Anak	RG0004
D0005	Desti	Kulit	RG0005

Table Region

Id_Region	Reg_Code	Region
RG0001	JKT	Jakarta
RG0002	BDG	Bandung
RG0003	SBY	Surabaya
RG0004	MDN	Medan
RG0005	YOG	Yogyakarta

Tabel di atas sudah memenuhi 3NF. Sudah tidak ada lagi Transitive Dependency. Saat melakukan perubahan nama region, cukup melakukan pergantian satu row pada tabel region saja, tabel dokter hanya mereferensikan datanya lewat Id_Region.

5. Boyce-Codd Normalization Form (BCNF)

Secara praktis, tujuan rancangan database adalah cukup sampai pada level 3NF. Akan tetapi untuk kasus-kasus tertentu kita bisa mendapatkan rancangan yang lebih baik lagi apabila bisa mencapai ke BCNF. Kondisi pada BCNF tidak ada Functional Dependency sama sekali

pada tabel yang memiliki kombinasi kolom Candidate Key untuk menghindari redundancy. Misalkan pada kasus tabel berikut ini :

Tabel Resep

Id_Resep	Nama Resep	Apoteker
RS0001	Resep_A1	Asri
RS0002	Resep_A2	Bella
RS0003	Resep_A3	Devi
RS0004	Resep_A4	Dita
RS0005	Resep_A5	Gista

Tabel Tindakan

Id_Tind	Keterangan Tindakan	Nama Dokter	Id_Resep
T0001	Pemeriksaan umum	Panca	RS0001
T0002	Antenatal Care	Bety	RS0002
T0003	Terapi	Ronald	RS0003
T0004	Pemeriksaan Lab	Satria	RS0004
T0005	Pemeriksaan umum	Desti	RS0005

Tabel di atas memenuhi 3NF karena tidak ada Transitive Dependency. Namun, melanggar BCNF karena

ada Functional Dependency. Terdapat yaitu kolom keterangan tindakan yang bergantung ke id_tind dan bukan Candidate Key.

Tabel Resep

Id_Resep	Nama Resep	Apoteker
RS0001	Resep_A1	Asri
RS0002	Resep_A2	Bella
RS0003	Resep_A3	Devi
RS0004	Resep_A4	Dita
RS0005	Resep_A5	Gista

Tabel Tindakan

Id_Tind	Keterangan Tindakan
T0001	Pemeriksaan umum
T0002	Antenatal Care
T0003	Terapi
T0004	Pemeriksaan Lab
T0005	Pemeriksaan umum

Tabel Detail Tindakan

Id_Tind	Id_Dokter	Id_Resep
T0001	D0001	RS0001
T0002	D0002	RS0002

T0003	D0003	RS0003
T0004	D0004	RS0004
T0005	D0005	RS0005

6. Fifth Normal Form (5NF)

Normalisasi hingga tingkat 5NF biasanya berkaitan dengan situasi yang sangat kompleks yang melibatkan multivariabel dan data yang sangat rumit. Dalam kasus sistem poliklinik sederhana ini, normalisasi hingga tingkat 3NF dan BCNF sudah mencukupi.

Langkah normalisasi basis data tersebut dapat diterapkan sesuai dengan masing-masing kebutuhan data, ada yang cukup sampai dengan normal ketiga (3NF), atau sampai dengan BCNF, namun untuk beberapa kasus dengan kompleksitas tinggi bisa sampai dalam bentuk normal kelima (5NF)



Data Warehouse

Bab ini akan membahas konsep dasar, desain, dan implementasi Data Warehouse, yang merupakan bagian penting dari dunia bisnis kontemporer. Data Warehouse memungkinkan organisasi untuk mengumpulkan, menyimpan, dan menganalisis data dari berbagai sumber untuk membantu mereka membuat keputusan yang lebih baik.

A. Konsep Dasar Data Warehouse

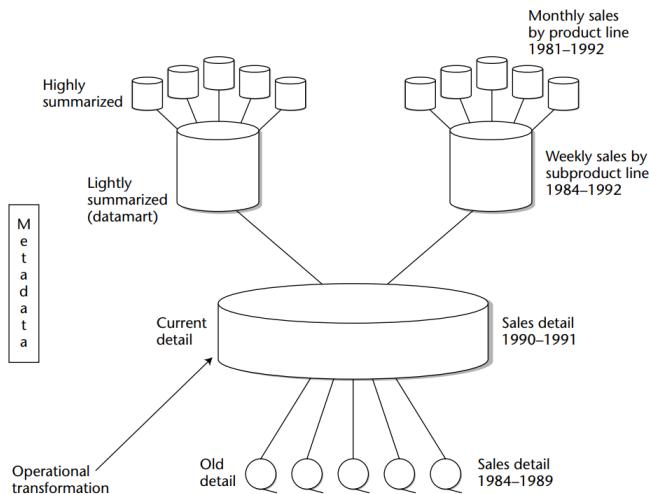
1. Definisi Data Warehouse

Data Warehouse adalah suatu sistem atau platform komputer yang dirancang khusus untuk menyimpan, mengelola, dan mengintegrasikan data historis untuk analisis dan pelaporan dari berbagai sumber.

Tujuan utama dari Data Warehouse adalah untuk menyediakan basis data yang kokoh dan terpusat sehingga dapat digunakan oleh organisasi untuk melakukan analisis yang lebih baik, melaporkan, dan membuat keputusan yang lebih baik. Data Warehouse dapat disebut sebagai dasar dari semua pemrosesan Sistem Pendukung Keputusan. (Inmon, 2005)

Beberapa karakteristik utama Data Warehouse adalah sebagai berikut:

- a. **Pemusatan dan Integrasi Data:** Data Warehouse adalah tempat di mana data dari berbagai sumber digabungkan dan disimpan secara terpusat dengan format yang seragam. Sehingga data historis dari berbagai sistem atau departemen perusahaan dapat digabungkan dan diakses dengan lebih mudah dan data bersifat konsisten.
- b. **Data Historis:** Data Warehouse biasanya memiliki sejarah atau riwayat data, yang memungkinkan untuk melacak tren dan perubahan dari waktu ke waktu. Hal tersebut sangat penting untuk analisis jangka panjang.
- c. **Berorientasi Subyek:** Data Warehouse diorganisasi berdasarkan subjek atau topik tertentu, bukan berdasarkan data operasi perusahaan yang sedang berjalan. Sehingga proses data warehousing lebih siap untuk menangani tema tertentu, dan memudahkan analisis lintas departemen atau fungsi.
- d. **Dapat Diakses untuk Analisis:** Data Warehouse dibuat untuk memudahkan analisis data mendalam dengan alat analisis, pelaporan, dan pertanyaan yang kompleks.
- e. **Stabilitas dan Konsistensi:** Data dalam Data Warehouse harus stabil dan konsisten, berarti perubahan hanya akan terjadi setelah proses ETL yang sesuai.



Gambar 1. Struktur Data Warehouse

Sumber : (Inmon, 2005)

Data Warehouse memungkinkan organisasi untuk menggali wawasan bisnis, menemukan tren, dan merencanakan strategi masa depan dengan lebih baik, serta memainkan peran penting dalam mendukung pengambilan keputusan berdasarkan bukti dan analisis data. (Albright and Winston, 2021)

2. Proses ETL (Ekstraksi, Transformasi, dan Pemuatan)

Proses ETL mengambil data dari berbagai sumber, mengubahnya sesuai kebutuhan, dan kemudian memasukkannya ke dalam Data Warehouse atau sistem penyimpanan data lainnya. (Kimbal and Ross, 2013). Berikut penjelasan lebih rinci tentang masing-masing komponen ETL:

- Ekstraksi (*Extraction*):** Tahap ini melibatkan pengambilan data dari berbagai sumber eksternal; sumber data dapat berupa basis data operasional,

file teks, *data streaming*, atau sumber data lainnya. Tujuan dari tahap ekstraksi adalah untuk memindahkan data dari sumber eksternal ke dalam lingkungan ETL untuk tahap selanjutnya.

- b. **Transformasi (*Transformation*):** Pada tahap ini, data yang telah diekstraksi diubah, diperbaiki, dan dipersiapkan agar sesuai dengan struktur dan format yang dibutuhkan oleh Data Warehouse atau sistem penyimpanan data yang dimaksud. Transformasi dapat mencakup berbagai tugas, seperti penggabungan data, penghapusan data yang tidak relevan, pembersihan data, normalisasi, perhitungan agregat, dan banyak lagi. Tujuannya adalah untuk menjadikan data konsisten, relevan, dan siap digunakan untuk analisis.
- c. **Pemuatan (*Loading*):** Tahap terakhir dalam proses ETL adalah pemuatan. Pada tahap ini, data yang telah diekstraksi dan diubah akan dipindahkan atau dimuat ke dalam Data Warehouse atau tempat penyimpanan data lainnya. Data dapat dimuat ke dalam struktur yang sesuai dengan model data warehouse, seperti skema bintang atau salju. Terdapat dua jenis utama tahap pemuatan yaitu pemuatan penuh (*full load*) dan pemuatan bertahap (*incremental load*).

Proses ETL memungkinkan organisasi untuk menyatukan data dari berbagai sumber, membersihkannya, dan menyajikannya dalam format yang cocok untuk analisis dan pengambilan keputusan.

B. Desain Data Warehouse

1. Model Data Warehouse

Model Data Warehouse adalah kerangka kerja atau struktur yang digunakan untuk merancang dan mengatur data dalam Data Warehouse. Model ini memainkan peran penting dalam menyediakan panduan tentang bagaimana data akan disimpan, diorganisasi, dan diakses dalam Data Warehouse. Terdapat dua model utama yang digunakan dalam Data Warehouse, yaitu Model Bintang (*Star Schema*) dan Model Salju (*Snowflake Schema*). (Inmon, 2005)

a. Model Bintang (*Star Schema*):

Dalam model ini, terdapat sebuah tabel fakta sentral yang berisi data fakta bisnis utama dan terhubung langsung ke tabel-tabel dimensi yang menggambarkan karakteristik atau atribut terkait. Struktur model bintang sederhana dan mudah dimengerti untuk analisis data.

Tabel fakta adalah pusat dari model bintang dan berisi metrik atau data yang akan diukur dan dianalisis, seperti penjualan, pendapatan, atau jumlah stok. Sedangkan tabel dimensi menggambarkan atribut-atribut yang berkaitan dengan data, seperti informasi pelanggan, produk, waktu, dan lokasi.

b. Model Salju (*Snowflake Schema*):

Model Salju adalah variasi dari model bintang yang lebih kompleks. Dalam model ini, tabel dimensi dibagi menjadi sub-dimensi yang lebih kecil atau normalisasi, sehingga menciptakan struktur yang menyerupai "kristal salju." Normalisasi dalam model salju dapat mengurangi

redundansi data, tetapi juga dapat menghasilkan struktur yang lebih kompleks dan sulit dimengerti.

Model salju sering digunakan ketika ada kebutuhan untuk mengelola data dengan tingkat normalisasi yang lebih tinggi atau ketika ada sumber data yang memerlukan normalisasi. (Kimbal and Ross, 2013)

2. Dimensi dan Fakta

Dimensi dan Fakta adalah dua komponen kunci dalam pemodelan dimensional yang digunakan dalam Data Warehouse.

- a. **Dimensi (*Dimensions*):** komponen dalam model Data Warehouse yang menggambarkan atribut atau karakteristik yang biasanya digunakan untuk menggolongkan, mengkategorikan, atau menggambarkan data. Misalnya: Tanggal, Produk, Pelanggan, dan Lokasi.

Dimensi digunakan sebagai panduan atau cara untuk melihat data. Mereka digunakan sebagai sumbu dalam analisis data dan memungkinkan pengguna untuk menyegmentasi, memfilter, atau melihat data dalam konteks tertentu.

- b. **Fakta (*Facts*):** komponen dalam model Data Warehouse yang berisi data yang diukur atau dihitung. Fakta adalah angka atau metrik bisnis yang ingin dianalisis. Misalnya: Total Penjualan, Jumlah Produk Terjual, atau Pendapatan.

Fakta biasanya terkait dengan dimensi, dan hubungan antara fakta dan dimensi menggambarkan bagaimana data diukur dalam konteks yang berbeda.

Fakta adalah inti dari informasi yang ingin ditemukan atau diungkap oleh pengguna Data Warehouse. (Kimbal and Ross, 2013)

C. Implementasi Data Warehouse

1. Pemilihan Perangkat Keras dan Perangkat Lunak

Pemilihan perangkat keras dan perangkat lunak adalah langkah penting dalam merancang dan mengimplementasikan sebuah sistem Data Warehouse yang efektif. Keduanya memiliki dampak besar terhadap kinerja, kehandalan, dan kemampuan sistem.

Pemilihan perangkat keras melibatkan pemilihan komputer, server, penyimpanan data, dan infrastruktur jaringan yang akan digunakan untuk menyimpan, mengelola, dan mengakses data dalam Data Warehouse. Perangkat keras harus memiliki kapasitas dan daya komputasi yang memadai untuk mendukung tugas analisis yang intensif. Selain itu harus mempertimbangkan redundansi, skalabilitas dan keamanan fisik.

Sedangkan pemilihan perangkat lunak berkaitan dengan pemilihan sistem manajemen basis data (DBMS), alat ETL, alat analisis data, dan perangkat lunak lain yang diperlukan untuk membangun dan mengoperasikan Data Warehouse.

Perangkat lunak harus mendukung berbagai tugas analisis seperti pembuatan laporan, analisis multi-dimensional, visualisasi data, dan pertanyaan kompleks. (Inmon, Terdema and Imhoff, 2016)

2. Kemanan Data Warehouse

Keamanan Data Warehouse adalah aspek kritis dalam merancang, mengelola, dan menjalankan sistem Data Warehouse. Keamanan bertujuan untuk melindungi data yang disimpan di dalamnya dari akses yang tidak sah, perubahan yang tidak sah, serta menjaga kerahasiaan, integritas, dan ketersediaan data.

D. Tantangan dan Kesempatan Data Warehouse

1. Tantangan Data Warehouse

Guna menjaga kinerja dan efektivitas Data Warehouse, penting untuk memahami dan mengatasi tantangan-tantangan Data Warehouse. Berikut adalah beberapa tantangan umum dalam Data Warehouse:

- a. Volume Data yang Besar: Mengelola dan menyimpan data dalam volume yang besar memerlukan kapasitas penyimpanan yang signifikan dan strategi pemrosesan yang efisien.
- b. Integrasi Data yang Rumit: Mengintegrasikan data dari berbagai sumber memerlukan pemodelan data yang cermat dan proses ETL yang kompleks.
- c. Kualitas Data yang Rendah: Mengatasi masalah kualitas, seperti data yang tidak lengkap, data yang tidak akurat, atau data yang duplikat merupakan tantangan besar.
- d. Kinerja yang Cepat: Desain yang tidak optimal atau hardware yang tidak memadai dapat menghambat kinerja.
- e. Keamanan dan Kepatuhan: Melindungi data dari akses yang tidak sah dan mematuhi regulasi

- keamanan dan privasi data adalah tantangan penting.
- f. Perubahan Bisnis yang Cepat: harus mampu menyesuaikan diri dengan perubahan dalam model data, persyaratan pelaporan, dan kebutuhan analisis.
 - g. Biaya yang Meningkat: Biaya perangkat keras, perangkat lunak, dan sumber daya manusia harus dikelola secara efisien.

2. Kesempatan Data Warehouse

Data Warehouse tidak hanya menghadirkan tantangan, tetapi juga membawa berbagai kesempatan dan manfaat bagi organisasi. Berikut adalah beberapa kesempatan utama dalam Data Warehouse:

- a. Analisis Data Mendalam: Data Warehouse memungkinkan organisasi untuk melakukan analisis data yang mendalam. Dengan data yang terkumpul secara terpusat dan terstruktur, pengguna dapat menjalankan analisis yang lebih kompleks dan mendalam untuk mengungkap wawasan bisnis yang berharga.
- b. Pengambilan Keputusan yang Lebih Baik: Dengan akses yang lebih mudah ke data yang relevan dan analisis yang lebih baik, organisasi dapat membuat keputusan yang lebih baik dan lebih terinformasi. Sehingga dapat membantu meningkatkan efisiensi operasional, merencanakan strategi yang lebih baik, dan mengidentifikasi peluang bisnis.
- c. Pemahaman Pelanggan yang Lebih Baik: Data Warehouse memungkinkan organisasi untuk mendapatkan pemahaman yang lebih baik tentang pelanggan mereka. Ini termasuk pemahaman

tentang perilaku pelanggan, preferensi, dan pola pembelian, yang dapat digunakan untuk meningkatkan pengalaman pelanggan dan strategi pemasaran.

- d. Peringatan Dini terhadap Perubahan Pasar: Dengan analisis data yang kontinu, organisasi dapat mendeteksi perubahan pasar, tren bisnis, atau perubahan perilaku pelanggan lebih cepat. Ini memungkinkan organisasi untuk merespons secara proaktif dan beradaptasi dengan perubahan tersebut.
- e. Keterbukaan Informasi: Data Warehouse dapat memberikan akses yang lebih besar ke data bagi berbagai pengguna dalam organisasi. Sehingga terjadi keterbukaan informasi dan kolaborasi lintas departemen.
- f. Pengurangan Redundansi Data: Data Warehouse membantu mengurangi redundansi data dan memastikan bahwa data yang sama digunakan secara konsisten dalam organisasi. Sehingga menghindari duplikasi dan inkonsistensi data.
- g. Penghematan Waktu dan Sumber Daya: Dengan data yang tersedia dalam format yang mudah diakses, organisasi dapat menghemat waktu dan sumber daya yang sebelumnya digunakan untuk mencari, membersihkan, dan menggabungkan data dari berbagai sumber.

E. Peran Data Warehouse dalam Sistem Informasi

Data Warehouse adalah fondasi yang krusial dan memiliki hubungan erat dengan konsep Sistem Informasi,

hal ini karena data Warehouse memberikan kemampuan organisasi untuk melakukan analisis data dan mengambil keputusan yang lebih baik berdasarkan data yang akurat dan relevan. Ini memungkinkan organisasi untuk merespons perubahan pasar dengan lebih baik, mengidentifikasi peluang bisnis, dan mengoptimalkan operasi mereka secara keseluruhan.

Data Warehouse dalam Sistem Informasi bertujuan untuk mengelola dan menyediakan akses terhadap data bisnis yang relevan, terstruktur, dan historis.

Integrasi data dalam Data Warehouse memungkinkan Sistem Informasi untuk memiliki visibilitas yang lebih lengkap dan terpadu atas seluruh aspek bisnis.

Data tersebut digunakan untuk menghasilkan laporan, menciptakan dasbor, menjalankan kueri kompleks, melakukan analisis prediktif dan melacak perubahan dan tren dalam data. Hal ini membantu organisasi dalam pengambilan keputusan yang lebih baik dan responsive dalam jangka waktu yang panjang.



Pengenalan ERD

Dalam sistem informasi, kita akan selalu bertemu dengan istilah database. Database adalah sebuah kumpulan informasi yang disimpan pada perangkat elektronik secara sistematis dan terstruktur sehingga selalu dapat dicari dan digunakan kembali dengan mudah sesuai kebutuhan. Namun dalam merancang sebuah database, kita dihadapkan pada masalah dalam menyusun database agar rapi, terukur dan terstruktur. Karena jika desain database yang dibuat tidak benar, permasalahan yang dapat muncul kemudian adalah inkonsistensi serta redundansi data.

Untuk dapat menghasilkan database yang baik, salah satu cara yang paling banyak digunakan dalam merancang database adalah dengan menggunakan ERD (Entity Relationship Diagram) (Afifah, Azzahra and Anggoro, 2022). ERD adalah sebuah diagram yang digunakan untuk menggambarkan keterkaitan atau hubungan yang terjadi antara entitas atau objek yang ada dalam sebuah database. Sebagai contoh Entitas pada objek barang akan terkait dengan entitas penjualan dengan menampilkan atribut-atribut masing-masing secara lengkap. Sehingga penggambaran keterkaitan antar objek/entitas dapat dilihat dengan jelas pada

ERD. Selain itu, dibeberapa kasus, jika terjadi masalah dengan database maka ERD akan digunakan sebagai *debugging database*.

A. Komponen Penyusunan ERD

Pada umumnya, ERD berhubungan langsung dengan Diagram Aliran Data untuk menggambarkan isi dari data *store*. Diagram ini akan dapat membantu dalam visualisasi koneksi data dan merupakan langkah penting dalam membangun basis data relasional. Sebelum mulai membuat ERD, penting untuk memahami beberapa istilah dan komponen umum yang terkait. Sebelum mengetahui cara membuatnya, mari kenali beberapa komponen-komponen penting ERD sebagai berikut (Yanto, 2016):

1. Entitas

Entitas atau *Entity* merupakan gambaran dari sebuah objek nyata, seperti mobil, buku, mahasiswa dan sebagainya. Penggambaran entitas ini menggunakan sebuah kotak persegi panjang. Entitas bisa dibedakan menjadi entitas yang bisa dilihat dan yang tidak bisa dilihat, untuk entitas yang dapat dilihat seperti yang disebutkan diatas, yaitu mobil, mahasiswa, buku dan lainnya, sedangkan entitas yang tidak dapat dilihat seperti Penjualan, Order dan lain sebagainya.

Mahasiswa

Gambar 1. Penggambaran Entitas

2. Atribut

Komponen penting berikutnya dari ERD adalah Atribut. Atribut atau elemen dapat diartikan sebagai identitas yang melekat pada Entitas yang berguna untuk menginformasikan karakteristik dari sebuah Entitas. Sebuah entitas akan selalu direpresentasikan dengan satu atau lebih atribut, yang mana salah satunya akan bertindak sebagai atribut kunci (key), atau atribut unik yang akan mewakili Entitas pada saat terjadi relasi antar entitas.



Gambar 2. Penggambaran Atribut pada ERD

Untuk menggambarkannya pada ERD, atribut dilambangkan sebagai elips yang didalamnya terdapat nama dari atribut tersebut. Sebagai contoh, Entitas mahasiswa akan mempunyai Atribut Nopb, Nama, alamat dan lain sebagainya. Nopb merupakan atribut kunci pada entitas mahasiswa karena dia bersifat unik dan dapat mewakili entitas mahasiswa apabila berelasi dengan entitas lainnya. Lalu, seberapa penting atribut kunci tersebut dan bagaimana cara mendefinisikannya. Ini adalah hal yang sering ditanyakan pada saat membangun ERD atau database. Atribut kunci adalah sebuah identitas yang bernilai unik yang ada pada atribut-atribut yang melekat pada Entitas. Mengenalinya sangat mudah, yaitu hanya dengan membutuhkan pemahaman bagaimana penggunaannya. Atribut kunci biasanya mewakili entitas atau objek dalam berelasi kepada entitas lain. Nilainya harus unik, tidak boleh ada

yang sama. Kita bisa mengambil contoh pada tabel entitas mahasiswa, mahasiswa dapat diwakili oleh sebuah atribut unik dengan nama Nobp, mengapa demikian?, karena Nobp bersifat unik, tidak ada yang sama, dalam sekumpulan data, hanya ada satu buah Nobp, sehingga atribut ini bisa dijadikan atribut kunci. Lalu bagaimana dengan objek yang tidak bisa dilihat?. Sebenarnya tidak terlalu sulit memberikan atribut kunci untuk entitas atau objek yang tidak bisa dilihat seperti “penjualan”, “order” dan sebagainya, hanya membutuhkan penambahan sebuah atribut dengan nama “Kode”. Untuk penjualan bisa memberikan nama “kode-penjualan”. Atribut ini nantinya akan menjadi sebuah atribut kunci yang unik yang akan mewakili objek penjualan.

3. Ralasi

Relasi atau *Relationship* adalah hubungan atau keterikatan yang terjadi antara Entitas satu dengan entitas lainnya. Relasi digambarkan atau diwakilkan dengan simbol belah ketupat.



Gambar 3. Gambar Relasi/Relationship

Didalam penggambaran relasi, selalu akan dihadapkan pada pemilihan jenis relasi yang tepat dan dapat menggambarkan kondisi yang sebenarnya dalam sebuah relasi. Berikut adalah beberapa jenis relasi dalam ERD:

- a. One To One (1 to 1) merupakan kondisi dimana sebuah entitas hanya boleh berelasi dengan 1 entitas lainnya. Implementasinya adalah hanya boleh terjadi relasi yang menghubungkan 1 baris data pada tabel A yang terhubung ke tabel B.
- b. One to Many (1 to M) merupakan relasi yang terjadi antara satu entitas dengan beberapa entitas lainnya.
- c. Many to Many adalah kondisi dimana terjadi relasi pada suatu entitas kepada satu atau lebih entitas lainnya, dan begitu juga sebaliknya, relasi jenis ini adalah relasi yang paling sederhana, yang selalu akan membutuhkan penyederhanaan relasi kembali sampai membentuk relasi minimal yaitu one to many.

B. Merancang ERD

Setelah mengetahui komponen-komponen dalam ERD, langkah selanjutnya adalah merancang atau membangun ERD. Dalam merancang ERD ada beberapa tahapan penting yang harus dilakukan untuk menjadikan rancangan ERD sesuai dengan yang seharusnya. Berikut adalah tahapan-tahapan atau langkah-langkah yang bisa dilakukan untuk membuat ERD dengan benar:

1. Identifikasi atau tentukan Entitas.

Langkah ini adalah langkah yang krusial, pada tahapan ini membutuhkan kemampuan dalam mengidentifikasi entitas apasaja yang muncul dan harus mewakili keperluan dalam sebuah database. Sebagai contoh, dalam sebuah database perusahaan, entitas yang mungkin harus ada adalah Unit, pimpinan, dan pegawai, kemudian gambarkan semua entitas ini dalam bentuk

persegi panjang seperti yang telah diuraikan diatas. Dalam sebuah basidata, kejelian dalam mendefinisikan setiap Entitas sangat diperlukan. Pada beberapa kasus, entitas dapat dikembangkan kembali menjadi entitas-entitas baru untuk meminimalisir terjadinya kesalahan baik redundansi data ataupun inkosistensi pada saat implementasi kedalam database.



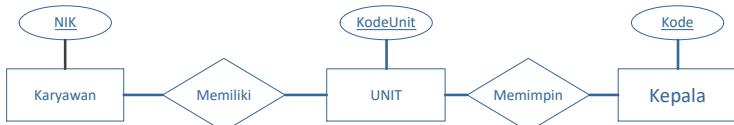
Gambar 4. Menentukan Entitas

Pada gambar 13.4 diatas, terlihat bagaimana mendefinisikan dan menggambarkan beberapa entitas seperti Unit, Kepala, dan karyawan, kemudian menentukan apa relasi yang ada atau yang mungkin ada diantara dua buah entitas. Diantara entitas Unit dan Kepala, terdapat relasi yang bisa dibuat yaitu “memimpin” relasi ini juga akan bertindak sebagai entitas baru yang muncul. Kemudian diantara Unit dan Karyawan, terdapat relasi lagi yang dapat ditulis dengan “Memiliki”. Dari gambaran kasar diagram ERD ini telah menghasilkan 3 buah entitas dan 2 buah entitas relasi.

2. Mengidentifikasi Atribut dan menambahkannya.

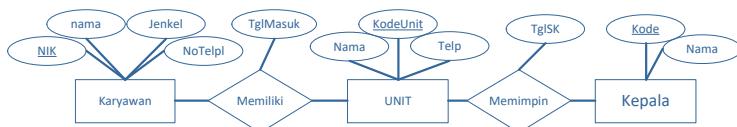
Setiap Entitas yang sudah didapatkan minimal harus mempunyai satu atribut yang mewakilinya. Untuk entitas yang bersifat berdiri sendiri atau entitas yang tidak tergantung pada entitas lain (pada contoh diatas adalah Unit, Kepala dan Karyawan), atau biasa disebut master, harus mempunyai 1 atribut kunci dan minimal 1 atribut

pelengkap. Sebagai contoh, untuk entitas Unit, setidaknya harus mempunyai KodeUnit sebagai atribut kunci dan ditambahkan beberapa atribut lain yang penting seperti nama_unit, Notelp dan sebagainya.



Gambar 5. Penambangan Atribut Kunci pada ERD

Setelah mendapatkan atribut kunci, selanjutkan adalah mengidentifikasi atribut-atribut tambahan yang menyertai setiap Entitas. Sebagai contoh, pada Entitas karyawan, atribut yang mungkin bisa muncul adalah NamaKaryawan, Alamat, Nomor Telepon, JenisKelamin dan sebagainya. Langkah ini dilakukan pada semua Entitas yang ada. Berikut adalah gambaran bagaimana atribut-atribut melekat pada setiap entitas yang ada.

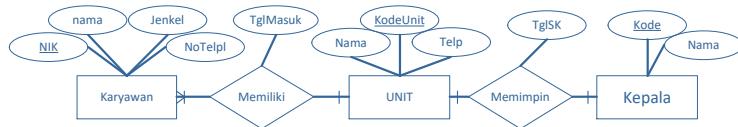


Gambar 6. Melengkapi atribut pada setiap Entitas

3. Tambahkan Derajat Keterikatan/Kardinalitas

Untuk menentukan derajat kardinalitas ini, yang harus dilakukan adalah menganalisa dan mengidentifikasi berapa data yang akan muncul dari setiap relasi yang ada. Tambahkan derajat kardinalitas sesuai

dengan penggambarannya. Ini adalah hal yang penting agar diagram mudah dipahami.



Gambar 7. Penambahan derajat kardinalitas

Pada tahapan ini, secara umum, pembuatan diagram sudah memenuhi kaidah perancangan ERD, hanya saja, untuk membuat ERD kerumitannya tergantung pada kasus yang diangkat, pada sebagian besar rancangan melibatkan relasi yang kompleks dan membutuhkan ketelitian lebih dalam membangunnya. Setelah menyelesaikan diagram ERD, yang diperlukan adalah menguji apakah ERD yang dibangun memenuhi kebutuhan dan terhindar dari kesalahan. Ini sangat penting dilakukan untuk menghindari terjadinya kesalahan pada database, karena jika rancangannya tidak sesuai, maka masalah yang akan muncul adalah inkonsistensi dan redundansi data. Pada bahasan ini, hanya sebatas pengenalan ERD secara sederhana, sehingga untuk merancang ERD yang lebih kompleks, membutuhkan latihan dan pengujian yang intensif.



Sistem Manajemen Pengetahuan

A. Manajemen Pengetahuan (Knowledge Management)

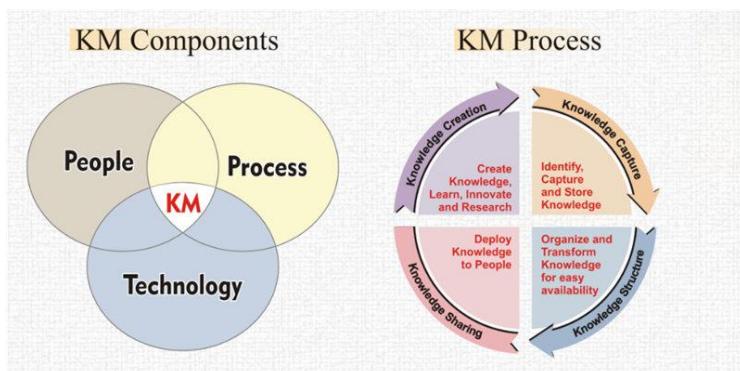
Pengetahuan merujuk pada informasi, fakta, atau pemahaman yang dimiliki oleh individu atau organisasi tentang berbagai hal. Ini dapat mencakup berbagai jenis pengetahuan, seperti pengetahuan ilmiah, pengetahuan praktis, atau pengetahuan budaya. Pengetahuan adalah hasil dari pengalaman, pembelajaran, dan pemahaman yang diperoleh oleh seseorang sepanjang waktu.

Pengetahuan dapat bersifat teoritis atau praktis. Pengetahuan teoritis mencakup konsep, teori, dan prinsip yang dapat digunakan untuk memahami dunia di sekitar kita, sementara pengetahuan praktis mencakup keterampilan, pengalaman, dan pandangan yang berguna dalam situasi sehari-hari.

Manajemen pengetahuan (knowledge management) adalah metode yang digunakan oleh individu atau organisasi untuk mengumpulkan, mengorganisasi, menyimpan, dan

berbagi informasi dengan efisien. Dengan manajemen pengetahuan yang baik, pengetahuan dapat diakses dengan lebih mudah, dibagikan, dan dimanfaatkan untuk tujuan tertentu, seperti inovasi, pengambilan keputusan, atau pemecahan masalah.

Manajemen pengetahuan terdiri dari tiga komponen utama: people, process, dan technology.



Gambar 1. Tiga komponen utama manajemen pengetahuan

1. People

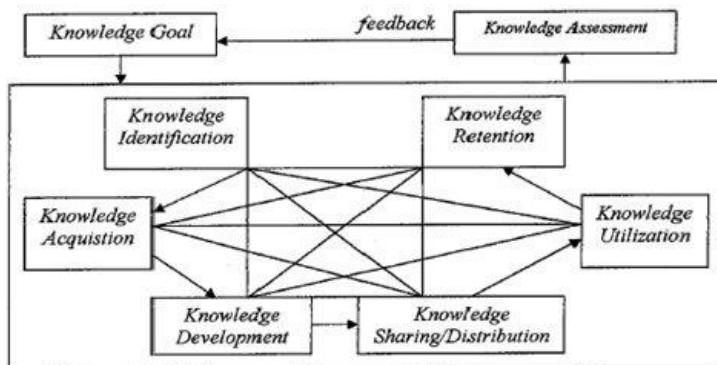
People merupakan mereka yang memiliki pengetahuan, mengelola sistem, dan terlibat dalam proses yang menggunakan pengetahuan untuk organisasi. Aktivitas berbagi membantu menyebarluaskan pengetahuan.



Gambar 2. People

2. Process

Process memastikan bahwa penggunaan manajemen pengetahuan berjalan dengan benar dengan menyesuaikan prinsip, strategi, praktik, dan proses.



Gambar 3. Process

3. Technology

Technology adalah sistem manajemen pengetahuan media yang memerlukan orang yang ahli untuk mengelolanya. Selama proses implementasi, banyak alat diperlukan untuk membantu berkomunikasi, mengelola konten, dan bekerja sama.

B. Sistem Manajemen Pengetahuan, juga dikenal sebagai KMS

Suatu metode yang digunakan oleh organisasi untuk mengumpulkan, mengatur, menyimpan, dan berbagi data. Meningkatkan inovasi, produktivitas, dan pengambilan keputusan organisasi adalah tujuan utama dari KMS.

C. Komponen utama dari sistem manajemen pengetahuan

1. Penyimpanan Pengetahuan: KMS mengintegrasikan teknologi informasi untuk menyimpan berbagai jenis pengetahuan, seperti dokumen, database, dan catatan. Penyimpanan ini memudahkan akses dan pencarian informasi yang relevan.
2. Pengaturan Pengetahuan: Informasi yang disimpan dalam KMS harus diorganisasi dengan baik agar mudah diakses dan dimengerti oleh anggota organisasi. Ini bisa meliputi pengelompokan berdasarkan topik, kategori, atau jenis.
3. Berbagi Pengetahuan: Salah satu aspek kunci dari KMS adalah membantu anggota organisasi berbagi pengetahuan. Ini dapat dilakukan dengan berbagai cara, seperti forum diskusi, wiki bersama, atau sistem kolaborasi.
4. Penggunaan Pengetahuan: KMS harus dirancang untuk memungkinkan penggunaan pengetahuan dalam pengambilan keputusan, pemecahan masalah, dan inovasi. Ini dapat melibatkan alat analisis data atau sistem pelaporan.
5. Pembaruan dan Penyempurnaan: Pengetahuan dalam organisasi selalu berkembang. Oleh karena itu, KMS harus memiliki mekanisme untuk memperbarui dan menyempurnakan pengetahuan yang ada agar tetap relevan.
6. Keamanan dan Hak Akses: Informasi yang disimpan dalam KMS harus dijaga keamanannya. Ini termasuk mengendalikan hak akses sehingga hanya orang-orang yang berwenang yang dapat mengakses informasi tertentu.

7. Pengukuran dan Evaluasi: Organisasi perlu mengukur efektivitas KMS mereka. Ini melibatkan pemantauan penggunaan, tingkat kepuasan pengguna, dan dampak positif pada kinerja organisasi.
8. Pelatihan dan Pendidikan: Penting bagi anggota organisasi untuk memahami cara menggunakan KMS dengan efektif. Oleh karena itu, pelatihan dan pendidikan tentang sistem ini juga merupakan komponen penting.

D. Manfaat Sistem Manajemen Pengetahuan

Agar perusahaan dapat terus berkembang dan tidak terlindas zaman, mereka harus memiliki pengetahuan yang mumpuni. Selain itu, pengetahuan ini digunakan untuk mengatasi segala kesulitan yang mungkin muncul. Salah satu keuntungan utama dari sistem manajemen pengetahuan adalah sebagai berikut :

1. Menghemat Biaya dan Waktu

Sumber daya pengetahuan yang ada dapat dengan mudah digunakan dalam konteks lain. Ini akan menghemat uang dan waktu.

2. Menambah Aset Pengetahuan

Karyawan dapat dengan mudah menggunakan sumber pengetahuan. Akibatnya, pemanfaatan pengetahuan meningkat, yang berarti kompetensi meningkat.

3. Perusahaan Lebih Siap untuk Adaptasi

Jika lingkungan bisnis mengalami perubahan, perusahaan akan lebih siap untuk beradaptasi. Ini karena

perusahaan memiliki pengetahuan tentang perubahan tersebut.

4. Meningkatkan Produktivitas

Pengetahuan dapat digunakan ulang dalam pembuatan barang yang sedang dikembangkan, meningkatkan produktivitas.

Untuk menyimpulkan, sistem manajemen pengetahuan (KMS) adalah metode penting untuk mengelola pengetahuan dan informasi organisasi. Dengan KMS yang efektif, organisasi dapat:

- a. Meningkatkan produktivitas: KMS memungkinkan akses yang lebih mudah ke pengetahuan yang relevan, sehingga anggota organisasi dapat bekerja lebih efisien.
- b. Mendorong inovasi: Dengan berbagi pengetahuan dan pengalaman, KMS memfasilitasi inovasi dalam organisasi, membantu dalam pengembangan produk, layanan, dan proses baru.
- c. Mendukung pengambilan keputusan: Informasi yang tersedia dalam KMS dapat digunakan sebagai dasar untuk pengambilan keputusan yang lebih baik dan lebih terinformasi.
- d. Meningkatkan kolaborasi: KMS memfasilitasi berbagi pengetahuan dan kolaborasi antara anggota organisasi, baik secara internal maupun eksternal.
- e. Meningkatkan pembelajaran organisasi: Organisasi dapat belajar dari pengalaman masa lalu dan menghindari kesalahan yang sama dengan memanfaatkan pengetahuan yang ada.
- f. Mengelola perubahan: KMS membantu organisasi dalam menghadapi perubahan dengan lebih baik

dengan menyediakan akses ke pengetahuan yang relevan tentang tren dan perkembangan baru dalam industri atau lingkungan bisnis.

- g. Meningkatkan efisiensi: Dengan memiliki akses cepat ke informasi yang diperlukan, KMS membantu organisasi menghemat waktu dan sumber daya.

Dengan demikian, implementasi yang baik dari Sistem Manajemen Pengetahuan dapat memberikan banyak manfaat bagi organisasi, membantu mereka menjadi lebih adaptif, inovatif, dan kompetitif dalam lingkungan bisnis yang terus berubah.

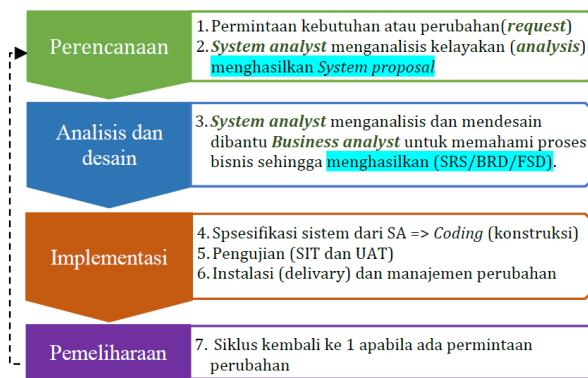


Pemodelan Proses Perangkat Lunak

Pada buku bagian ini akan membahas tentang pemodelan proses perangkat lunak yang mencakup dua elemen utama. Pertama adalah pemodelan dalam *Software Development Life Cycle* (SDLC) dan yang ke dua adalah terkait pemodelan perangkat lunak khusus pada tahap analisis dan desain. Di dalam SDLC memuat langkah-langkah seperti perencanaan, analisis, desain, implementasi, pengujian, dan pemeliharaan. Pada tahap analisis dan desain terdapat sebuah pemodelan yang dinamakan *Unified Modeling Language* (UML). Di dalam UML membahas lebih spesifik lagi yaitu terkait macam-macam diagram yang digunakan dalam memodelkan berbagai aspek sistem perangkat lunak, seperti *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*. UML inilah yang nantinya akan membantu tim pengembang perangkat lunak untuk berkomunikasi secara lebih efektif. Sekaligus dengan adanya UML ini nantinya akan tergambaran struktur serta perilaku suatu sistem dengan jelas.

A. Proses Pemodelan Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan sebuah disiplin ilmu yang mengintegrasikan sebuah proses, metode, dan alat bantu tertentu untuk memenuhi suatu kebutuhan yang dianggap lebih menguntungkan. Perangkat lunak dikatakan berkualitas bukan karena dikembangkan dengan teknologi terbaru atau *framework* terbaru, bukan pula karena dikembangkan dengan bahasa pemrograman berorientasi objek. Tetapi lebih kepada bermanfaat sesuai kebutuhan dan ada keuntungannya. Tentu Anda pernah menggunakan berbagai perangkat lunak yang canggih seperti pada *smartphone* ataupun komputer, namun tahukah bahwa di balik tampilan dan fitur-fitur yang kita nikmati itu terdapat proses panjang dan rumit yang membuat perangkat lunak tersebut menjadi suatu kebutuhan. Bagaimana sebenarnya proses perangkat lunak tersebut dibuat? Apa saja yang diperlukan seseorang untuk dapat membangun sebuah perangkat lunak?. Berikut kita pelajari tahap demi tahapnya bagaimana sebuah perangkat lunak menjadi ada dan kita gunakan seperti saat ini.



Bagan 1. Siklus Pengembangan Perangkat Lunak

Ketika ada satu orang melihat bahwa jika dengan bantuan perangkat lunak dapat menyelesaikan sebuah permasalahan, di situlah awal sebuah perangkat lunak dimulai. Kalimat kunci dalam tahap ini adalah mengapa perangkat lunak harus dikembangkan?. Pastikan dari tiga hal penting berikut salah satunya terpenuhi. Tiga hal itu adalah permintaan pembuatan sistem itu haruslah dapat mengurangi pembiayaan, meningkatkan produktivitas, atau dapat meningkatkan keuntungan. Jika pada tahap perencanaan ini seorang analis sistem menyatakan bahwa permintaan tadi layak, maka di tahap inilah lahir apa yang dinamakan *system proposal*. *System proposal* inilah yang nantinya digunakan untuk mengeluarkan *Down Payment* (DP) dari *client* untuk perangkat lunak yang akan dibuat.

Tahap selanjutnya adalah analisis dan desain. Beberapa ada yang memisahkan antara analisis dan desain secara tersendiri. Analisis berisi terkait siapa pengguna dan bagaimana alur kerja dari perangkat lunak yang akan dibuat. Sedangkan desain lebih kepada bagaimana komposisi dari perangkat lunak. Antara alur kerja dan komposisi memang sedikit dekat jadi ada yang menyatukannya namun ada pula yang memisahkannya. Menurut (Dennis et al., 2018) empat fase utama yang selalu ada di semua projek pengembangan perangkat lunak adalah *Planning*, *Analysis*, *Design*, dan *Implementation*. Di tahap analisis dan desain inilah lahir apa yang dinamakan *Software Requirements Specification* (SRS). SRS memiliki fokus pada persyaratan teknis dan fungsionalitas perangkat lunak yang dikembangkan, serta mendefinisikan secara rinci apa yang harus dicapai oleh sebuah perangkat lunak dari sudut pandang teknis. SRS memiliki kemiripan dengan FSD (*Functional Specification Document*) yang fokusnya lebih pada bagaimana fungsi-

fungsi ini akan diimplementasikan. Sedangkan ada pula yang menggunakan *Business Requirements Document* (BRD) yang memiliki fokus berbeda. BRD lebih fokus pada kebutuhan bisnis dan tujuan suatu organisasi untuk memotivasi pengembangan perangkat lunak. Jadi BRD lebih membahas mengapa projek ini diperlukan dan apa dampaknya bagi bisnis atau organisasi.

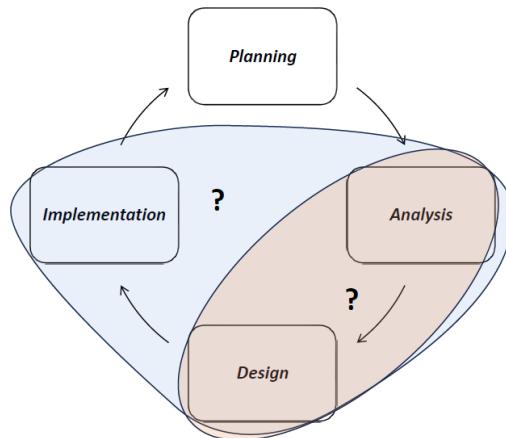
Tahap selanjutnya adalah tahap implementasi. Setelah diperoleh *system specification* dari tahap sebelumnya yang wujudnya bisa SRS/FSD/BRD dan juga *client* sudah setuju maka tahap selanjutnya *system specification* oleh *system analyst* diserahkan ke *programmer* untuk dilakukan *coding*. Kemudian hasil konstruksi berupa kode program dari *programmer* diserahkan ke *software tester* untuk dilakukan pengujian. Pengujian sendiri ada SIT (*System Integration Testing*) dan UAT (*User Acceptance Testing*). SIT biasanya dilakukan selama tahap pengembangan perangkat lunak secara internal oleh tim pengembang, setelah unit-unit individu telah diuji. SIT memastikan bahwa semua komponen perangkat lunak bekerja dan berfungsi dengan baik ketika diintegrasikan secara menyeluruh. Selanjutnya UAT biasanya dilakukan setelah tahap pengembangan perangkat lunak selesai dan sebelum peluncuran produk yang dilakukan langsung bersama pengguna akhir atau tim yang mewakili pengguna akhir. Mereka melakukan pengujian dengan situasi yang mendekati penggunaan sehari-hari. *Testing* ini dilakukan dengan tujuan agar memastikan perangkat lunak yang dibuat telah memenuhi persyaratan bisnis dan fitur-fitur yang ada telah berfungsi dengan baik serta pengguna dapat menggunakaninya dengan nyaman sesuai kebutuhan mereka. Setelah lolos uji barulah dilakukan instalasi (*delivery*). Sebuah perangkat lunak itu tidak hanya sekedar

kode program saja melainkan juga termasuk dokumentasinya. Jadi semua itu juga perlu diserahkan ke pihak *client*.

Setelah semua tahapan telah dilalui, beberapa pengembang perangkat lunak menyematkan tahap pemeliharaan. Tahap ini bisa jadi ada bisa pula tidak ada, tergantung kontrak di awal antara pengembang dan pihak *client*. Jika terdapat kontrak pemeliharaan maka pihak pengembang berkewajiban melakukan perubahan atau perbaikan masalah yang diajukan pengguna (*corrective maintenance*). Perubahan biasanya dilakukan sebagai bentuk penyesuaian terhadap perubahan lingkungan maupun perubahan pada platform perangkat keras maupun perangkat lunak yang digunakan (*adaptive maintenance*). Ada pula pemeliharaan preventif (*preventive maintenance*) yang secara rutin dilakukan untuk mengidentifikasi potensi masalah atau kerentanan keamanan yang bisa saja terjadi. Jika terjadi perubahan yang cukup besar terkait arsitektur, fitur utama, dan tujuan dari perangkat lunak secara signifikan maka tahapan perlu kembali lagi ke awal yaitu perencanaan.

Aktivitas yang berulang dalam suatu proses urutan tertentu seperti pada bagan 1 atau apa yang telah dijelaskan sebelumnya dapat kita sebut sebagai sebuah siklus. Proses ini sering dikenal sebagai "Siklus Hidup Pengembangan Perangkat Lunak" atau SDLC (*Software Development Life Cycle*). Untuk menghasilkan suatu produk perangkat lunak yang sama bisa jadi antara satu pengembang dan pengembang lain menggunakan proses yang berbeda. Suatu proses bisa jadi lebih cocok dari lainnya untuk beberapa tipe aplikasi tertentu, namun apabila penggunaan proses yang tidak tepat justru mengurangi kualitas produk yang dihasilkan.

Siklus itu adalah suatu cara seseorang mengembangkan perangkat lunak, sedangkan pendekatan yang seseorang gunakan untuk bisa menerapkan siklus itu sendiri dinamakan metodologi pengembangan perangkat lunak. Lalu bukankah bagan 1 di awal itu adalah *waterfall*? . Waterfall itu termasuk dalam metode atau metodologi?. Mari kita pelajari lebih jauh mengenai posisi *waterfall* dalam SDLC.



Bagan 2. Metodologi Pengembangan Perangkat Lunak

Alur dasar dari pengembangan perangkat lunak adalah seperti pada bagan 2. Apakah seseorang harus menyelesaikan tahap *planning* terlebih dahulu baru kemudian masuk ke tahap *analysis* dan selanjutnya *design*? . Atau tidak perlu menunggu?. Atau bahkan tahap *analysis*, *design*, dan *implementation* dijadikan satu dan berjalan secara bersama-sama?. Di situlah kita mengenal metodologi yang di dalamnya terdapat metode-metode. Metode-metode inilah yang biasa kita kenal dengan metode pengembangan perangkat lunak *waterwall*, *prototype*, *extreme programming*, *scrum*, dan lain sebagainya. Metode-metode itu dikumpulkan berdasarkan karakteristiknya masing-masing ke dalam sebuah metodologi pengembangan perangkat lunak.

Metodologi yang mengharuskan seseorang menunggu suatu tahap selesai baru kemudian bisa lanjut ke tahap selanjutnya dinamakan *structured design*. *Structured design* merupakan metodologi yang lebih mengutamakan dokumentasi dibandingkan komunikasi. Sedangkan sebaliknya metodologi yang lebih mengutamakan komunikasi dibandingkan dengan dokumentasi adalah *agile development*. Sementara metodologi yang berada ditengah-tenganya disebut *rapid application development* (RAD). Selengkapnya bisa dilihat pada bagan 3 berikut.

Structured Design	<ul style="list-style-type: none">➤ Waterfall➤ Parallel development
<ul style="list-style-type: none">– Gaya preskriptif,– Pendekatan linear,– Lebih mengutamakan dokumentasi,– Harus menunggu suatu fase selesai baru bisa lanjut ke fase selanjutnya	
Rapid Application Development	<ul style="list-style-type: none">➤ Phased development➤ Prototyping
<ul style="list-style-type: none">– Gaya iteratif– Fokus pada pembaruan dan perbaikan berulang selama siklus– Setiap iterasi dapat menghasilkan produk yang lebih baik atau lebih lengkap dari yang sebelumnya	
Agile Development	<ul style="list-style-type: none">➤ Extreme Programming (XP)➤ Scrum
<ul style="list-style-type: none">– Gaya adaptif,– Fleksibel dan responsif terhadap perubahan– Lebih membutuhkan komunikasi dibandingkan dokumentasi,– Kolaborasi antara anggota tim pengembangan sangat penting	

Bagan 3. Metodologi berdasarkan gayanya

RAD lebih cocok untuk projek yang memerlukan pengiriman cepat namun dengan desain awal yang kuat, sedangkan *agile* lebih cocok untuk projek yang sering

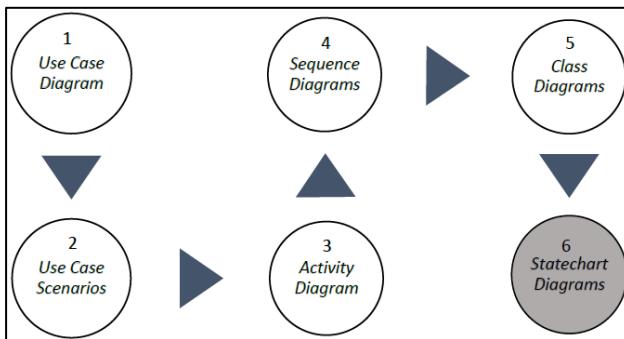
mengalami perubahan dan memerlukan fleksibilitas yang tinggi. Namun untuk *structured design* adalah sebaliknya lebih cocok untuk projek-projek yang memiliki lingkup terbatas dan tidak terlalu kompleks serta bisa ditangani secara individu.

B. Pemodelan pada tahap analisis dan desain

Proses pemodelan yang berada pada tahap analisis dan desain biasa disebut dengan dokumentasi pengembangan perangkat lunak. Dokumentasi ini dapat berupa diagram-diagram UML. Mengapa UML?, yaitu karena bahasa pemrograman yang digunakan berorientasi objek. Jika bukan maka menggunakan *flowchart*, DFD (*Data Flow Diagram*), dan ERD (*Entity-Relationship Diagram*).

1. UML (*Unified Modeling Language*)

Diagram pada UML yang sifatnya lebih menggambarkan perilaku sebuah sistem berkumpul pada tahap analisis. Diagram itu adalah *use case diagram*, *activity diagram*, dan *sequence diagram*. Sedangkan diagram yang bersifat statis berkumpul pada tahap desain. Contohnya adalah *class diagram* (Kendall and Kendall, 2020). Dari sekian banyak diagam-diagram yang ada di dalam UML tentunya timbul pertanyaan, apakah semua diagram wajib digunakan?. Diagram apa sajakah yang perlu ada, dan seperti apa urutannya?. (Hari Sayekti, 2023) menggambarkan urutannya seperti berikut.

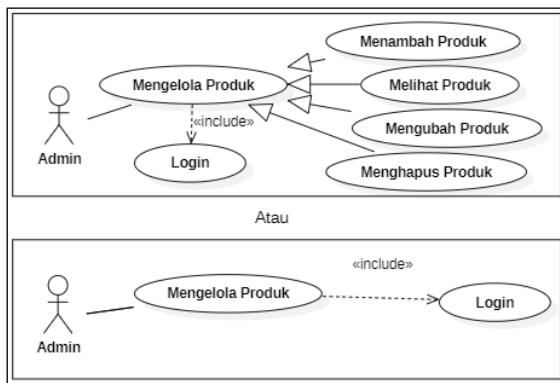


Bagan 4. Urutan proses pemodelan dengan diagram UML

UML adalah sebuah standar pemodelan, bukan metodologi dalam pengembangan perangkat lunak (Podeswa, 2010). Proses pemodelan dengan UML tidak harus selalu urut, tetapi urutan tertentu dalam pembuatan diagram UML akan sangat membantu dalam pemahaman dan komunikasi secara lebih baik. Pada UML beberapa diagram memiliki ketergantungan satu sama lain sehingga pengurutan menjadi penting. Setiap satu *use case* di *use case diagram* seharunya menghasilkan satu *activity diagram*. Selanjutnya di *sequence diagram* juga sama, berasal dari satu *use case*, meskipun bisa saja lebih dari satu *sequence diagram* untuk setiap satu *use case* yang ada.

2. *Use Case*

Use case merupakan ringkasan dari sebuah sistem yang isinya adalah apa yang dilakukan *user* terhadap sistem, bukan apa yang sistem lakukan. Untuk menggambarkan suatu sistem yang sama bisa jadi mengalami perbedaan pandangan antara satu orang dengan orang yang lainnya.

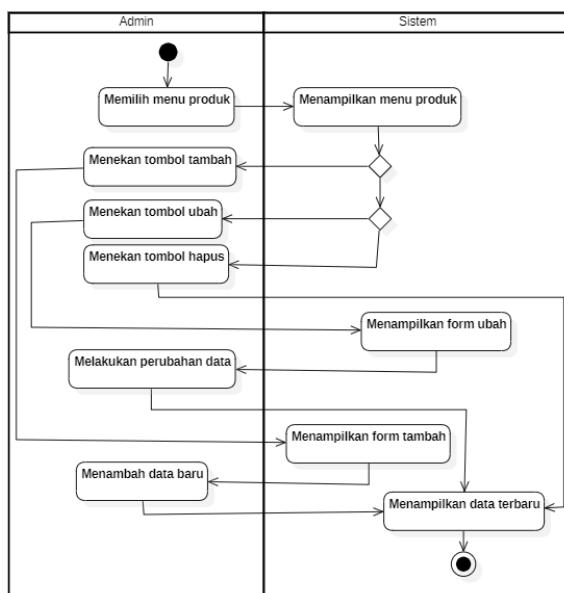
*Bagan 5. Contoh Use Case Diagram*

Dari bagan 5 terlihat sebuah *use case* dengan nama “Mengelola Produk”, bisa saja dipecah menjadi empat yaitu “Menambah Produk”, “Melihat Produk”, “Mengubah Produk”, dan “Menghapus Produk”. Jika kasusnya dipecah maka semua *use case* pecahannya perlu dijabarkan menjadi *activity diagram* satu per satu. Hal ini akan menjadikan *use case* yang dibuat menjadi semakin rumit. Buatlah *use case diagram* sesederhana mungkin namun tetap menggambarkan interaksi pengguna terhadap sistem karena hal-hal detail akan dijelaskan dalam *activity diagram*. Terkait arah panah untuk relasi *include* digambarkan dengan garis putus-putus yang berasal dari *use case* utama menuju ke *including use case* (bapak ke anak). Jika ada relasi *extend* arah panahnya adalah sebaliknya yaitu dari *extended*-nya ke *home*. Jika ada hubungan generalisasi arah panahnya dari khusus ke umum dengan garis solid (Sukamto and Shalahuddin, 2016).

3. Activity Diagram

Activity diagram digunakan dalam pemodelan sistem atau proses bisnis untuk menggambarkan urutan

tindakan yang terjadi (Hariyanto, 2004). Jika dibandingkan dengan *flowchart*, *activity diagram* dapat memodelkan sistem secara lebih luas. Sementara *flowchart* hanya spesifik untuk pemrograman. *Activity diagram* juga memiliki tingkat abstraksi yang lebih tinggi dan *flowchart* cenderung lebih menyajikan detail dari langkah-langkah sebuah proses atau algoritma. *Activity diagram* secara eksplisit juga mendukung aktivitas paralel dan sinkronisasi antar aktivitas yang ada. Jika *use case* “Mengelola Produk” tanpa generalisasi yang digambarkan pada bagan 5 tadi kita turunkan maka kurang lebih akan menghasilkan satu *activity diagram* seperti berikut.

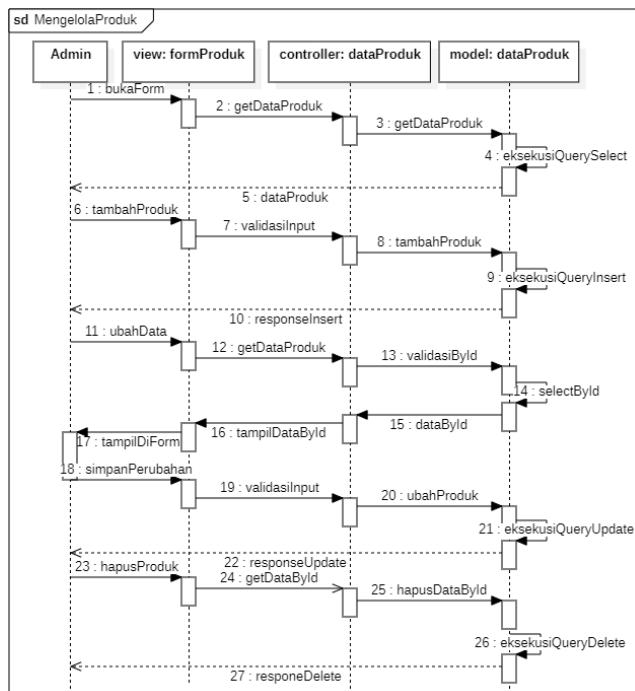


Bagan 6. Activity diagram mengelola produk

4. Sequence Diagram

Diagram ini memvisualisasikan bagaimana antar objek berkomunikasi dan berinteraksi satu sama lain

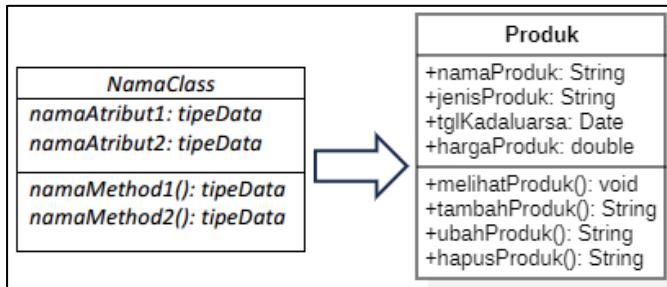
sepanjang waktu (Hariyanto, 2004). Di sini sudah terlihat kapan pesan dikirim dan bagaimana pesan itu dikirim, serta respon apa yang diberikan. Pada diagram ini, waktu berjalan dari atas ke bawah, sehingga urutan pesan dapat terlihat dengan jelas.



Bagan 7. Sequence Diagram Mengelola Produk

5. Class Diagram

Class diagram menggambarkan struktur sistem yang terdiri dari tiga area pokok yaitu nama, atribut, dan operasi/method. Diagram ini juga menggambarkan hubungan antar kelas seperti pewarisan, asosiasi, dan lain-lain.



Bagan 8. Contoh Class Diagram

Jika sudah sampai pada tahap ini biasanya seorang *programmer* sudah bisa menerjemahkannya ke dalam sebuah kode program sesuai kebutuhan. Selain itu, *programmer* juga harus memastikan bahwa hubungan antar kelas seperti pewarisan (inheritance), asosiasi, komposisi, dan lainnya telah diimplementasikan dengan benar sesuai dengan spesifikasi yang telah disepakati.

Daftar Pustaka

- Albright, S.C. and Winston, W.L. (2021) *Business Analytics: Data Analysis & Decision Making*. 6th edn. Cengage Learning.
- Ahmad, L., dan Munawir (2018). Sistem Informasi Manajemen: Buku Referensi. Banda Aceh: Penerbit Lembaga Komunitas Informasi Teknologi Aceh. ISBN 978-602-9451-03-0.
- Amin, R. (2017). Rancang bangun sistem informasi penerimaan siswa baru pada SMK Budhi Warman 1 Jakarta. *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, 2(2), 113-121.
- Afiifah, K., Azzahra, Z. F. and Anggoro, A. D. (2022) ‘Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database Sebuah Literature Review’, *Intech*, 3(2), pp. 18-22. doi: 10.54895/intech.v3i2.1682.
- Blaha, M. Rumbaugh, J. (2005) Object-Oriented Modeling and Design with UML. 2nd edn. Prentice Hall.
- Connolly, Thomas and Begg, Carolyn. (2010). Database Systems A Practical Approach to Design, Implementation, and Management Fifth Edition. Boston: Pearson Education
- Dr. Budi Raharjo, S.Kom., M.Kom., MM., Sistem Manajeman Database,(2021) Penerbit Yayasan prima agus teknik
- D. Y. Wurara, S. R. U. A. Sompie, S. D. E. Paturusi, and H. V. F. Kainde, “Rancang bangun aplikasi game pembelajaran

dan simulasi sistem bilangan digital berbasis android," *J. Tek. Inform.*, vol. 15, no. 1, pp. 13–22, 2020, [Online]. Available:
<https://ejournal.unsrat.ac.id/index.php/informatika/article/view/29026>

Dennis, A., Wixom, B. H., & Tegarden, D. (2015). "Systems Analysis and Design: An Object-Oriented Approach with UML." John Wiley & Sons.

Dennis, A., Wixom, B. H., & Roth, R. M. (2014). Systems analysis and design. John Wiley & Sons.

Dharmawan, W. S., Purwaningtias, D., & Risdiansyah, D. (2018). Penerapan metode SDLC waterfall dalam perancangan sistem informasi administrasi keuangan berbasis desktop. *Jurnal Khatulistiwa Informatika*, 6(2).

DosenPendidikan. 2021. DBMS (Database Management System).

Dennis, A., Haley Wixom, B., M. Roth, R., 2018. Systems Analysis and Design, 7th ed. Wiley.

Elmasri, N. (2015) Fundamentals of Database Systems. Pearson Education. Ltd.

Erickson, J. (2014) Impala Performance Update: Now Reaching DBMS-Class Speed. Cloudera Inc.

Frisdayanti, Alfriza.(2019) Peranan Brainware Dalam Sistem Informasi Manajemen. *Jurnal Ekonomi Manajemen Sistem Informasi*, 2019, 1.1: 60-69.

Gao, Chr. Jimmy L. (2008). Sistem Informasi Manajemen: Pemahaman dan Aplikasi. Jakarta: Grasindo. ISBN 978-979-025-104-5.

- Hari Sayekti, I., 2023. Pemrograman Berbasis Objek. In: Dasar-Dasar Pemrograman. Penamuda Media, Yogyakarta, pp. 170–180.
- Harahap, E. F., Adisuwiryo, S., Fitriana, R. (2022). Analisis dan Perancangan Sistem Informasi. Wawasan Ilmu.
- Hoffer, J. A., George, J., & Valacich, J. S. (2017). "Modern Systems Analysis and Design." Pearson.
- Hills, T. (2016) No SQL and SQL Data Modelling, Jurnal Penelitian Pendidikan Guru Sekolah Dasar.
- Hariyanto, B., 2004. Rekayasa Sistem Berorientasi Objek. Informatika, Bandung.
- Hutahaean, J. (2015). Konsep sistem informasi. Deepublish.
- Hartono, Jogiyanto. 2005. *Sistem Teknologi Informasi*. Edisi ketiga. Yogyakarta: Penerbit Andi.
- Inmon, W.H. (2005) *Building the Data Warehouse, Fourth Edition*. Wiley Publishing, Inc.
- Inmon, W.H., Terdema, D. and Imhoff, C. (2016) *Managing the Data Warehouse*. John Wiley & Sons.
- Jatnika, H. (2013) *Pengantar Sistem Basis Data*. 1st edn. Edited by P. Cristian. yogyakarta: Andi Offset.
- Kendall, K.E., Kendall, J.E., 2020. Systems Analysis and Design, 10th Edition, 10th ed. Pearson Education Limited, United Kingdom.
- Kustina, K. T., Nurhayati, M., Pratiwi, E., Lesi Hertati, S. E., CAPM, A. C. C. A. C., Qodari, A., ... & Abdul Munim, S. E. (2022). Sistem informasi manajemen. Cendikia Mulia Mandiri.

- Kurnia, R. (2023). Kamus Matematika Istilah, Rumus, Perhitungan. Nuansa Cendekia.
- Kimbal, R. and Ross, M. (2013) *The Data Warehouse Toolkit*. John Wiley & Sons, Inc.
- Laudon, Kenneth C.dan Laudon, Jane P. 2012. Management Information Systems - Managing The Digital Firm.12th Edition. Pearson Prentice Hall.
- McLeod, R., & Schell, G. (2014). "Management Information Systems." Pearson.
- Nugroho, A. (2004) *Konsep Pengembangan Sistem basis Data*. 1st edn. Edited by Adi Nugroho. Bandung: Informatika Bandung.
- Podeswa, H., 2010. UML for the IT Business Analyst, Second Edition: A Practical Guide to Requirements Gathering Using the Unified Modeling Language, 2nd ed. Course Technology PTR, Boston.
- Purnama, Chamdan (2016). Anam, Chairul, ed. Sistem Informasi Manajemen. Mojokerto: Insan Global.
- Pressman, R. S. (2015). Software Engineering: A Practitioner's Approach. McGraw-Hill Education.
- Putra, Y. M. (2020). Sistem Manajemen Basis Data. Modul Kuliah Sistem Informasi Manajemen. FEB-Universitas Mercu Buana: Jakarta
- Purnamafajari, R., Budiman and Niqotaini, Z. (2021) 'Design of Management Information System for Employee Performance Appraisal Using Service Oriented Architecture (SOA) Method', *IOP Conference Series: Materials Science and Engineering*, 1115(1), p. 012013. Available at: <https://doi.org/10.1088/1757->

899x/1115/1/012013.

- Sukamto, R.A., Shalahuddin, M., 2016. Rekayasa Perangkat Lunak: Terstruktur dan Berorientasi Objek, 4th ed. Informatika, Bandung.
- Simsion, G. dan Witt, G. (2005) *Data Modeling Essentials*. 3 ed. Morgan Kaufmann.
- Silberschatz, A., Korth, H.F., Sudarshan, S., 2020. Database System Concepts.
- Schwalbe, K. (2018). Information Technology Project Management. Cengage Learning.
- Senn, J. A. (2016). "Analysis and Design of Information Systems." Cengage Learning.
- Sommerville, I. (2019). "Software Engineering." Pearson Education Limited.
- Stamelos, I., Tsoukias, A., & Oikonomou, N. (2008). "A survey of software metrics education." ACM Computing Surveys (CSUR), 40(3), 1-44.
- Sudirman, A., dkk. (2020). Sistem Informasi Manajemen. Yayasan Kita Menulis. ISBN 978-623-6512-39-5.
- Sudiro, A., Rahadian, P., dan Prima, N. (2011). Sistem Informasi Manajemen. Malang: Universitas Brawijaya Press. ISBN 978-602-8960-45-8.
- Sutabri, Tata. *Konsep sistem informasi*. Penerbit Andi, 2012.
- Tyoso, Jaluanto Sunu Punjul (2016). Sistem Informasi Manajemen. Sleman: Deepublish. ISBN 978-602-401-312-7.
- Teorey, T., Lightstone, S. dan Nadeau, T. (2006) *Database Modeling and Design*. 4 ed. Morgan Kaufmann.
- Watt, A. and Eng, N. (2012) 'Database Design - 2nd Edition', *Creative Commons Attribution 4.0 International License*,

pp. 1–8. Available at: <http://open.bccampus.ca>.

Wijaya, R. E. (2021). Vernacular Accounting : Mengungkap Keunikan Komunikasi dalam Sistem Informasi Akuntansi. Penerbit Peneleh.

W. Stallings, *Syllabus Suggestions Computer Organization and a Rchitecture*. 2012.

Wijoyo, H., dkk. (2021). Akbar, Mada Faisa, ed. Sistem Informasi Manajemen. Solok: Penerbit Insan Cendekia Mandiri. ISBN 978-623-6090-57-2.

Yanto, R. (2016) *Manajemen Basis Data Menggunakan MySql*. DeepPublish.

Zatin Niqotaini, Indah Purnamasari, Cholid Fauzi, Yoga Sahria, Dartono, Dian Nursantika, I., Afriliana, Cahyo Prihantoro, Petrus Christo, Andi Wijaya, Anang Anggono Lutfi, M. and Robihul Mufid, Arif Rizki Marsa, Y.W. (2023) *Rekayasa Perangkat Lunak*. Edited by E. Mardiani. Yogyakarta: PT Penamuda Media. Available at: www.penamuda.com.

Zatin Niqotaini, S.Tr.Kom, M.K. et al. (2023) *ANALISA DAN PERANCANGAN SISTEM INFORMASI DENGAN UNIFIED MODELLING LANGUAGE*. Edited by M.K. Dudih Gustian. Bandung: INDIE PRESS. Available at: <https://indiepress.id/>.

<https://www.praxisframework.org/id/knowledge/information-management>

<https://accurate.id/marketing-manajemen/apa-itu-manajemen-informasi/>

Tentang Penulis



Arif Rizki Marsa, S.Kom., M.Kom Penulis menyelesaikan Pendidikan Strata 1 Program Studi sistem Informasi Konsentrasi sistem Analis dan Database pada Universitas Putra Indonesia "YPTK" Padang tahun 2010, menyelesaikan Pendidikan Strata 2 Program Studi Magister Teknik Informatika Konsentrasi Sistem Informasi pada Universitas Universitas Putra Indonesia "YPTK" Padang tahun 2013. Penulis adalah dosen tetap Program Studi S1 Informatika Sekolah Tinggi Teknologi Payakumbuh di payakumbuh. Pengajaran dan penelitian penulis berfokus pada bidang rekayasa perangkat lunak.yang membahas seputar pemrograman komputer.

Rosda Syelly, S.Kom., M.Kom. Penulis adalah dosen di Sekolah Tinggi Teknologi Payakumbuh (STT-Payakumbuh). Menyelesaikan Pendidikan pasca sarjana di Universitas Putra Indonesia "YPTK" Padang Sumatera Barat Bidang Sistem Informasi. Bidang keilmuan adalah sistem informasi. Besar harapan penulis agar tulisan ini dapat bermanfaat untuk kita semua dan dapat menambah wawasan bagi para pembaca. Terima kasih.



Riska Amelia, lahir di Jakarta pada 1 Mei 1986 dan sekarang menetap di kota Padang. Menyelesaikan Pendidikan Strata satu dan dua di Universitas Putra Indonesia “YPTK”.

Mulai bergabung menjadi Dosen Yayasan Bung Hatta sejak tahun 2010. Saat ini menjadi Dosen Tetap di Prodi Teknologi Rekayasa Komputer Jaringan, Fakultas Teknologi Industri, Universitas Bung Hatta.

Buku – buku yang telah dihasilkan diantaranya Pemrograman Berorientasi Obyek, Bangun Sendiri Website E-Learning Anda Dengan Moodle.



Ajang Sopandi, S.Kom., M.Kom, lahir di Subang, Jawa Barat. Penulis memperoleh gelar Sarjana dari Univeristas Islam Negeri Syarif Hidayatullah dan gelar Magister dari Universitas Budi Luhur Jakarta, saat ini Penulis sedang melanjutkan kuliah jenjang Doktor disalah satu kampus di Malaysia. Selain berkarier sebagai Tenaga Pengajar di salah satu Univeritas di Jakarta dan Tangerang, Penulis juga berkarir sebagai IT Project Manager pada salah satu perusahaan swasta yang ada di Jakarta dan menjadi Tenaga Ahli di DPR-RI.



Suharsono, S.Kom. M.Kom. Lahir di Dusun Alue Dondong I Aceh Timur pada 11 November 1988. Penulis merupakan lulusan SD Negeri Paya Palas Kab. Aceh Timur tahun 2000 kemudian melanjutkan pendidikan di SMPN 1 Kedung-

banteng Banyuman Jawa Tengah lulus tahun 2003.

Tahun 2006 setelah lulus dari SMAN Blang Barom penulis diterima melalui jalur Penelusuran Minat dan Prestasi (PMP) pada Program Studi Ilmu Komputer FMIPA Universitas Sumatera Utara dan lulus tahun 2010. Penulis melanjutkan Pendidikan Magister Teknik Informatika Fasilkom-TI di universitas yang sama dan mendapat gelar Magister Komputer tahun 2014.

Ketertarikan penulis dengan dunia komputer diperoleh sejak kuliah. Sedangkan minat menjadi seorang pengajar penulis peroleh ketika menjadi asisten laboratorium. Setelah lulus S1 penulis mengajar di SMK selama 11 tahun (2011-2022) dan mengajar sebagai Dosen di salah PTS di Medan dan Pematangsiantar. Terhitung Mei 2022 penulis menjadi Dosen PNS di Politeknik Negeri Pontianak hingga sekarang.

Email : suhar2006@gmail.com WA : 085296499123



Rifka Dwi Amalia, S.Pd., M.Kom. Lahir di Brebes, Jawa Tengah, 09 Januari 1990. Merupakan lulusan Magister Sistem Informasi Universitas Diponegoro dan anggota Ikatan Alumni MSI Undip yang turut serta aktif dalam keanggotan dan menggelar seminar dalam bidang ilmu komputer / informatika, sistem informasi, kecerdasan buatan, internet of things, elektro serta bidang lain yang terkait dengan ilmu komputer. Telah aktif dalam dunia Pendidikan dan Pelatihan khususnya bidang komputer untuk penempatan kerja di industri dan perkantoran dari tahun 2012 s.d 2017 dengan bergabung di LP3i Jawa Tengah. Saat ini merupakan Dosen di Program Studi S1

- Sistem Informasi di Universitas Pembangunan Nasional “Veteran” Jakarta. Bidang kajian yang diminati adalah Sistem Informasi Manajemen. Selain aktif sebagai penulis dan Dosen, aktif pula dikeanggotaan Aptikom Jateng, juga aktif sebagai peneliti. Penulis juga aktif menjadi Reviewer jurnal Nasional UPN Veteran Jakarta bidang komputer, serta Editor di Jurnal Nasional. Profil penulis dapat dilihat pada:

[http://www.linkedin.com/in/rifka-dwi-amalia-093b3594.](http://www.linkedin.com/in/rifka-dwi-amalia-093b3594)



Ranti Irsa. Penulis merupakan Dosen tetap di program studi Informatika di Sekolah Tinggi Teknologi Payakumbuh sejak tahun 2006, lahir dan berdomisili Payakumbuh, Sumatera Barat . Penulis menyelesaikan S1 dan S2 di Universitas Putra Indonesia, Padang Sumatera Barat.



Widiyawati, S.Kom , M.kom. Penulis lahir di Jakarta 14 Juni 1971 merupakan anak ke empat dari (Alm) bapak Zulsamin Sianip dan ibu Umbiyati , Menyelesaikan pemdidikan Strata 1 program Manajemen Informati, Universitas Gunadarma,pada tahun 1995 Penulis kemudian melanjutkan Pendidikan Strata 2 Program Studi Tehnik Informatika di STMIK ERESHA tahun 2014 . Pada saat ini penulis bekerja sebagai dosen Tetap Program studi Manajemen Informatika dari tahun 2003 sampai saat ini , diberikan kepercayaan mengajar mata kuliah Analisis

Perancangan Sistem Informasi dan Rekayasa Perangkat Lunak di Universitas BaniSaleh, Bekasi Jawa Barat. Penulis pun ikut serta dalam penulisan book chapter di penerbit Widya dan media sains Indonesia . Karya yang pernah di terbitkan adalah publikasi di jurnal internasional ter index Scopus dan Nasional terakreditasi .



Nurhuda Maulana memperoleh gelar sarjana dalam bidang Teknik Sistem Komputer dari Universitas Diponegoro, serta gelar magister dalam bidang Teknik Komputer dari Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung. Saat ini, beliau menjabat sebagai dosen di Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta. Fokus utama penelitiannya melibatkan Teknik Komputer, Sistem Tertanam, Jaringan Komputer, Internet of Things, dan Jaringan Sensor Nirkabel.



Nindy Irzavika, lahir pada tahun 1992 di Padang Panjang, sebuah kota kecil di Sumatera Barat, Indonesia. Penulis menyelesaikan pendidikan sarjana di Universitas Andalas, program studi Sistem Informasi pada tahun 2015. Kemudian melanjutkan pendidikan dengan meraih gelar magister di Institut Teknologi Bandung (ITB), program studi Informatika pada tahun 2019.

Selama kariernya, penulis pernah bekerja sebagai seorang analyst di WallStreetDocs, sebuah perusahaan internasional yang berbasis di London dan berfokus pada otomatisasi dokumen. Namun, saat ini, penulis bekerja sebagai seorang dosen di Universitas Pembangunan Nasional Veteran Jakarta (UPNVJ).



Musthofa Galih Pradana, M.Kom adalah seorang Dosen yang bertugas di Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta sejak tahun 2022. Mulai menjadi dosen pada tahun 2019 dengan *Area of Interest* di bidang Data Mining, Text Mining, Natural Language Processing dan Decision Support System.



Kharisma Wati Gusti, S.T.,M.T. Lahir di Garut, 19 Juni 1990. Penulis merupakan lulusan S1 Teknik Informatika Sekolah Tinggi Teknologi Garut (sekarang Institut Teknologi Garut). Dan S2 Informatika, konsentrasi Sistem Intelijen di Institut Teknologi Bandung. Saat ini penulis menjadi dosen Program Studi Informatika di Fakultas Ilmu Komputer UPN Veteran Jakarta.



Arif Budiman, S.Kom., M.Kom. Penulis adalah staf pengajar pada Program Studi Teknik Komputer, Sekolah Tinggi Teknologi Payakumbuh. Mengajar matakuliah Pemrograman Visual, Pemrograman Web dan basisdata. Pendidikan S1 dan S2 ditempuh di Universitas Putra Indoneisa, konsentrasi Sistem Informasi dan sekarang sedang melanjutkan pendidikan doktor di Universiti Malaysia Sabah mengambil jurusan Computer Science.



Zatin Niqotaini, S.Tr.Kom., M.Kom, lahir pada 10 Februari 1993 di Brebes, Provinsi Jawa Tengah. Penulis menyelesaikan pendidikan D4 Teknik Informatika di Politeknik Pos Indonesia Bandung dan S2 Sistem Informasi di STMIK Likmi Bandung. Penulis merupakan Dosen Pegawai Negeri Sipil di Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta dan homebasenya di Program Studi S1 Sistem Informasi. Bidang peminatan penulis berkaitan dengan Sistem Informasi Bisnis.



Imanaji Hari Sayekti, S.Pd., M.Pd. sejak 2013 mulai berkarir di dunia pendidikan dan industri *software development*. Ia menyelesaikan program sarjananya di Universitas Negeri Yogyakarta pada tahun 2013. Kemudian pada

tahun 2014 melanjutkan pendidikan magisternya di kampus yang sama yaitu Universitas Negeri Yogyakarta dan selesai pada tahun 2016. Selama menjalani pendidikan magisternya sampai dengan buku ini ditulis ia terjun langsung di dunia pendidikan maupun industri *software house*. Dalam kesehariannya ia bekerja sebagai staf pengajar di suatu kampus di Kebumen sekaligus sebagai team *programmer* pada instansi pemerintah di Kebumen. Ia memiliki kemampuan yang ideal dalam hal pendidikan maupun industri karena memiliki latar belakang dan pengalaman sebagai pengajar sekaligus sebagai *programmer* selama kurang lebih 10 tahun terakhir. Bagi pembaca yang ingin bertukar pikiran dengan penulis bisa menghubunginya melalui email di:

imanajihari@gmail.com

KONSEP SISTEM INFORMASI

"Konsep Sistem Informasi" adalah buku yang menggali aspek fundamental tentang bagaimana informasi dikelola dan digunakan dalam dunia modern. Buku ini membuka pintu pemahaman kita tentang pentingnya sistem informasi dalam berbagai bidang kehidupan.

Pertama-tama, buku ini membahas esensi sistem informasi. Sistem informasi adalah kombinasi dari teknologi, proses, dan manusia yang bekerja bersama untuk mengumpulkan, menyimpan, mengolah, dan menyebarluaskan informasi yang diperlukan dalam organisasi atau konteks tertentu. Ini mencakup perangkat keras, perangkat lunak, serta manusia yang menggunakanannya.

Selanjutnya, buku ini menjelaskan pentingnya data dan informasi. Data adalah fakta-fakta mentah yang memiliki potensi untuk menjadi informasi. Informasi, di sisi lain, adalah data yang telah diolah dan diinterpretasikan sehingga memiliki nilai untuk pengambilan keputusan. Buku ini mengilustrasikan bagaimana sistem informasi membantu mengubah data menjadi informasi yang bermanfaat.

Selanjutnya, buku ini mengulas berbagai jenis sistem informasi. Ini termasuk Sistem Informasi Manajemen (SIM), Sistem Informasi Eksekutif (SIE), Sistem Informasi Akuntansi (SIA), dan banyak lagi. Setiap jenis sistem ini memiliki peran dan fungsinya sendiri dalam berbagai tingkatan manajemen dan departemen dalam organisasi.

Selain itu, buku ini membahas konsep penting seperti pengambilan keputusan berbasis data. Pengambilan keputusan yang baik memerlukan akses yang cepat dan akurat terhadap informasi relevan. Buku ini menjelaskan bagaimana sistem informasi dapat membantu para pemimpin dan pengambil keputusan dalam mengevaluasi opsi mereka.

Buku ini juga membahas aspek keamanan informasi. Dalam era digital, melindungi informasi dari ancaman dan kebocoran menjadi sangat penting. Sistem informasi harus dilengkapi dengan mekanisme keamanan yang kuat, dan buku ini membantu pembaca memahami betapa pentingnya hal ini.

Selanjutnya, buku ini mengulas peran penting etika dalam penggunaan sistem informasi. Bagaimana data diproses, disimpan, dan digunakan harus mematuhi prinsip-prinsip etika yang ketat. Buku ini membimbing pembaca dalam memahami bagaimana menjaga integritas dan privasi data.

Terakhir, buku ini menyoroti tren terbaru dalam dunia sistem informasi, seperti analisis data besar (big data), kecerdasan buatan (AI), dan komputasi awan (cloud computing). Ini adalah konsep yang terus berkembang dan memengaruhi cara organisasi mengelola informasi.

"Konsep Sistem Informasi" adalah sumber yang berharga bagi siapa saja yang ingin memahami landasan sistem informasi. Dengan membaca buku ini, pembaca akan mendapatkan pemahaman yang kokoh tentang bagaimana sistem informasi memainkan peran sentral dalam dunia bisnis dan teknologi informasi saat ini.

ISBN 978-623-09-5871-7



PT Penerbit Penamuda Media
Godean, Yogyakarta
085700592256
@penamuda_media
penamuda.com