

Denne kolonne er forbeholdt sensor.

## OPPGAVE ① Teknologi

a) Lokalnett kan ses på nesten som et intranett. Lokalnett er basert på et begrenset geografisk område. Gjente f.eks. en bedrift som StatoilHydro el.l. Dette er en stor bedrift som trenger et eget sikkert nett og med et LAN kan de ha et eget nettverk som knytter alle sammen i bedriften, uten at utenforstående kommer inn. Webserveren er i samme bygg.

WAN er en større utgave av LAN og er spreddt over et større område f.eks. StatoilHydro sine kontorer over hele Sør-norge og ikke bare en kontorbygning. Wan kan også ses på som en type intranett hvor man kan begrense nettet til tilatte deler. Serveren er off-site.

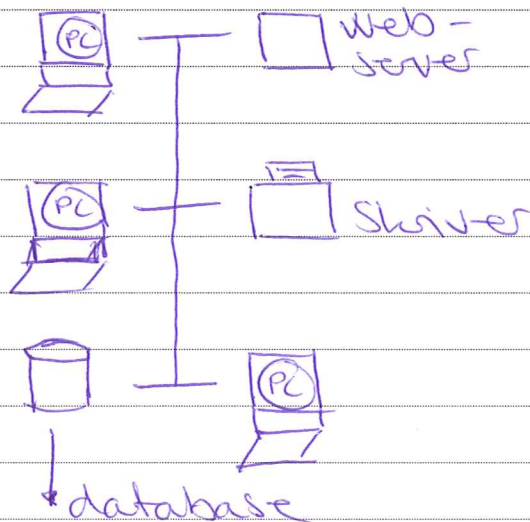
Internett derimot er knyttet til hele verden hvor alle kan legge til og ha tilgang til alt. Dette gjør at vi får en stor mer ubegrenset spredning av informasjon og ikke alle har gode hverdager på Internett med tanke på virus, ormer el. Dette gjør at bedrifter ofte foretrekker ett av de to overstående nettverkene.

b) Som nevnt i forrige avsnitt kan man foretrekke lokalnett fordi det er sikrere,

Denne kolonne er forbeholdt sensor.

man er tryggere fordi det ikke kan komme angrep utenfra like lett. Nettet vil i tillegg være mye kjappere i LAN og man er kun avhengig av seg selv.

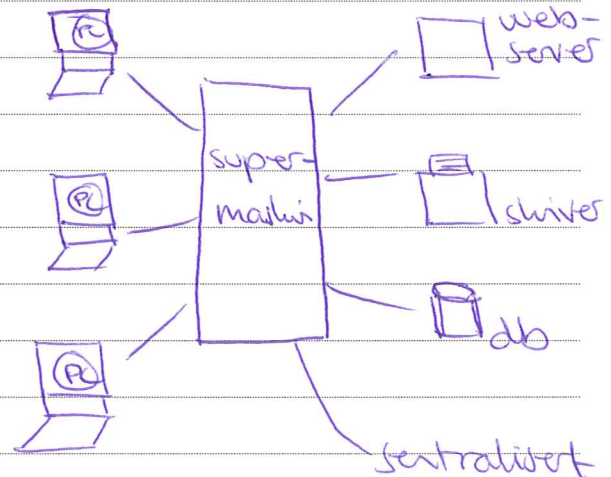
g) Klient/tjener system passer fint i bruk under nettverkstypene "Nettverk" og "Distribuert". Disse er satt opp slike:  
Distribuert nettverk:



Her er systemet bygd opp med at alle har utelligeste (ikke bare terminaler) maskiner som samarbeider om felles server, skriver db osv. Her blir PC'ene klienter og de andre tjenerne. Det mest vanligste/brukte systemet.

Nettverk:

Her er systemet satt opp nesten på samme måte som det distribuerte bortsett fra at alt må gjennom super-maskiner. PC'ene er klienter → Resten tjenerne.





Denne kolonne er forbeholdt sensor.

Klient/tjener-system er altså bygd opp som vist her, og et eksempel på et slikt program er "Novell". Noe som UiA har tatt i bruk. Fordelene med systemet er at alle har tilgang på felles database ol. men alle har ulikvekt eger maskinkraft.

- d) Klient/tjener krevger godt sammen med lokalnett fordi klient/tjener er et lokalt system som gir sikkerhet. Man bruker innlogging og kan på denne måten holde ute uvedkommende. Et lokalnett er dessuten bygd opp med tanke på å ha alt man trenger (av databaser, webserver, skrivere ol.) i selve bygningen/området og klient/tjener er tenkt for samme prinsippet.

## OPPGAVE 2) Personvern og Laggingning

- a) IT og personvern krevger nøye sammen fordi IT har forandret hvordan vi lagrer og sender informasjon. Personinformasjon lagret i et arkiv i Bergen opplevdes ikke som like sikkert lagret som noe som finnes på nett. Sikkerheten opplevdes som dårligere i sammenheng med IT og det er en lavere terskel for å stjele informasjon via teknologi enn å faktisk gjøre innbrudd.



Kurskode/Fag : IS-100  
Kandidatnr. : 3226  
Dato : 09.12.08  
Ark nr. : 4 av 17

Denne kolonne er forbeholdt sensor.

Et tillegg til planlagt tyveri av personlig informasjon via IT, har man uhell som ikke ville ha skjedd uten IT. For eksempel Tele2's uhell med personnr, som ble liggende tilgjengelig på deres nettsider.

b) Fødselsnr (henvist til som personnr i førte avsnitt) er spesielt interessant fordi det er på en måte inngangsbilletten. Det er nummeret som unikt identifiserer oss og som i mange (konstige alt for mange) tilfeller brukes som et slags brukernavn. Med dette brukernavnet kan man bestille og avbestille vidtvide ting, man kan bruke det som er nøkkel til å finne mye informasjon om en person. Dette er grunn til at fødselsnr er hemmelig og ikke skal gis ut til hvem som helst. Allikevel må det gis ut ofte i sammenheng med registreringer, betalinger, påmelding osv. Dermed er det etterhvert mange som har nummeret og det er viktig å kunne stole på at det ikke blir gitt til uvedkommende eller stjålet.

c) Personopplysningsloven er til for å gi personvern. Dvs. at man skal kunne være trygg på at personopplysninger blir behandlet på en riktig måte. Man skal vite hva slags informasjon som finnes



Denne kolonne er forbeholdt sensor.

om seg selv og ha en viss kontroll over hva der brukes til. Under personopplysningsloven har man noen viktige punkter:

- Konfeksjonsplikt: At man skal søke om konfeksjon (lov til) å samte inn, bearbeide eller formidde sensitive personopplysninger.
- Sensitive opplysninger: I følge loven er disse opplysningene regnet som sensitive: helseinformasjon, rolleblad, medlemskap i fagforeninger, selskvelle forhold, rasemessige ting og politisk tilhørighet.
- Meldteplikt: Man er pliktet til å melde fra om når man har terket til å samte inn, bearbeide eller formidde personopplysninger. Man må gi begrunnelse for hva det skal brukes til og man må gi beskjed 30 dager på forhånd. Skal man sende informasjonen noe sted skal også dette opplyses om. (Eks: når en mobilteleværdør sender ut brev om at en kredittspede er foretatt.) Skal brukes av informasjonen endres må man melde på nytt.
- Opphavsrett/årsverk: Man har full råderett over det man selv har produsert frem til død og 70 år etterpå.

Denne kolonne er forbeholdt sensor.

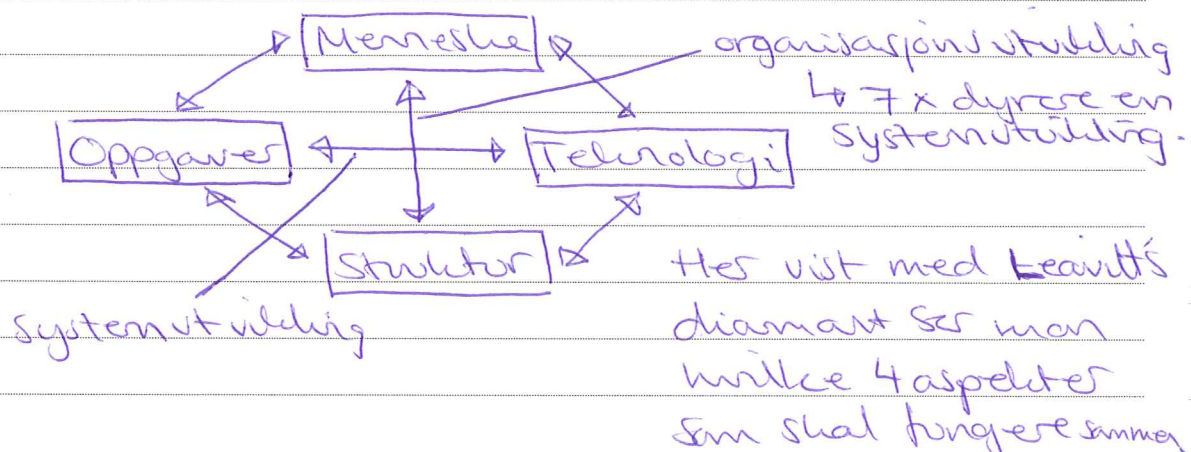
Opphavsretten gjelder de fleste typer produkter som f.eks musikk, film, skriftlige verk, kunst og programmer.

↳ Selv om dataprogrammer har en særstilling. Dessom man skriver et program mens man jobber i et firma tilfaller råderetten til arbeidsgiver, og ikke programmereren selv.

d) Innsynsrett mener at man som person har rett til å vite hvilke opplysninger som finnes om meg. Informasjonsplikt betyr at man er pliktet til å korrigere den informasjonen som finnes om meg. F.eks: i folkegjesteret.

### OPPGAVE 3) Systemutvikling

a) Hoved utfordringene i systemutvikling er i all hovedsak å få teknikkene til å fungere i samhold med menneskene.

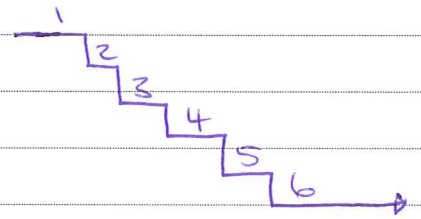




Denne kolonne er  
forbeholdt sensor.

Et annet veldig viktig aspekt av systemutvikling er å kartlegge hva kunder egentlig vil ha. Målet er å få det kunder ønsker og har behov for til å være så nært opp mot produktet som mulig. Meningen er å lage et system som fungerer bra og som brukerne forstår og kan bruke. Å beregne tid og kostnad for utvikling er også en utfordring.

- b) Hovedaktiviteter i systemutvikling:
- 1) Kravspesifikasjoner må kartlegges.
  - 2) Man må analysere behov
  - 3) Design systemet
  - 4) Kode systemet
  - 5) Implementere det
  - 6) Evalvere/teste.



- c) Fasefall arrangeres i samme rekkefølge som vist i forrige avsnitt i én sammenhengende periode. Fasefall begynner på punkt 1 og jobber seg gjennom uter å se seg tilbake. Alle faser blir kun gjennomført én gang og dette kan være over en kort eller lang tidsperiode - avhengig av prosjektets størrelse. En iterasjon altså. SCRUM har derimot flere iterasjoner. Denne metoden har de samme fasene som fasefall, men hele systemet gjentas. Dus at man kartlegger kravspesifikasjoner,



Kurskode/Fag : 15-100  
Kandidatnr. : 3226  
Dato : 09.12.08  
Ark nr. : 8 av 17

Denne kolonne er forbeholdt sensor.

analyserer, designer, koder, implementerer og tester systemet i flere omganger. Man lager små biter av systemet om ganger og viser disse til kunde som kan teste ferdige biter og kan tilbake melde synspunkter på disse uten at det vil være et problem om noe skulle trenge forandring. Alt foregår i iterasjoner som vil si korte tidsperioder. Dette kan være alt fra noen timer til et par uker, alt etter som hva man jobber med. På denne måten kan man gjenta prosessen helt til kunden er fornøyd, uten at det trenger å koste mye.

- d) Valg av systemutviklingsmetode er mye basert på hvor bestemt kunden er. Dersom kunden har full kontroll på hva som ønskes, hvordan det skal brukes, hvordan det skal se ut og oppføre seg så kan man bruke fossefall. Det finnes allikevel mange argumenter mot fossefall som f.eks: utrolig sjeldent at kunden vet 100% og holder seg til denne planen gjennom hele utviklingen, det er vanskelig å bestemme sånt 100%, teknologier kan forandre seg over utviklingsfiden osv. Derfor vil man i de fleste tilfeller gå for iterativ utvikling eller utvikling basert på iterasjoner (els SCRUM) eller



Denne kolonne er forbeholdt sensor.

prototyping. I alle disse tre siste er kunder mye mer involvert og har/hun har mulighet til å endre mening underveis og det blir lettere/mer naturlig å utvikle ideer samtidig som man jobber. Jeg ville nok personlig gått for en av disse (eller en kombinasjon) i systemutvikling, men man kan også si at disse 3 metodene passer best for kunder som ikke er helt sikre på hva de vil eller trenger.

e) Systemutviklingsprosjekt er blir ofte dyrere/mer tidkrevende enn forventet fordi det er vanskelig å treffe 100% hva kunden ønsker. Det er vanskelig å treffe kundens ønsker/behov, nye fordi vi har ulike mentale modeller og det er vanskelig for programmerer å sette seg inn i intrikate bestillingsrutiner for et oljefirma, og det er ikke vanskelig for en lagarsjef å vite/forstå hvordan ting fungerer i diverse programmeringsspråk. Kommunikasjonssvikt er kort sagt, det aller største problemet. Og når ting må gjøres 10 ganger etter at man er mye nærmere målet blir prosjektet naturligvis mer kostbart og tidkrevende. Andre ganger kan man risikere å ende opp med en løsning som er enda dårligere enn det som allerede finnes og velge å forkaste alt man har gjort til nå. Ehs. Tress 90



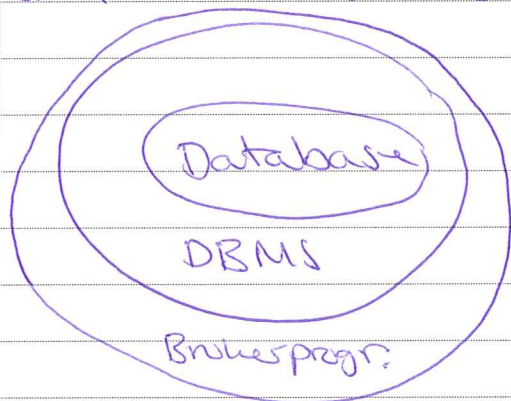
Denne kolonne er  
 forbeholdt sensor.

## OPPGAVE ④ Databaser

a) En database er en strukturert samling av elektronisk lagrede data. De er tilgjengelige via en datamaskin og har forhåndsgitte datatyper som systematiserer informasjonen som er lagret til ett felles formål.

b) Ved første tanke kan man kanskje si at WWW er en database fordi der er elektronisk lagrede data, men når man ser på resten av definisjonen er det tydelig at WWW ikke er en database. WWW er ikke strukturert særlig godt, det er ikke til ett felles formål og datatypene er definitivt ikke forhåndsgitte.

c) Data Base Management System er systemet som har "hovedansvar" for databaser.



DBMS er det som ligger nært opp mot databasen og denne delen har all informasjon om hvordan databaser skal "oppføre seg". Ekst. Datadefinisjon: Her ligger info om hva man kan ha i

datafeltene i databaser. At "Navn" bare kan inneholde bokstaver og maks 50 tegn f.eks. Datadefinisjonen legger begrensning på hva man kan fylle inn. Tall, bokstaver osv.

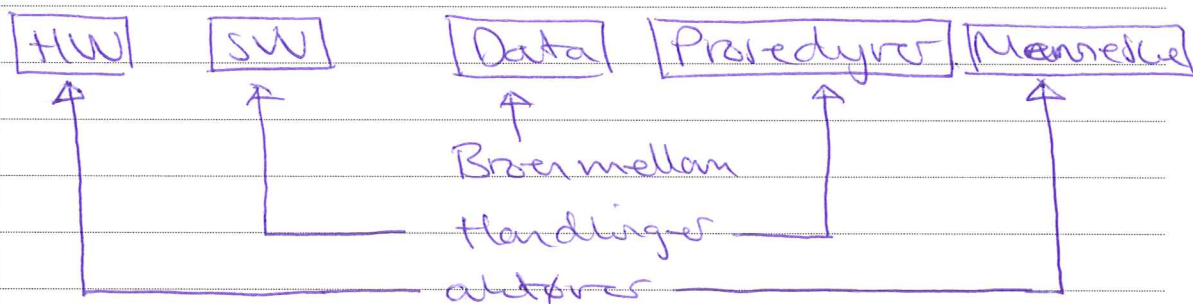


Denne kolonne er forbeholdt sensor.

DBMS har i tillegg integritetsbeskranking: som vil si regler for hva slags input databasen vil ha. Eks: at fødselsnr har 11 siffer eller at det bare kan være ett liket siffer. tall. DBMS og integritetsbeskranking har i tillegg ansvar for å gi eventuelle feilmeldinger til bruker. Utenfor DBMS har vi Brukerprogrammører som f.eks. gir kontrollert tilgang til de aspektene av databasen som er nødvendig. Eks: på DBMS er MySQL.

### OPPGAVE 5 Informasjonssystemer og Forretningsprosesser.

- a) Et informasjonssystem er et system som skal samle inn, bearbeide og formidle informasjon. Det består av 5 komponenter



- \* Hardware (HW): Systemets tekniske, fysiske bit'er. Eks: skrym, tastatur, harddisk ol.
- \* Software: Selve programmet som får HW til å fungere.

Denne kolonne er forbeholdt sensor.

- \* Data: For eksempel tall, ~~informasjon~~ beløp, bestillinger el.l. input egentlig.
- \* Prosedyrer: Det man gjør med systemet. kommandoer el. Fhs: å starte word el.l.
- \* Mennedje: Brukeren av systemet.

- \* Det finnes mange typer informasjonssystemer. Her er noen:
  - MIS: Management Information Systems
  - Beslutningsstøttesystemer
  - Ekspertsystemer
  - Kontorautomatiseringssystemer.
  - ERP-systemer. (Enterprise Resource Planning.)

- \* Eksempler på systemer kan være:
  - SAP er et typiske ERP-system. Det er et system som skal brukes i alle avdelinger av en bedrift og på denne måten ta seg av alt fra lønning til bestilling, sykefravær, osv.
  - Manuelt Datax er et fakturerings-system og kan kanskje kategoriseres som et MIS fordi det hjelper bedriften å nå mål.

- b) Forretningsprosessen vil si sammenhengen mellom starter fra forretningside til forretningsmål, produksjon og salg. Verdikjede henviser til hvordan verdien i forretning stiger mot slutten av forretningsprosessen.



Denne kolonne er forbeholdt sensor.

g) Bedrifter kan skaffe seg konkurransefordeler dersom de finner en hurtig måte å utnytte informasjonssystemer på. Man kan spare både tid og penger og allikevel utføre oppgavene bedre, mer nøyaktig og effektivt med det rette systemet.

- Et glimrende eksempel på dette som vi bruker nesten hver dag er nettbank. Med dette genialt systemet har bankene klart å få kunder til å gjøre hele jobber (nesten i alle fall) og allikevel kunne ta betalt.

↳ Dette er også et godt eksempel på BPR (Business Planning Reengineering) hvor man (som bankene har gjort) istedet for å bare automatisere eksisterende oppgaver heller har restrukturert hele måter og jobber på. Man kan da på denne måten gjøre mer på å tenke ut helt nye løsninger fremfor å bare fikse litt på de som allerede finnes. På denne måten får man utnyttet teknologier og de nye løsningene på en mye bedre måte.

Denne kolonne er  
forbeholdt sensor.

## OPPGAVE 6) Obligatoriske oppgaver.

a) HTML → Brukes for å lage nettsider. Her kan man strukturere tekstfelt og plassering av disse. Bilder, linker osv gjøres også med html. Man kan skrive koden selv eller bruke en kodegenerator. Eks: kompozer eller dreamweaver.

CSS → Brukes sammen med html for å definere utseende på nettsider. CSS kan sette farge, font, størrelse, plassering osv.

PHP → Gir liv til en ellers "død" nettside. PHP henter ting fra en database og presenterer dette når man trykker på noe. PHP kan f.eks. brukes til å personalisere en nettside.

b) Echo/print: Kommando for "vis på skjerm". Vi har mest brukt echo, men print betyr akkurat det samme.

Include: Betyr akkurat det som det høres ut som. Det inkluderer en fil el.l.

While-løkke: En funksjon som går om igjen og om igjen til et fastsett punkt. Ettersom man å sette slutt på denne går den evig. (eller til serveren væsjer.)



Denne kolonne er forbeholdt sensor.

Function: Metode for å hente/kalle en funksjon man har satt tidligere i koden. Ehs: en funksjon for å lagre noe når man trykker lagre-knapp.

If/else: Brukes for å sette kriterier for noe som skjer i koden eller en funksjon. Kan oversettes med: "Hvis" / "Hvis ikke" hvis sann og sann er sant → så skjer dette (else) så skjer dette.

Email: Funksjon for å sende epost. Her kan man definere hvem den skal til, hva som skal stå i fra-felt, meldings felt, emne-felt osv. Kan være nyttig om man skal fylle ut et skjema og ønsker å sende broker erwittering.

\* Alle elementene nevnt over er benyttet i min oppgave for å få skjemaet jeg laget mest mulig ~~funksjonelt~~ funksjonelt. Echo for å vise hvordan en twitteringside etter innfylling, include ble brukt mest for meg og enkelhet da jeg inkluderte en fil med brukernavn og passord til databaser for å slippe å skrive det flere ganger derom jeg bruket flere filer. While ble brukt for å hente fra databasen og function ble brukt for å



Denne kolonne er forbeholdt sensor.

strukturere funksjonene for f.eks. hente alle postene og vise på skjerm, lagre siste innlegg eller sende epost med kutting til både bruker og meg. Ifølge ble brukt for å sjekke om pålogging til databaser fungerer og om innsettning var vellykket og ~~epost~~ email ble brukt naturlig nok for å sende epost.

- g) ① Lager variabelen \$query ved å hente alt fra tabellen "gjest" og sortere dem etter "registrert"-feltet i synkende rekkefølge
- ② Lager variabelen \$resultat ved å bruke funksjonen query (som er en spørring) fra variabelen \$query.
- ③ Lager variabelen \$antall ved å telle antall rader i \$resultat. (som inneholder alt fra tabellen gjest.)
- ④ Setter \$linjennr til å starte på 0
- ⑤ Starter while-løp så lenge \$linjennr er mindre en \$antall.
- ⑥ Lager variabelen \$dato ved å bruke funksjonen result som der hentes fra variabelensresultat som er hele tabellen "gjest" innhold på \$linjennr fra felt "registrert" {viser starter på while
- ⑦ og ⑧ gir tilsvarende linje ⑥ men de lager variablene \$navn og \$melding istedet.
- ⑨ Viser på skjerm innholdet i variablene \$dato \$navn \$melding. (echo talter



Denne kolonne er forbeholdt sensor.

innholdet i variablene framfor å vise  
at det som står i koden fordi  
det står i doble quotes " " og ikke i  
single ' '.

⑩ Forteller at linjen skal øke med  
1 hver gang. Dette gjør at linje ⑥,  
⑦ og ⑧ henviser fra ~~nye~~ linjer i stedet  
for å bruke linje ④ om igjen og om igjen.  
Ettersom vi så på linje ④ at linjen er 0.  
⑪ Avslutter while.

d) En Absolutt referanse er en referanse  
som står stille. Den er fast varselt  
hvor man flytter formelen.

Eks: hvis man sier at  $A = \$A\$1 + \$A\$2$   
så vil formelen bevare hva som er  
i felt A1 + A2 varselt hvor jeg  
setter inn formelen og den vil ikke  
endre seg hvis jeg flytter der.

En absolutt referanse kan fastsettes  
ved bruk av \$ foran enten rad eller  
kolonne eller begge deler. - Eller  
man kan navngi celler. Da vil den  
også bli automatisk absolutt.

En absolutt referanse er det motsatte  
av en relativ som vil endre seg selv  
ettersom hvor man bruker formelen.

De brukes hvis man skal bruke en del  
av en formel flere steder f.eks. Eks  
hvis et prosenttall i C5 skal brukes  
flere steder.