

Denne kolonne er
forbeholdt sensor.

Oppg. 1 a)

LAN er et nettverk som kan benyttes f.eks. innenfor en bedrift når denne er lokalisert på ett geografisk sted. Maskiner i et LAN-nettverk er knyttet sammen via kabler. (Derfor må alle maskinene være på ett sted). Har man f.eks. flere etasjer med kontorer ~~her~~ er gjerne maskinene i en etasje knyttet til en server som igjen er koblet sammen med en server i en annen etasje.

WAN ~~er~~ kan man si er en slags utvidelse av LAN. Hvis en bedrift har kontorer to steder og hver av disse har et LAN-nettverk, kan disse to stedene igjen knyttes sammen med WAN. ~~WAN~~ Man leier gjerne nett/kabel av et selskap, og denne går da mellom de to ulike geografiske plasserte stedene.

Internett er et nettverk av nettverk. Disse er knyttet sammen på mange ulike måter (trådløst, coaxial-kabel, fiber, telefonlinje, kabel-tv nett osv.) For at alle disse nettverkene skal fungere sammen må de bruke en protokoll (TCP/IP for internett) som sier noe om hvordan data skal overføres.

Denne kolonne er forbeholdt sensor.

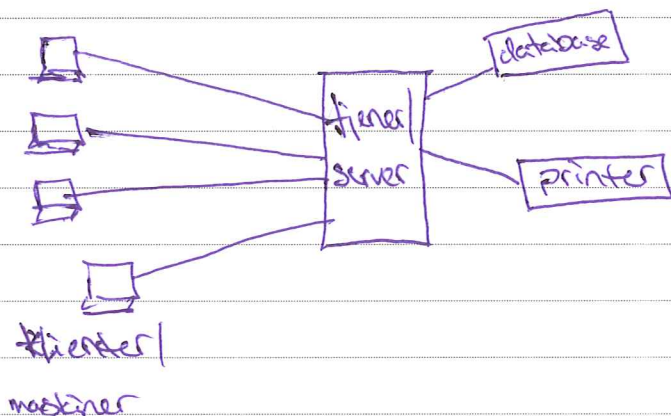
Oppg. 1 b)

1 og med et nettet ~~er~~ er knyttet sammen med kabel innen denne organisasjonen er det forbundet mindre sikkerhetsrisiko ved overføring av filer ol. Alt skjer lokalt innen organisasjonen.

Man kan knytte flere maskiner ~~til~~ til f.eks printere, scannere og andre maskiner innen organisasjonen.

Oppg. 1 c)

Klient / tjener arkitektur baserer seg på at informasjon skal være samlet på en server (tjener) slik at det kan være tilgjengelig for flere (klienter).



Flere maskiner (klienter) kan være knyttet til tjeneren. Slik kan kommunikasjonen gå innom tjeneren, denne kan formidle kontakt til f.eks felles databaser, felles printere osv.

Man skiller mellom "tykke" og "tynne" klienter.

Tynne er maskiner med kun skjerm og input-mulighet (tastatur, mus, scanner). Disse har ingen mulighet for å kjøre egne programmer / utføre egne

Denne kolonne er
forbeholdt sensor.

oppgaver, men kommuniserer kun med serveren.
(Eks: Kassasjapparat i butikk)

Tykt klienter er f.eks datamaskin på et kontor, der ~~bruker~~ brukeren har egne programmer på maskinen og kan utføre oppgaver på denne, samtidig som den kan kommunisere med serveren.

Oppg. 1a)

Klient/tjener og LAN ligner hverandre i struktur. De koder sammen ulike deler f.eks i en organisasjon og sørger for sammenheng og kommunikasjon mellom de ulike delene (maskinene).

Oppg. 2 a)

- Fordi IT og ny teknologi gir nye måter å lagre data på.
- IT øker muligheten for rask spredning av opplysninger.
- Spredning av data kan skje både vel og urett og intensjonelt (hacking)
-

Denne kolonne er
forbeholdt sensor.

Oppg. 2 b)

Fødselsnummeret kan brukes som middel for identifikasjon. Det kan brukes til å "stjale" identiteten til mennesker. Det kan brukes for å søke om kredittkort og lån. Det er med andre ord mange måter et fødselsnummer kan misbrukes på, og derfor er det ~~et~~ viktig at dette dekkes av personvernloven.

Oppg. 2 c)

Formålet med Personopplysningsloven er å sikre at ikke personopplysninger kommer i gale hender. Den skal sikre privatlivets fred, retten til å verne om private opplysninger og sikre mot ~~krenkelsen~~ misbruk av personopplysninger.

Hovedpunkter

- Alle som skal opprette et personregister har meldepunkt til Datatilsynet
- Dette skal gjøres ^{minst} 20 dager i forkant slik at Datatilsynet skal kunne vurdere om når og opplysningene ikke kan brukes eller om det inneholder sensitive personopplysninger.
- I tillegg av registre som inneholder sensitive personopplysninger, kreves det konsesjon
- ~~Skal brukes~~ Man skal melde fra om hva registeret skal brukes til, hvordan det skal lagres, sikkerhet, hva som registreres osv
- Skal brukes av opplysningene endres må det sendes inn ny melding

Denne kolonne er
forbeholdt sensor.

- Som sensitive personopplysninger regnes bl.a. politisk og religiøs overbevisning, medlem i fagforeninger og partier, legning, sykkelomshistorie osv.

Oppg. 2a)

Med innsynsrett menes at enhver som er med i et personregister har rett til å få se hva som er registrert om han/henne. Man har rett til innsyn i de registerne man er oppført i og har rett til å reservere seg mot bruk i enkelte registre. De som skal opprette personregistre har plikt til å informere de personer som ~~et~~ skal registreres personopplysninger om, om ~~oppsett~~ opprettelsen og hva informasjonen skal brukes til.

Oppg. 3a)

Hovedutfordringene ved systemutvikling er at det kan være vanskelig å etablere de viktigste ~~krav~~, alle/nok bruksspesifikasjoner / brukskrav. I store prosjekter kan det også være et problem at det blir samlet inn så mange krav at det blir så godt som umulig å holde styr på dem og å få tatt hensyn til alle. Det kan også være en stor utfordring å sette rammevilkår for tidsbruk og kostnader i og med at det ofte er vanskelig å se for seg omfanget av systemet og utviklingen. Dette er vanskeligere jo mer komplekst systemet er og særlig om brukerne ikke er sikre på hva slags

Denne kolonne er
forbeholdt sensor.

system de vil ha og hva det skal gjøre.

Oppg. 3b)

1) Etablere kravspesifikasjoner / rammer

- Her settes rammene for prosjektet, intervjuer av brukere etablerer kravene for systemet

2) Analyse / design

- Her kommer også intervjuer med brukerne inn. Det gjennomføres analyser for å finne ut hva systemet skal gjøre og hvordan det skal se ut. Designet henger sammen med analysen for å etablere hvordan designet kan være med på å understøtte det som er kommet frem til at systemet skal gjøre/være.

3) Koding

- Her ^{lages} kodes systemet etter de målene/kravene som er etablert i punkt 1 og 2.

4) Testing

- Her testes systemet når det er ferdig programmert. Test-fasen involverer brukerne av systemet og gjøres for å avdekke feil og mangler ved systemet. Test-fasen er meget viktig og tar ofte lang tid. Det er her viktig å få det testet / kjørt igjennom alle deler av kodingen for å avdekke evt. svakheter.

5) Implementering

- Her settes det ferdige systemet i bruk.

Dette kan gjøres på ulike måter.

I ~~første~~ installere det nye systemet i en liten del av organisasjonen. Hvis det her skulle vise seg å ikke fungere så vil det bare påvirke

Denne kolonne er forbeholdt sensor.

~~II~~ denne lille delen av organisasjonen.

II Kjøpe det gamle og det nye systemet parallelt. Dette er den mest tilkrevende måten å implementere på, i og med at brukerne må "jobbe dobbelt". Det er også den sikreste, i og med at man bruker det gamle systemet til man ser at det nye virker og dermed ikke trenger bekymre seg for evt. feil i det nye systemet.

III Plunge / hopp rett i det. Man kutter seg med det gamle systemet og starter med det nye. Denne måten anbefales ikke, fordi man her er totalt sårbar overfor feil i det nye systemet. Skulle ikke dette virke, står man helt uten noen ting til man får det gamle opp og gå igjen.

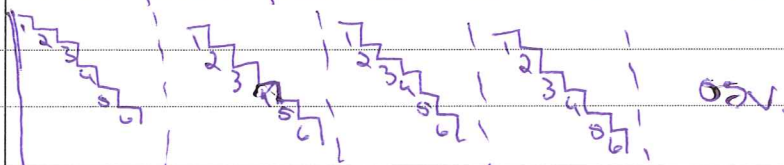
b) Bruk / vedlikehold

- Dette er den daglige bruk / drift av systemet med bruk for brukerstøtte, opplæring, reparasjon osv.

Oppg. 3c)

I fasedata metoden kjøres hver fase igjennom og avsluttes for man går videre til neste fase, og man går ikke tilbake.

I inkrementell utvikling lages systemet i deler / iterasjoner. I hver del lages en liten del av systemet ferdig til bruk for man går videre til neste del. Man kan si at det kjøres en mini-fasedata metode i hver iterasjon



Denne kolonne er forbeholdt sensor.

Man starter med de vanskeligste oppgavene i de tidlige iterasjonene. De aktivitetene man ikke fikk tid til i en iterasjon flyttes til neste.

Oppg. 3 d)

Hvis man har mye erfaring og vet ganske nøyaktig hva man vil oppnå / hva systemet skal gjøre kan det være greit å bruke trossfallsmetoden.

Hvis man derimot er usikker på hva systemet skal gjøre er ikke trossfallsmetoden å anbefale. Jo lenger man kommer i ~~utviklingen~~ utviklingen med trossfall, jo større blir kostnadene for å endre på systemet. Som regel B vil brukerne endre mening etter hvert, det kommer ny teknologi underveis o.l. Da er det bedre å bruke inkrementell utvikling. I og med at man lager systemet i iterasjoner vil det ikke være så problematisk å endre mening underveis.

I slike tilfeller kan det også være ok å velge prototyping. Her lages et skjelett av systemet og dette gir brukerne noe som er lettere å se for seg.

Prototypen kan også brukes i videre utvikling av systemet.

Oppg. 3 e)

Fordei det er vanskelig å estimere hvor lang tid det vil ta å lage systemet, og dermed vil det også bli vanskelig å estimere kostnadene. Mange prosjekter overskrider kostnadesestimertene med 100-200%.

Årsaken til at brukernes forventninger ikke blir

Denne kolonne er
forbeholdt sensor.

opptykt kan være at kravspesifikasjonene og/eller analysen er feil/mangelfulle og dermed har man laget hele systemet "etter feil mål".

Det kan også være at prosjektet har tatt for lang tid å utvikle, så når det endelig er ferdig er det allerede utdatert i forhold til ny teknologi.

Oppg. 4 a)

En database er et ^{strukturert} ~~stort~~ samling elektronisk lagrede data. Dataene er samlet inn for å brukes til samme formål og de har samme format.

Oppg. 4 b)

Vi kan ikke si at WWW er en database.

Det er riktig nok er samling elektronisk lagrede data, men dataene som ligger på WWW har verken samme format eller formål.

Oppg. 4 c)

Et DBMS er et system/program som brukes for å kommunisere med databasen.

Database / Det ligger mellom brukeren og databasen
DBMS / og sørger for at man kan opprette, sette
felter inn i og hente ut av en database.

Denne kolonne er
forbeholdt sensor.

Oppg. 5 a)

Et informasjonssystem består av 5 komponenter.

Hardware - software - data - ~~teknologi~~ ressurser - mennesker

Alle disse komponentene ^{må} samspille for at informasjonssystemet skal kunne være til nytte for bedriften.

Eksempler på IS:

Kønssystem

Beslutningsstøttesystem / ekspertsystem - samler data i en database og slik kan disse brukes senere til å beregne sannsynlighet er dermed hjelpe til å ta beslutninger. Eks: leting etter olje. Seismiske målinger lagres. Ny målinger kan sammenlignes med database og si noe om sannsynligheten for å finne olje.

ERP-systemer - systemer som samler flere forretningsprosesser i et system. Feks lønn, personalopplysnings osv. Hjelper til å ~~se~~ hindre at samme data blir lagret flere steder og sørger dermed for bedre bruk av systemer og ressurser.

Oppg. 5 b)

Forretningsprosess - et sett av aktiviteter og ressurser som skal være med å skape verdier for en organisasjon (interne og ~~eksterne~~ eksterne kunder) og hjelpe organisasjonen med å nå sine mål.

En verdikjede utledes fra forretningsprosessen. Den er de viktigste forretningsprosessene samlet og organisert for å skape verdier for organisasjonen og for ~~at~~ at den skal

Denne kolonne er
forbeholdt sensor.

Kunne nå sine mål på en effektiv og tilfredsstillende måte.

Oppg. 5c)

Beskriften kan skaffe seg konkurransefordeler ved omorganisering av forretningsprosessene og verdikjedene ved å benytte seg av ny teknologi for å kunne utnytte sine ressurser på en bedre og mer effektiv måte. Skal man ta i bruk ny teknologi / nye informasjonssystemer må man også endre forretningsprosessene og dermed også verdikjedene.

Eks: Når internett ble mer utbredt begynte bankene å lansere nettbanker. Ved å ~~flytte~~ gjøre det mulig å flytte mye av banktjenestene fra de bankansatte til at brukerne kunne gjøre det selv, ga dette en enorm effektivisering av både organisasjonene, og bruken av banktjenester for brukerne. De største bankene som var ute med et godt system for dette fikk da en god konkurransefordel fremfor de andre bankene.

Denne kolonne er
forbeholdt sensor.

Oppg. b a.)

HTML - er kode som brukes for å lage statiske nettsider. Denne brukte jeg i oblig 1 for å lage en hjemmeside om mag selv. HTML er et programmeringsspråk som gjør at nettleseren tolker det som står innenfor tagger på riktig måte. `<h1>...</h1>` sier at det som står innenfor skal være en overskrift-nivå 1. `LENKE` sier at dette er en lenke til en annen side (e.l.)

CSS - ~~er~~ Cascading Style Sheets. ~~Det~~ Dette er også et språk, men i disse filene sier man noe om hvordan nettsiden skal se ut. Her kan man bestemme skriftfarge, -størrelse, bakgrunnsfarge, punktliste utseende, lenkeutseende osv. Dette har jeg da altså brukt til å styre hvordan alt skal se ut på siden min, både farger og hvor bildene skal ligge. (img/flot: left?)
~~Det~~ Det kanner seg å ha alle CSS-kommandoene i en egen fil selv om det er mulig å sette CSS-kode rett inn i HTML ~~kode~~ kode også. Dette fordi det gir et mye ryddigere inntrykk oppsett og ikke minst fordi man ved å sette lenke fra HTML-sidene til CSS-filen kan styre utseende på alle filene samtidig i stedet for at man må endre utseende i hver HTML fil om man vil forandre noe.

PHP - er et programmeringsspråk som sørger for at man kan lage dynamiske nettsider. PHP koden tolkes også. Her kan man skrive programmer som kjøres på nettsiden og sørger for at det er mulig å bruke nettsiden til noe mer enn å bare se på.

Denne kolonne er
forbeholdt sensor.

Oppg. 6 b)

I min oblig 2 laget jeg en gjestebok.

De mest sentrale elementene jeg brukte var

SQL-spørring - spør mot databasen. Gjorde det mulig å sette inn i og hente ut fra gjesteboka.

Funksjonen hent - skrev en funksjon som skulle sørge for at alle innleggene ble hentet fram fra gjesteboka og vist. (Kjorte en løkke som er nesten helt lik den i oppg. 6c.) Når man lager en funksjon kan man bruke den hvor man vil bare den er definert en plass. (hent ();)

-Laget også en funksjon lagre(), som skulle sørge for at når folk skrev inn i gjesteboka ble dette lagret i databasen. Den så noe slikt som dette ut (med forbehold om enkelte småfeil...)

```
mysql-query INSERT INTO gjestebok ('navn', 'epost', 'melding');
```

↳ Setter innholdet i feltene navn, epost og melding inn i tabellen gjestebok.

Oppg. 6 c

1: velger alle deler fra tabellen gjest og sorterer dem etter når de ble registrert i synkende rekkefølge (nyeste først). Plasserer dette i variabelen \$query.

2: Kjører spørringen (mysql-query er en fast funksjon der at her kommer en SQL spørring mot databasen) fra \$query (den som ble lagret på linje 1) og lagrer resultatet av spørringen i variabelen \$resultat.

3: mysql-numrows teller hvor mange linjer den finner, her i fra spørringen som er lagret i \$resultat. Antall linjer finnes lagres i

Denne kolonne er forbeholdt sensor.

variablen \$ antall.

4: Gir en variabel (\$linjer) verdien 0, slik at denne kan brukes for å klare løkken.

5: starter while-løkken og sier at så lenge \$ antall (antallet linjer som ble felt på linje 3) er ~~større~~ større enn variabelen \$linjer skal løkken kjøre.

(6 og 11)-del som skal skje i en while-løkke må stå innenfor { } ellers tolkes bare den første linjen.

6: Setter ~~resultatet~~^{cert} som finnes når spørringen kjøres og linjer telles fra registrert-kolonne i databasen inn i variabelen \$dato.

7: Samme som linje 6 - bare med navnekolonnen til \$navn

8: Samme som 6 og 7 bare med kolonnen melding til variabelen \$melding.

9: printer til skjermen dato, navn og melding.

10: fortsettelsesbetingelsen for while løkken.

legger på 1 til \$linjenummer. Løkken kjøres til \$linjenummer er lik \$antall.

11: Når det har skjedd - stutter whileløkken.

(Oppg. 6d)

Absolute referanser betyr at en verdi er "låst fast" i en celle i et regneark. Vi lager absolute referanser enten ved å sette \$-tegn foran radnr og kolonne. (\$(\$3)) eller å navngi cellen. Absolute referanser brukes gjerne i formler hvis man har behov for å bruke dette tallet flere steder. Endrer man ~~den~~ verdien i den absolute referansen, endres også formelen.