Værstasjon i Hessdalen

HØGSKOLEN I ØSTFOLD

Avdeling for Informasjonsteknologi
Remmen
1757 Halden
Telefon: 69 21 50 00
URL: www.hiof.no

BACHELOROPPGAVE

Prosjektkategori: Bacheloroppgave	X	Fritt tilgjengelig
Omfgang i studiepoeng: 20	(30/12 2029)	Fritt tilgjengelig etter
Fagområde: Informasjonsteknologi	(X)	Tilgjengelig etter avtale med oppdragsgiver

Tittel: LaTeX mal for bacheloroppgaven	Dato: 3. mars 2014
Forfatterere: Mikael Johansen Grimstad, Kristoffer Jensen, Kristian Norum Karlsen og Morten Lindstad	Veileder: Einar Von Krogh
Avdeling / Program: Avdeling for Informasjonsteknologi (alle programmer)	Prosjektnummer: H13D42
Oppdragsgiver: Erling Petter Strand	Kontaktperson hos oppdragsgiver: Monica Kristiansen

Ekstrakt:
Det har vært en økende vektlegging på dokumentasjonen i bacheloroppgavene ved HiØ, slik at hoveddokumentet nå er grunnlaget for karaktersettingen. Formålet med dette prosjektet er å gjøre det enklere for studentene å produsere dokumentasjon med hensiktsmessig innhold, tradisjonell struktur, og profesjonell utforming. Rapporten starter med å redegjøre for generelle krav til vitenskapelige og tekniske rapporter. Det blir lagt spesielt vekt på kravene som stilles ved HiØ. Det gies en kort oversikt over hvordan man produserer og vedlikeholder dokumenter, både analoge og digitale. Deretter blir det utformet en mal som angir struktur og innhold i hoveddokumentet. Etter en ha utviklet en sett med minimumskrav til programvarene som skal brukes, blir det klart at kun to verktøy er aktuelle: LaTeX og *OpenOffice Writer*. En selvforklarende mal blir implementert i dokumentverktøyet LaTeX (en mer eller mindre identisk mal for OpenOffice er beskrevet i prosjektet *OpenOffice mal for bacheloroppgaven*).

3 emneord:

Foo
Bar
FooBar

Forord[1]

Dette er en mal beregnet til bruk i Bacheloroppgaven ved HiØ/IT. Malen gir en pekepinn om både struktur og innhold, og hvordan ting kan løses rent skriveteknisk, typisk ved å klippe og lime.

Malen er utformet som dokumentasjon på et fiktivt prosjekt, der formålet er å gjøre det lettere og enklere å dokumentere en bacheloropppgave (og liknende prosjekter). De fleste kapitler er innledet med generelle retningslinjer for hva som skal med (dette er uthevet i grått).

Det er tenkt at malen skal kunne brukes i alle de ulike prosjekttypene: utvikling, utredning og medieproduksjon. Dermed er mange overskrifter generiske, og må selvfølgelig tilpasses de enkelte prosjektene. Det kan også være aktuelt å slå sammen enkelte deler av malen, eller legge til kapitler.

Det er ikke obligatorisk å bruke malen.

[1] Dette forordet skal som du skjønner ikke med i det endelige dokumentet ☺

Sammendrag

Sammedraget er hele rapporten komprimert til max 1 side. Sammendraget skal gi leseren et godt og tilnærmet komplett bilde av innholdet i dokumentet. Akademiske sammendrag kalles på engelsk for “Abstract”, og i mer kommersielle sammenhenger heter det gjerne “Executive Summary”. I det siste tilfelle har sammendraget som hensikt å gi ledelsen i en bedrift nok informasjon til å ta økonomiske og/eller administrative avgjørelser. . . uten å lese hele rapporten (!). Tradisjonelt blir sammendraget formattert som et sammenhengende avsnitt. I et bachelorprosjekt, vil hovedformålet være å gi leseren (kanskje i første rekke sensor?) et informativt (og appetittvekkende) bilde av prosjektet. Det er ikke vanlig å bruke litteratur- eller kryssreferanser i sammendraget. Som en regel kan vi si at alt som står i sammendraget, kan det leses mer om i rapporten. Dermed blir utfordringen å belyse alle viktige hovedpunkter, kort og presist. For denne rapporten, kan det f.eks. bli som dette:

Takk Til

Det er vanlig, men ikke nødvendig, å nevne personer og miljøer som har hatt en positiv betydning for prosjektet, f.eks. på denne måten:

Innhold

Figurer

Tabeller

Kodeliste

Kapittel 1

Introduksjon

Det som er markert med grått, er forklaringer på hva de enkelte delene av rapportene skal inneholde, og som er det minimum leseren bør skumme gjennom før malen taes i bruk. Det som ikke er markert med grått, er eksempeltekst som kunne tenkes brukt i et fiktivt prosjekt der formålet er å utarbeide en mal for hoveddokumentet i en bacheloroppgave ved HiØ/IT. Introduksjonen skal gi leseren et bilde av rammene rundt prosjektet, prosjektets formål, metoder og leveranser. Den bør også inneholde en oversikt over resten av dokumentet[1]. Husk at kapitler, sections etc. bør ha et par setninger med "innledning" før man starter på neste undernivå.

1.1 Prosjektgruppen

Det er vanlig å starte med å presentere prosjektgruppen, litt om hver enkelt av deltagerne, deres kompetanse og interesser, og litt om hvordan dere har kommet sammen, f.eks. om dere har jobbet sammen i andre fag.

1.2 Oppdragsgiver

Beskriv oppdragsgiver, både firma og kontaktpersoner.

1.3 Oppdraget

Forklar hva slags problem oppdragsgiver ønsker å løse, og hvordan det er tenkt gjort. Dette skal ikke være en inngående analyse, men en utvidet og oppdatert versjon av det som opprinnelig fantes i prosjektbeskrivelsen, f.eks. slik:

n[1]. Erfaringsmessig hersker det endel forvirring om hva som skal med i dette dokumentet, hvordan det skal struktureres, og hvordan det bør se ut. I tillegg har vi tre ulike hovedvarianter av prosjekter:

[1] "Hoveddelen av karakteren utgjøres av sluttrapporten og prosjektets faglige nivå og utforming" `http://www.it.hiof.no/hovedp/evaluering2013.htm`

Utredninger: Dette er den vanlige formen i IT-ledelse. Her er det ikke snakk om å lage et produkt, men heller foreta en undersøkelse, f.eks. en evaluering av av et sett med programvare med tanke på hva som egner seg best for oppdragsgivers behov.

Mediaproduksjoner: I DMPRO består ofte prosjektene i større eller mindre grad å produsere en video, et radioprogram eller liknende.

Utvikling: På Informatikk, Dataingeniør og Webutvikling er det vanligst at oppdraget går ut på å laget et program, en app, eller f.eks. et nettsted.

Utfordringen i dette prosjektet er å lage en felles mal, både for innhold, struktur og utforming, som kan brukes i alle typer bacheloroppgaver. Malen bør implementeres med to ulike typer skriveverktøy. Det er et krav at malene skal være representert i åpne formater, og at de kan brukes på de fleste typer plattformer (minstekrav: Mac, Windows, Linux og beslektede).

1.4 Hvorfor, hva og hvordan: Formål, leveranser og metode

I dette kapittelet tar dere for dere tre sentrale aspekter: Formål, leveranser og metode. *Formålet* (ofte bare kalt *målet*, skal beskrive virkningen av prosjektet på et overordnet plan (f.eks. øke omsetningen i et firma). *Leveransene* er konkrete resultater (tangibles) som blir produsert underveis (f.eks. programvare med tilhørende brukerdokumentasjon), mao. *hva* som skal produseres. *Metoden* er *hvordan* formål og leveranser skal oppnås (f.eks. analysere dagens situasjon og designe og utvikle en ny nettbutikk). Jo mer teoretisk og "akademisk" prosjektet er, jo større vekt må man legge på metoden. Tradisjonelt er det metodiske aspektet relativt nedtonet i en bacheloroppgave i forhold til et master- eller PhD-prosjekt. Erfaringsmessig oppfatter studentene dette som en litt fremmed måte å betrakte et prosjekt på, men den er utbredt i både akademia og næringslivet, og gjør det lettere å holde tunga rett i munnen underveis. Formålet uttrykkes gjerne som ett hovedmål, og et par-tre delmål som utdyper hovedmålet. Beskrivelsen av et mål starter nesten alltid med et verb. I dette prosjektet kan formål, resultater og metode beskrives slik:

1.4.1 Formål

Hovedmål Gjøre det lettere og enklere å dokumentere en bacheloroppgave (og liknende prosjekter).

- **Delmål 1** Gjøre det lettere og enklere å få en god struktur på dokumentasjonen.
- **Delmål 2** Gjøre det lettere og enklere å få et godt innhold.
- **Delmål 3** Gjøre det lettere og enklere å få til en profesjonell og konsistent utforming.

1.4.2 Leveranser

Hovedresultatet av prosjektet vil være en *komplett, selvforklarende mal* for en bachelorrapport ved HiØ, tilgjengelig på HiØ sine nettsider, i to versjoner for to ulike skriveverktøy.

1.4.3 Metode

For å oppnå dette vil det bli utført en ad-hoc undersøkelse av krav og retningslinjer til bacheloroppgaver ved andre høgskoler og universiteter, både nasjonalt og internasjonalt. I tillegg vil forfatterens erfaring som sensor og veileder av bacheloroppgaver bli brukt som utgangspunkt for å lage en anbefalt struktur med tilhørende innhold. Videre vil det bli gjort en studie av omtaler og beskrivelser av verktøy for produksjon, vedlikehold og publisering av store og komplekse dokumenter.

Det vil bli lagt spesiell vekt på muligheter og erfaringer med å bruker maler. I tillegg vil sentral og kritisk funksjonalitet bli utprøvet ved de mest lovende verktøyene. På denne bakgrunnen (samt forfatterens egne erfaringer) vil det bli valgt ut to verktøy. Verktøyene vil til slutt bli brukt til å lage to maler, som tilbyr mer eller mindre samme funksjonalitet, og mer eller mindre likt utseende resultat. Malene skal utformes slik at de inneholder all nødvendig informasjon, både når det gjelder hvordan de skal brukes, hvordan rapporten skal struktureres, og hva slags innhold de enkelte delene skal ha.

1.5 Rapportstruktur

Det er vanlig å avslutte innledningen med en oversikt over resten av rapporten, f.eks. slik som dette:

I Kapittel 2 starter vi med å se på generelle krav til akademiske og tekniske dokumenter, og ikke minst hva som skiller disse fra "vanlige" dokumenter. Vi ser deretter nærmere på krav og retningslinjer til bachelorrapporter ved nasjonale og internasjonale læresteder. Vi går også gjennom HiØ/IT sine egen beskrivelse av hovedprosjektet. Vi gir eksempler på maler fra andre læresteder. Deretter ser vi på de tekniske sidene ved å produsere store og komplekse dokumenter, med spesiell vekt på aktuelle programvareverktøy. Vi presenterer en overordnet design av vår rapportmal i Kapittel 3, og beskriver den konkrete implementasjon i OpenOffice i Kapittel 4. Løsningen blir ad-hoc evaluert i Kapittel 5, og i Kapittel 6 diskuterer vi resultatet av prosjektet. Rapporten avsluttes med en kort konklusjon i Kapittel 7. En mer detaljert gjennomgang av hvordan malen kan brukes finnes i Vedlegg A.

Kapittel 2

Analyse (Generisk tittel)

Kapittelet tar for seg analysedelen av arbeidet. Den består av to hoveddeler, en grundig beskrivelse av oppgaven basert på skissen gitt av oppdragsgiver, og en undersøkelse av hva som finnes av relatert arbeid, *best practise* og relevant teknologi.

I dette kapittelet danner vi oss et bilde av hvordan en akademisk/teknisk rapport bør se ut, både med hensyn på innhold, struktur og utforming. Vi ser også på hvordan store og komplekse dokumenter blir produsert (både analogt og digitalt), og hva slags (digitale) verktøy som kan tenkes å brukes. Vi går også gjennom endel tilgjengelig materiale som kan gjøre det lettere å velge hva slags digitale verktøy man burde bruke i en bacheloroppgave[1].

2.1 Akademiske og tekniske dokumenter

Akademiske og tekniske dokumenter ser noe forskjellige ut avhengig av fagfeltet. Likevel, en litt grundigere analyse avslører en felles hovedstruktur, som egentlig er ganske intuitiv. . . og omtrent lik den vi lærte på barneskolen: *Hode, kropp og hale*. Hver av disse hoveddelene består som regel av omtrent de samme elementene.

2.1.1 Mayfield Handbook of Technical & Scientific Writing

I følge engelsk terminologi deler man gjerne et dokument opp i *Front matter, Body og End (Back) matter*, og det gjøres konsekvent i Mayfield Handbook of Technical & Scientific Writing [**?**]. Denne utmerkede kilden finnes også på elektronisk form[2]. I kapittelet *Elements of Technical Documents*[3] blir den følgende generiske strukturen foreslått:

- Front Matter
  - Title page
  - Abstract
  - Table of contents
  - List of figures
  - List of tables
  - List of terms
  - Acknowledgments

[1] Utvalget er noe overfladisk utført, i hovedsak basert på forfatterens erfaringer.

[2] `http://www.mhhe.com/mayfieldpub/tsw/home.htm`

[3] `http://www.mhhe.com/mayfieldpub/tsw/elemtech.htm`

- Body
  - Introduction
  - Background
  - Theory
  - Design criteria
  - Materials and apparatus
  - Procedure
  - Workplan
  - Results
  - Discussion
  - Conclusion
  - Recommendations
- End Matter
  - References
  - Appendixes
  - Index

Hvert av punktene blir nærmere beskrevet, med mange eksempler på godt (og dårlig) innhold, og det anbefales på det sterkeste å studere dette kapitellet nærmere.

Denne strukturen er så nær man kommer en universell mal for en teknisk/vitenskapelig rapport.

2.1.2 HiØ/IT

Ved HiØ/IT har man valgt å skille grovt sett mellom to typer dokumentasjon. *Prosessdokumentasjonen* omfatter en forprosjektrapport samt individuelle refleksjonsnotater, og selve produktet er omtalt i *hoveddokumentet*[4](eller rett og slett det vi kaller bacheloroppgaven). Når det gjelder oppbygging og innhold er hoveddokumentet å regne som en typisk akademisk/teknisk rapport/artikkel:

- Tittelside
- Sammendrag
- (Takk til)
- Innholdsfortegnelse
- (Liste over figurer)
- (Liste over tabeller)
- Introduksjon
- Analyse
- Design
- Implementasjon
- Evaluering
- Diskusjon
- Konklusjon
- Referanser
- Referanser

[4] `https://wiki.hiof.no/index.php/Bacheloroppgaven_-_Leveranser`

2.2 Skriveverktøy for store og komplekse dokumenter

For å starte på et stort og komplekst dokument, vil mange åpne sin vanlige teksteditor og sette igang. Det går sjelden bra. Som ved alle typer større oppgaver, bør man velge sine verktøy med omhu. Før man setter igang å lage et digitalt dokument, kan det være fornuftig å dvele litt ved hvordan dokumenter har blitt, og blir, produsert på tradisjonelt vis.

2.2.1 Forfatter, redaktør, typograf, trykkeri

Tradisjonelt hersker det en streng arbeidsdeling innen produksjon av dokumenter, som f.eks. en bok, eller en avis. Forfatteren (eller journalisten) skriver såkalt brødtekst, som rett og slett er tekst, skrevet for hånd, på skrivemaskin, eller talt inn på diktafon. I teksten er det gjerne markert ulike strukturelle elementer. Det er markert hvor et nytt avsnitt begynner, kanskje det står skrevet at her kommer det en illustrasjon, etc. Dette råmaterialet går gjerne til en redaktør, som kvalitetssikrer innholdet og tildels struktur, og eventuelt foreslår/foretar endringer. Deretter tar typografen (eller den grafiske formgiveren) over, og bestemmer skriftstørrelser, fonter, marger, linjeavstand etc. Til slutt går trykkoriginalen til trykkeriet[5].

Denne framgangsmåten ivaretar en klar tredeling av arbeidet; innhold, struktur, og form. Innholdet, og grovstrukturen, bestemmes av forfatteren. Redaktøren sørger for å kvalitetssikre innholdet, og eventuelle justere strukturen. Den endelig formgivingen av dokumentet blir typografens rolle. Det er viktig å merke seg at alle disse tre rollene krever høy kompetanse innen helt forskjellige fagfelt. De fleste mener også at det er ganske unaturlig for en forfatter å legge ned mye tankearbeid i hvilken font og størrelse det bør være på overskriftene. På den annen side ville vel det ikke være rett at typografen blandet seg bort i innholdet.

2.2.2 The curse of the cursor -eller- DTP: DeskTop Publishing eller Dom Tokigas Paradis?

Og etterhvert kom datamaskinene, som tilsynelatende kunne gjøre "alt", og dermed begynte mange å gjøre "alt", selv. Teksteditorer og desktop publishing verktøy gjorde det mulig for at en person kunne stå for hele prosessen, både skrive, redigere, og formgi, uten å ha særlig kompetanse, ferdighet eller erfaring i noen av feltene.

Personlig husker jeg godt mitt første møte med DTP, tidlig i IT-utdannelsen. Oppgaven var å lage en liten avis, med bilder og det hele, ved hjelp av en søt liten Mac, som på et mysteriøst vis var knyttet sammen med en skriver som til og med skrev ut illustrasjoner og fotografier i grov oppløsning. Den blinkende markøren i teksteditoren var hypnotisk, halvt innbydende, halvt skremmende. Og så måtte man bare kaste seg ut i det. Skrive noen setninger, markere en del som "bold" og fontstørrelse 18, fin overskrift, et par "enter", forsette med vanlig font, nei, kanskje en litt mer fancy kalligrafifont, kanskje til og med blå? Og dermed var det gjort, forvirringen var total. Noe senere erfarte jeg at i enkelte svenske miljøer mente man at DTP sto for "Dom Tokigas Paradis". Innen visse segmenter av reklamebransjen mener jeg fremdeles å se spor etter slike traumatiske møter med datastøttet publisering.

Man kan jo spørre seg om det egentlig er noen forskjell mellom et blankt ark, enten på et skrivebord eller i en skrivemaskin, og en blinkende markør øverst i venstre hjørne i en editor? Uansett svar, det blanke arket gir deg ikke anledning til å eksperimentere med fonter, marger,

[5] Her kan jo være på sine plass å tenke litt på XHTML og CSS, og hvor de har sine røtter... og da tenker jeg ikke på Håkon Wium Lie ☺.

skriftstørrelse og plassering av bilder. Arket handler kun om innhold, og hvordan det skal struktureres, hvilken passasje som følger en annen, og hvor det begynner et nytt avsnitt.

2.2.3 Programmering av dokumenter

Et par år senere skal jeg produsere min første akademiske artikkel, i LaTeX med AMS stilen. Hæh?

Det tok et par dager før jeg skjønte poenget. Skrive innholdet i filer (ark) i en editor uten formateringsmulighet, sette inn kryptiske kommandoer for å markere avsnitt eller matematiske uttrykk, sette sammen delene i en styrefil, legge til noen kommandoer som avgjorde hvordan dokumentet ble seendes ut. Deretter *kompilere* hele greia! Som så resulterte i en DVI-fil[6], som igjen kunne konverteres til alle mulige trykk- og printbare formater. Og det beste av alt, resultatet, rent formmessig, ble helt strøkent, akkurat som i lærebøkene og artikkelsamlingene.

Dette var jo akkurat som å programmere. Full kontroll.

2.3 Digital dokumentproduksjon

Vi foreslår et sett med minimumskrav til digitale dokumentverktøy, basert på den forutgående analysen. Deretter gir vi kort beskrivelse av LaTeX og OpenOffice Writer, som etter en kort vurdering er de to verktøyene som tilfredstiller de fleste av kravene.

2.3.1 Generelle krav

Kravene er utarbeidet delvis med basis i den forutgående analysen, men i hovedsak ut fra forfatterens erfaringer.

Plattformuavhengighet I et prosjekt med flere deltagere, er det direkte arrogant å forvente at alle skal bruke samme plattform, fordi valget av dokumentasjonsverktøy krever dette.

Robusthet En hovedrapport er ofte relativt stor, og tildels med komplekst innhold (figurer, tabeller etc). Det er ikke uvanlig at selv kostbare og vidt utbredte verktøy kræsjer i en slik kontekst, eller verre, begynner å oppføre seg inkonsistent og uforutsigbart Det å åpne et dokument, bare for å oppleve at formatering som ble foretatt kvelden før er ødelagt, er ingen særlig givende erfaring, særlig ikke hvis det er to timer til rapporten skal leveres inn.

Maler/stiler Det må kunne være mulig å skifte mal på et og samme dokument uten å gjøre ekstra tilpasninger. Dermed oppnår man et klart skille mellom innhold og utforming. Hvis f.eks. retningslinjene for hovedrapporten endrer seg, så er det bare å lage en ny mal og bruke denne.

Oppsplitting Det bør kunne være mulig å splitte dokumentet i mindre deler, f.eks. en fil for hvert kapittel. Delene må kunne knyttes sammen i et masterdokument. På denne måten vil det være lett å endre rekkefølgen på komponentene. Da er også strukturen uavhengig av det faktiske innholdet. Oppsplitting er også en forutsetning for å kunne jobbe distribuert, der de ulike deltagerne jobber parallelt på ulike deler av rapporten.

Kryssreferanser Det må være mulig å bruke dynamiske kryssreferanser, slik at man f.eks. kan referere til en nummerert figur. Hvis da nummeret til figuren endrer seg pga. av at det blir lagt inn en figur tidligere i dokumentet, så må referanse oppdateres automatisk.

Bibliografi De må være enkelt å referere til en ekstern samling bibliografiske referanser. Det også være enkelt å når som helst endre referansestil (f.eks. IEEE eller Harvard).

[6] DVI: DeVice Independent

Brukermasse Det bør eksistere en stor og variert brukermasse, noe som i seg selv kan være en antydning om at verktøyet holder mål.

Dokumentasjon Verktøyet må være godt dokumentert, både den innebyggede dokumentasjonen, samt i ulike fora på nettet.

Versjonskontroll Det må være enkelt å håndtere versjonering, dvs. at man lett kan rekonstruere tidlige versjoner, og lett se hva som er endret fra versjon til versjon.

Framtids/fortidssikker Verktøyet bør kunne uten videre behandle dokumenter som er laget i tidligere versjoner. Dette vil også være en indikasjon på at dokumentet du jobber fremdeles "lever" i rimelig overskuelig framtid.

Stave/grammatikkontroll Minimumskravet er at det er mulig å utføre effektiv stavekontroll, for ulike språk, ikke minst norsk bokmål. I tillegg vil det være et pluss om det er mulig å verifisere grammatikk.

2.3.2 LaTeX

Klipt og limt fra Wikipedia[7]:

> LaTeX er et typesettingssystem for dokumentproduksjon. Det skrives normalt LaTeX i dokumenter som ikke har hevet og senket tekst. LaTeX ble opprinnelig utviklet av Leslie Lamport i 1984 og tilbyr idag en lang rekke dokumentproduksjonspakker som f.eks. automatisk opprettelse av innholdsfortegnelser, lister med tabeller og figurer, inkludering av bilder, kryssreferanser, sideoppsett, bibliografier mm.
>
> LaTeX baserer seg på at forfatteren av et dokument kun skal være forfatter av dokumentet og slippe å bry seg om utforming og hvordan ting ser ut; noe han normalt sett ikke er ekspert på uansett. Det forfatteren gjør er å dele opp dokumentet i logiske strukturer før han lar LaTeX ta seg av selve oppsettet. LaTeX ansees ofte som WYSIWYM (det du ser er det du mener) fremfor WYSIWYG (det du ser er det du får) editor. Dette fører til at det ofte anerkjennes som overlegent fremfor andre skriveprogrammer grunnet at endringer som påvirker hele dokumentet er meget enkle å gjennomføre. De trenger ofte bare en endring et sted, så er endringen gjennomført i hele dokumentet.

2.3.3 OpenOffice Writer (med kloner)

Klipt og limt fra Wikipedia[8]:

> Apache OpenOffice (AOO), er en kontorprogramvare som inneholder tekstbehandlingsprogram, regneark, presentasjonsprogram, tegneprogram og et databaseprogram. AOO er fri programvare, og distribueres av Apache Software Foundation under Apache-lisensen. Det betyr at både programmet og dets kildekode kan lastes ned gratis, og privatpersoner eller bedrifter kan selv gjøre endringer eller reparere feil.
>
> AOO benytter ODF som sitt primære dokumentformat. Dette er en ISO-standard som blant annet er tatt i bruk av offentlig sektor i Norge og EU for utveksling av digitale dokumenter. I tillegg støttes formatene til Microsoft Office og en rekke andre kontorprogrammer. AOO er multiplattform programvare, tilgjengelig for mange forskjellige operativsystemer, deriblant Microsoft Windows, Mac OS X, Linux, Solaris, FreeBSD, NetBSD og OpenBSD.

[7] `http://no.wikipedia.org/wiki/LaTeX`
[8] `http://no.wikipedia.org/wiki/OpenOffice`

Kapittel 3

Design / Utforming/ Planlegging (Generisk tittel)

Basert på beskrivelsen av oppgaven (Avsnitt 1.3) og analysen i (Kapittel 2), skal denne delen beskrive hvordan man har tenkt å utforme løsningen. Beskrivelsen er noe avhengig av type prosjekt.

Utredning Først og fremst må det redegjøres for hvordan man skal innhente informasjonen som skal danne grunnlaget for utredningen. Deretter må man bestemme hvordan man skal behandle materialet, f.eks. om man skal bruke statistiske metoder. Etter det er det naturlig å presentere en disposisjon for selve utredningen, som i dette tilfellet vil være hovedresultatet i prosjektet. Selve utredningen bør være et frittstående dokument, på samme måte som en film vil være et frittstående dokument i en mediaproduksjon, og et program vil være i et utviklingsprosjekt.

Mediaproduksjon Her vil det være naturlig å presentere story-boards og liknende metoder for å beskrive produksjonen.

Utvikling Her er det bare å ta for seg av ulike måter å beskrive systemer på, alt fra overordnet arkitektur, moduler, meldingsprotokoller etc. Bruk gjerne formelle beskrivelsesspråk, typisk UML.

I det videre tar vi for oss oppgaven med å utvikle struktur, innhold og formgiving av rapportmalen, uavhengig av hvilket verktøy som skal brukes ved implementasjonen.

3.1 Struktur

Vi tar utgangspunkt i den tradisjonelle og generiske strukturen beskrevet i Avsnitt 2.1.1. Det er etter min mening vanskelig å se noen grunn til å gjøre noe annet.

3.2 Innhold

Vi så Avsnitt 2.1.2 at man ved HiØ skiller mellom produkt- og prosessdokumentasjon, og operer med flere frittstående dokumenter. Det er ønskelig at hovedokumentet omhandler løsningen av oppdraget, dvs. selve produktet. Struktur og innhold bør være som som følger:
Front Matter Som i Avsnitt 2.1.1.
Body Hoveddelen av dokumentet (gjennomføringen).

Introduksjon Skal gi leseren et bilde av rammene rundt prosjektet: Kort beskrivelse av oppdraget, litt om oppdragsgiver og prosjektgruppen, prosjektets formål, leveranser og metoder. Den bør også inneholde en oversikt over resten av rapporten.

Analyse Kapittelet tar for seg analysedelen av arbeidet. Den består av to hoveddeler, en grundig beskrivelse av oppgaven basert på skissen gitt av oppdragsgiver, og en undersøkelse av hva som finnes av relatert arbeid, *best practise* og relevant teknologi.

Design Basert på beskrivelsen av oppgaven i introduksjonen og analysen, skal denne delen beskrive hvordan man har tenkt å utforme løsningen. Beskrivelsen er nødvendigvis noe avhengig av type prosjekt. Designet er uavhengig hva slags verktøy man skal bruke i implementasjonen.

Implementasjon Her skal det beskrives hvordan man faktisk produserte leveransene i prosjektet, og viktigst, beskrive selve produktet. Hvilke verktøy brukte man, hvordan foregikk produksjonen, etc. Utformingen av dette kapittelet avhenger helt klart av type prosjekt.

Evaluering De fleste prosjekter avsluttes med en eller annen form for evaluering av resultatene. I utviklingsprosjekter vil det være naturlig med teknisk testing, fungerer programvaren som den skal? Teknisk testing kan utføres av utviklerne selv, eller en ekstern part, f.eks. oppdragsgiver. En oppdragsgiver ønsker ofte å utføre en akseptansetest, dvs. en test som vil avgjøre om de har fått det de har betalt for". Ellers vil det i mange tilfeller være nyttig og viktig med en brukertest, dvs. en strukturert test der sluttbrukerne får komme til orde.

Diskusjon Diskusjonskapittelet er viktig, både for dere selv og sensor. Dette kapitelet er det som i hovedsak skiller et akademisk prosjekt fra et rent næringslivsprosjekt. Ble resultatet som forventet? Ble oppdragsgiver fornøyd? Kunne dere gjort noe anderledes/bedre? Det er her dere skal dokumentere at dere har lært noe underveis, ikke bare levert et produkt til en oppdragsgiver.

Konklusjon Konklusjon er på et vis et sammendrag av diskusjonskapittelet. Her bør dere legge vekt på de viktigste funnene, både når det gjelder produktet og prosessen. Med et fyldig diskusjonskapittel bør trenger ikke konklusjonen bør være mer enn en side.

Back Matter Som i Avsnitt 2.1.1.

3.3 Utformingen

Det følger med en del standard, ferdig definerte stiler med enhver LaTeX distribusjon. Disse er gjennomprøvet over flere 10-år, og har tålt tidens tann både teknisk og estetisk. Når man skal lage sine egne design, vil det som regel alltid være fornuftig å gå ut i fra en av standardstilene. Det gjør vi også i dette prosjektet, og tar utgangspunkt i stilen *report*, parametrisert for dobbeltsidig A4-format, med 11pt basis fontstørrelse. Vi foreslår følgende endringer i report-stilen:

Marger I følge egen og kollegers erfaring er marginene noe store i standardrapporten, slik at det blir litt "trangt" for tekst, tabeller og illustrasjoner. Vi foreslår en layout med reduserte marger på alle sider.

Topptekst / bunntekst Når man slår opp en dobbeltsidig bok, kalles venstre side for *verso*, og høyre for *recto*. Vi ønsker at bunnteksten skal være uten andre elementer enn fotnoter. Sidetallet bør trykkes i topptekstен, til venstre på verso sider, og til høyre på recto sider. På denne måten blir det lett å bla fort igjen for å finne et gitt sidetall, som f.eks. er funnet i en indeks. Videre er det

ønskelig at aktuelt kapittel er angitt til høyre på verso sidene, og aktuelt underkapittel til venstre på recto sidene.

Tittelsiden Det skal være to tittelsider, først en fengende forside, som studentene utformer selv, og deretter en standard bacheloroppgave-side som er lik for alle.

3.4 Leveransene og malen

Første versjon av hoveddokumentet bør bestå av introduksjonen og analyse- og designkapitlene (Kapittel 1, 2 og Kapittel 3), og andre versjon bør være en mer eller mindre komplett beskrivelse av selve resultatet (Kapittel 4). Endelig leveranse tilsvarer den komplette rapporten pluss selve produktet (og poster og presentasjon).

Kapittel 4

Implementasjon / Produksjon / Gjennomføring (Generisk tittel)

Her skal det beskrives hvordan man faktisk produserte resultatene i prosjektet, og viktigst, beskrive selve produktet. Hvilke verktøy brukte man, hvordan foregikk produksjonen, etc. Utformingen av dette kapittelet avhenger helt klart av type prosjekt.

4.1 Utredning

Det er mulig at dette kapitellet er overflødig i et utredningsprosjekt. Utredningen er jo et eget dokument, og trenger vel ikke med kontekst for å kunne evalueres.

4.2 Mediaproduksjon

For denne typen prosjekter kan det være relevant å beskrive og rapportere fra selve produksjonen. Det er vel relativt vanlig at man må endre og improvisere i forhold til opprinnelig plan, og det bør jo absolutt dokumenteres, ikke minst i forhold til diskusjonen (Kapittel 6.2). er mulig at dette kapitellet er overflødig i et utredningsprosjekt. Utredningen er jo et eget dokument, og trenger vel ikke med kontekst for å kunne evalueres.

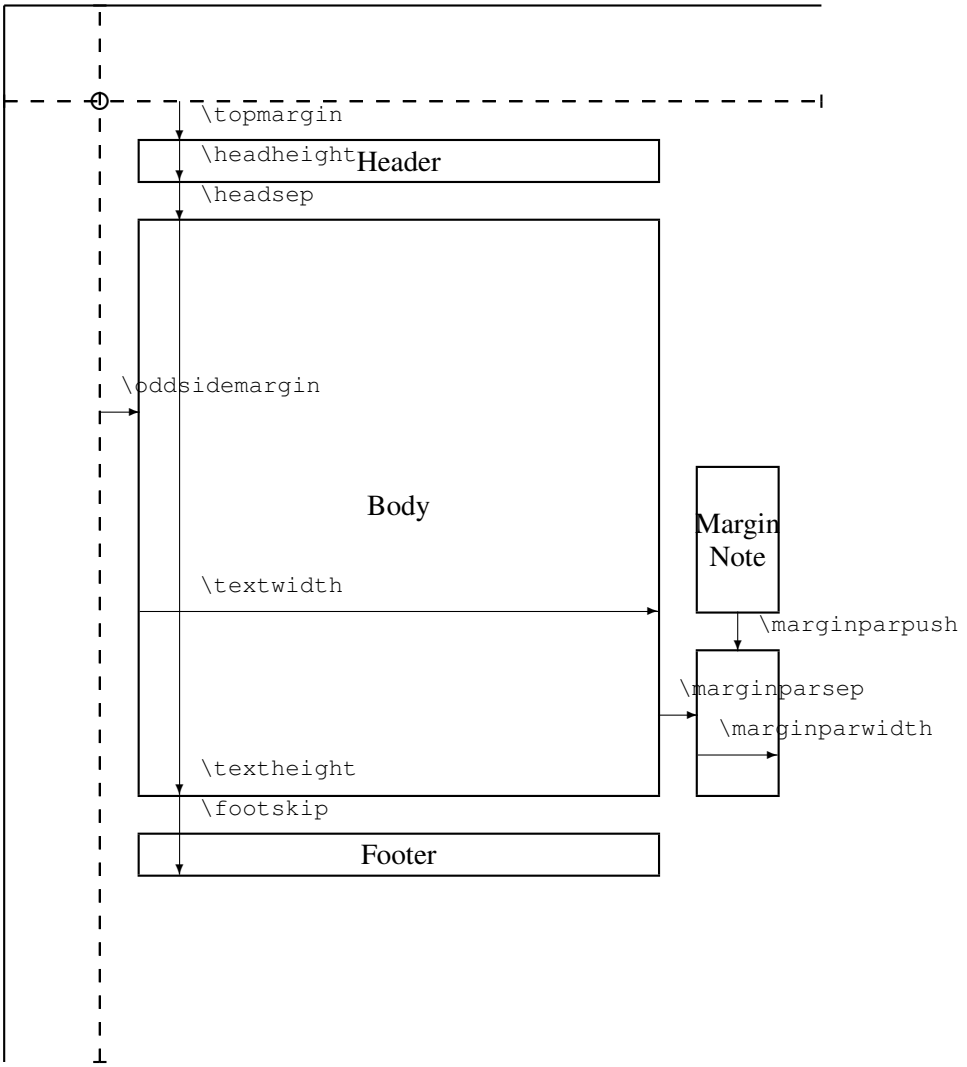
4.3 OpenOffice Writer

Se vedlegg i rapporten *OpenOffice mal for hovedprosjektrapport*.

4.4 LaTeX

For et nærmere innblikk i hvordan denne malen er implementert, se kildekoden som følger med. Resultatet av den ønskede layout kan selvfølgelig sees i dokumentet du leser nå, eller i mer formell form i Figur 4.1.

The circle is at 1 inch from the top and left of the page. Dashed lines represent (`\hoffset + 1 inch`) and (`\voffset + 1 inch`) from the top and left of the page.



Actual page layout values.

`\paperheight` = 296.9965mm	`\paperwidth` = 209.99753mm
`\hoffset` = 0mm	`\voffset` = 0mm
`\evensidemargin` = 9.59988mm	`\oddsidemargin` = -0.4mm
`\topmargin` = -8.4039mm	`\headheight` = 4.21747mm
`\headsep` = 8.78639mm	`\textheight` = 226.99733mm
`\textwidth` = 149.99825mm	`\footskip` = 10.54367mm
`\marginparsep` = 3.51456mm	`\marginparpush` = 1.75728mm
`\columnsep` = 3.51456mm	`\columnseprule` = 0mm
1em = 3.84843mm	1ex = 1.73178mm

Figur 4.1: Layout for recto sider

Kapittel 5

Testing / Evaluering (Generisk Tittel)

De fleste prosjekter avsluttes med en eller annen form for evaluering av resultatene fra prosjektene. I utviklingsprosjekter vil det være naturlig med teknisk testing, fungerer programvaren som den skal? Teknisk testing kan utføres av utviklerne selv, eller en ekstern part, f.eks. oppdragsgiver. En oppdragsgiver ønsker ofte å utføre en akseptansetest, dvs. en test som vil avgjøre om de har fått det de har betalt for". Ellers vil det i mange tilfeller være nyttig og viktig med en brukertest, dvs. en strukturert test der sluttbrukerne får komme til orde.
Evalueringsmetoden vil altså avhenge sterkt av typen av prosjekt. Likevel vil de samme overordnete prinsippene gjelde for alle typer leveranser. En utredning bør kunne passere en akseptansetest, et mediaprosjekt bør kunne underlegges både akseptanse- og brukertest (jfr. kritikerprosessen rundt nye spillefilmer), og et programvareprosjekt bør i tillegg gjennomgå en teknisk test.

Testingen av de to verktøyene er utført på en uformell og ikke særlig vitenskapelig basert måte, og resultatene baserer seg hovedsakelig på forfatterens egne vurderinger[1].

I Tabell 5.2 er det foretatt en summarisk sammenlikning mellom OpenOffice og LaTeX med hensyn på kravene i Kapittel 2.3.1.

[1] I et reelt prosjekt ville denne metoden være litt for tynn ☺

Krav	**OpenOffice**	**LaTeX**
Robusthet	Tildels betapreget. Man har av og til følelsen av å bevege seg på tynn is.	Bunnsolid, Gigabyte type dokumenter ingen problemer.
Åpne formater	Ja. En ODF fil er en zipfil som inneholder en rekke skjemadefinerte XML filer.	Ren, "lettlest" ASCII.
Maler/stiler	Ja, men noe rotete, lett å overstyre lokalt uten at du merker det.	LaTex "oppfant" prinsippet med å skille innhold, struktur og presentasjon. Enormt tilfang av maler og stiler for det meste.
Oppsplitting	Mulig, men ikke trivielt/robust.	Trivielt. Modus Operandi for store dokumenter, f.eks. Proceedings som samler "standalone"-artikler fra mange forfattere.
Kryssreferanser	Går greit, men ikke helt intuitivt.	Konsistent og enkelt.
Bibliografi	Greit med bruk plugins/extensions, f. eks, Zotero.	BibTex innførte prinsippet om å referere til eksterne bibliografisamlinger. En mengde ulike stiler.
Brukermasse	Stadig økende, men med stort innslag av entusiaster og "pionerer".	Stor og kompetent, særlig innen den naturvitenskapelige og tekniske delen av akademia.
Dokumentasjon	Relativt god, men bærer preg av at produktet er "ferskt".	Meget omfattende, men godt spredt i ulike fora.
Versjonering	En odf-fil er en binærfil (zippet og komprimert samling XML-dokumenter), og egner seg dårlig for tradisjonell versjonskontroll. Det finnes en innebygget versjoneringsmekanisme, men det er usikkert hvor hensiktsmessig denne er.	Kan ikke bli bedre? Egen erfaring med komplekse og store dokumenter fra tidlig 90-tall som problemløst kan behandles i dagens verktøy.
Framtidssikker	Uklart, siden det er et "ferskt" verktøy, men potensialet er der.	Filene er ren ASCII, og håndteres derfor trivielt i alle typer versjoneringssystemer.
Staving/grammatikk	Innebygget, mulig å laste inn ulike typer eksterne ordlister etc.	Mange vektøy, fordi filene er ren ASCII.

Tabell 5.2: Summarisk vurdering av sentrale aspekter i OpenOffice og LaTeX

Kapittel 6

Diskusjon

Diskusjonskapittelet er viktig, både for dere selv og sensor. Dette kapitellet er det som i hovedsak skiller et akademisk prosjekt fra et rent næringslivsprosjekt. Det er her dere skal dokumentere at dere har lært noe underveis, ikke bare levert et produkt til en oppdragsgiver. Her går dere tilbake til Avsnitt 1.4. I hvilken grad ble målene oppnådd (Avsnitt 1.4.1)? Leverte dere de forventede resultatene (Avsnitt 1.4.2)? Fungerte metoden dere beskrev i Avsnitt 1.4.3? Hva ble bra? Hva ble ikke fullt så bra? I begge tilfeller, hvorfor? Hva slags problemer støtte dere på? Hva ville dere gjort anderledes, sett i ettertid?

Kapittel 7

Konklusjon

Konklusjon er på et vis et sammendrag av diskusjonskapittelet. Her bør dere legge vekt på de viktigste funnene. Med et fyldig diskusjonskapittel bør trenger ikke konklusjonen bør være mer enn en side, men legg vekt på tydelig og godt språk. Husk at sensor antagelig først leser sammedraget, deretter konklusjonen. Etter det står resten av dokumentet for tur. Fokuser på hvordan produktet ble i forhold til forventningene til oppdragsgiver og dere selv. Gjenta i kortform de viktigste punktene fra diskusjonskapitlet. I tillegg bør dere nå se videre, om hva som burde bli gjort ved en videreføring av prosjektet (framtidig arbeid). I tillegg kan dere her gi råd om videre arbeid.

Prosjektet har demonstrert at det er mulig å lage maler for hovedrapporten i bachelorprosjektet ved HiØ/IT i både OpenOffice ogLaTeX. Malene tilfredstiller kravene til struktur og utforming, og tilbyr nødvendig funksjonalitet. Det er imidlertid åpenbart at at LaTeX er det mest robuste, fleksible, forutsigbare og modne alternativet. OpenOffice egner seg relativt dårlig til kollaborativ skriving, fordi det er vanskelig å splitte dokumentet i biter, og vanskelig å gjennomføre en effektiv versjonskontroll. Hvis man likevel ønsker å bruke OO, anbefales det på det sterkeste at en enkelt person har ansvaret for dokumentet under hele prosjektet (resten av gruppa kan med fordel bidra, men da ved å forsyne "brødtekst" til den dokumentansvarlige).

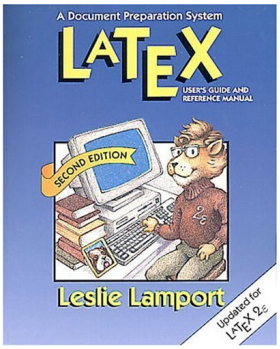
Quisque consectetuer. In suscipit mauris a dolor pellentesque consectetuer. Mauris convallis neque non erat. In lacinia. Pellentesque leo eros, sagittis quis, fermentum quis, tincidunt ut, sapien. Maecenas sem. Curabitur eros odio, interdum eu, feugiat eu, porta ac, nisl. Curabitur nunc. Etiam fermentum convallis velit. Pellentesque laoreet lacus. Quisque sed elit. Nam quis tellus. Aliquam tellus arcu, adipiscing non, tincidunt eleifend, adipiscing quis, augue. Vivamus elementum placerat enim. Suspendisse ut tortor. Integer faucibus adipiscing felis. Aenean consectetuer mattis lectus. Morbi malesuada faucibus dolor. Nam lacus. Etiam arcu libero, malesuada vitae, aliquam vitae, blandit tristique, nisl.

Tillegg A

How to use this template

Here we briefly explain how to use this template. The template is designed to be rather self-explanatory, and all of the features you need are present somewhere in the source code, so you will come a long way by cutting and pasting.

It is assumed that the user has (or provides herself with) basic knowledge of LaTeX. There are numerous good tutorials online[1], but we warmly recommend the original documentation: "LaTeX: A Document Preparation System" (Figure A.1) [**?**]. LaTeX is basically a collection of macros written in TeX. This system, which is a low level tool for digital typesetting, is known for producing scientific documents of unprecedented quality. It was developed by Donald Knuth, one of the giants of computer science.



Figur A.1: The LaTeX "bible", 2. edition

There are plenty of good LaTeX editors, and of course, there are many possibilities for the Emacs users[2]. Personally, I use *texmaker*[3] for OSX, and *Kile*[4] for Ubuntu (and *WinEDT*[5] for MS, before I abandonded that platform for good).

[1] `latex-project.org` is a good starting point
[2] `www.gnu.org/software/emacs`
[3] `www.xm1math.net/texmaker/`
[4] `kile.sourceforge.net`
[5] `www.winedt.com`

A.1 Compilation

Making documents with LaTeX is basically like writing software. This document is produced by compiling a collection of files, all in the same folder. There is a top level file called `main.tex`, which contains all commands that decide the format, layout etc., or in other words, the *style*[6]. It also includes the files containing the actual text (in general one file for each chapter).

To compile the document to produce a pdf file, use your terminal/command window (or use the build function in your editor), go to the document folder, and issue this command twice:

```
pdflatex main
```

This process produces a pdf file, `main.pdf`. When printing the document, remember to select the double page option.

However, as you may have experienced when compiling source code, there might be syntax errors, missing files etc. to be fixed. LaTeX is quite verbose when compiling, and does a lot of complaining (warnings), which you most often can ignore. However, it can be a bit tricky to find the source for an error. You sholud typically search backwards from the end of the compilation output. Listing A.1 is an example from compiling this document, where the error is misspelling of the LaTeX macro (should be `\LaTeX`, not `\LateX`). The key error message is `! Undefined control sequence`, followed by a quotation of the line where the error has occurred, along with the line number. The name of current file is found a couple of lines above: `(./how-to.tex`.

Kode A.1: LaTeX error output

```
...

Overfull \hbox (6.0pt too wide) in paragraph at lines 43--44
[][][][][][]

Underfull \hbox (badness 10000) in paragraph at lines 43--44

) (./conclusion.tex [20]
Chapter 7.
[21]) (./main.bbl [22]) [23] [24]
No file main.ind.
(./how-to.tex
Appendix A.
[25]
! Undefined control sequence.
l.48 misspelling of the  \LateX
                         \  macro (should be \verb|\La...

?
}
```

A.2 Chapters/sections/paragraphs

LaTeX lets you break the document down into chapters, sections, subsections, subsubsections and paragraphs. By default subsubsections and paragraphs are not numbered or included in the table of contents.

[6] Think of HTML and stylesheets ... guess where that idea came from ...

A *chapters* may contain plain text, elements like figures and tables, and *sections*. Plain text is commonly structured by *paragraphs*. Paragraphs are separated by one or more *empty lines*[7]. A section may contain *subsections*, and the next level is *subsubsection*. Finally we have a special type of *paragraph*.

Below follows examples of all these constructs.

A.3 Section

This is a section. Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

A.3.1 Sub section

This is a sub section. Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

[7] Correspondingly, when there are two or more consecutive whitespaces in the text, these will be ignored

Sub sub section

This is a sub sub section. Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

Paragraph This is a titled paragraph. Please note the difference between the standard paragraphs produced by blank lines, and this type, which is the lowest level of elements with titles.

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam congue neque id dolor.

Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

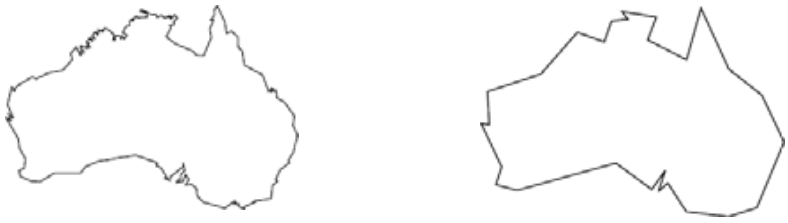
A.4 Figures, tables, equations, etc.

Please see the source code (`how-to.tex`) for details on the implementation of these elements.

In general, figures and tables shall be numbered, and have a caption. Figures and tables *must* be referenced to at least once in the text. Equations and similar elements should also be numbered, but they are not always referred to.

Figures, tables, equations, and similar constructs are so-called *floats*. This means that LaTeX will place them in a position that is "best", taking many aspects into consideration. The result is that the elements may not be positioned exactly where the writer wants (in particular when you have many floats near each other, like in text you are reading now), and this is in my experience very frustrating for the novice user . . . see Section **??**.

Figures are most often produced from files on common graphics formats (like pdf, png, jpg, etc). You can use a single image file, as in Figure A.1, or you can combine several images, see



(a) input (b) output

Figur A.2: Input and result from running the Douglas-Peucker line simplification algorithm [**?**] (from [**?**])

Figure A.2, consisting of Figures A.2(a) and A.2(b).

Mastering tables has a relatively steep learning curve. Still, simple tables, like Table A.1, are relatively easy to make. A more complex example is demonstrated in Table A.2.

X		
	X	
		X

Tabell A.1: Simple table

Combination	**Included Optional Steps**			
	1	**2**	**3**	**4**
1	X			
13			X	X
14				X
15	Nano Particles Deposited, Not Sintered			
16	Only Grinded Wafer 1, No Particles Deposited, Not Sintered			
17	Only Grinded Wafer 2, No Particles Deposited, Not Sintered			

Tabell A.2: Complex table

Within mathematics and natural sciences there is a common belief that LaTeX is unrivaled when it comes to typesetting formulas, equations, and complex specialized notation, as the following examples demonstrate.

You can have inline equations, like this: $\alpha = \beta\gamma\delta$, or you can formulate them as numbered floats, as in Equation A.1.

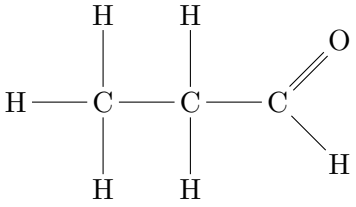
$$\alpha = \beta\gamma\delta \tag{A.1}$$

Equation A.2 is a bit more complicated:

$$I_{zz} = \int_{-b/2}^{b/2} \int_{-h/2}^{h/2} y^2 dydx = \frac{bh^3}{12} \tag{A.2}$$

There are loads of special characters, like $\approx$, $\pm$, $\times$, $\div$, $\propto$, $\leq$, $\geq$, $\ll$, $\gg$, $\neq$, ∇, $\Re$, $\Im$, $\flat$, $\sharp$, ∂, ∞, and $\heartsuit$.

Here is a chemistry related example:



See the next section for a complete example of a mathematical proof.

A.5 Proof of the area of a circle formula $A = \pi r^2$

Theorem 1 *The area of circle with radius r is πr^2.*

Proof: The equation of a circle centered at the origin is

$$x^2 + y^2 = r^2,$$

where r is the radius. We write y in terms of the variable x and the constant r:

$$\frac{x^2}{r^2} + \frac{y^2}{r^2} = 1$$

$$\frac{y}{r} = \sqrt{1 - \frac{x^2}{r^2}}$$

$$y = r\sqrt{1 - \frac{x^2}{r^2}}$$

By symmetry, the area of a circle centered at the origin is four times the area of the circle between $(0, 0)$ and $(r, 0)$ above the x-axis. We can integrate to find the area (A):

$$A = 4r \int_0^r \sqrt{1 - \frac{x^2}{r^2}}\, dx$$

To evaluate the antiderivative of $\sqrt{1 - \frac{x^2}{r^2}}$, we make the substitutions:

$$x = r \sin\theta$$

$$\theta = \arcsin\frac{x}{r}$$

$$dx = r\cos\theta\, d\theta$$

Thus, our integral becomes:

$$A = 4r \int_0^r \sqrt{1 - \frac{x^2}{r^2}}\, dx = 4r \int_0^{\pi/2} r\sqrt{1 - \sin^2\theta}\cos\theta\, d\theta$$

We can use the trigonometric identity $1 - \sin^2\theta = \cos^2\theta$:

$$A = 4r \int_0^{\pi/2} r\sqrt{1 - \sin^2\theta}\cos\theta\, d\theta = 4r^2 \int_0^{\pi/2} \cos^2\theta\, d\theta$$

We then apply $\cos^2\theta = \frac{1}{2}(1 + \cos 2\theta)$:

$$
\begin{aligned}
4r^2 \int_0^{\pi/2} \cos^2\theta \, d\theta &= 4r^2 \int_0^{\pi/2} \frac{1}{2}(1+\cos 2\theta)\, d\theta \\
&= 2r^2\theta\Big|_0^{\pi/2} + 2r^2 \int_0^{\pi/2} \cos 2\theta \, d\theta \\
&= \pi r^2 + 2r^2(\sin 2\theta)\Big|_0^{\pi/2} \\
&= \pi r^2
\end{aligned}
$$

Thus, the area of a circle with radius r is πr^2. ■

A.6 Listings and other *environments*

You can apply specialized layout by using *environments*. Environments are constructed like this:

```
\begin{some-environment}
The text and other contents goes here
\end{some-environment}
```

The most common environments are the following three different list types[8]. First, the bullet list:

- First item
- Second item
- Third item

Then, the enumerated list:

1. First item
2. Second item
3. Third item

And finally the decription list:

First item First description Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Second item Second description

Third item Third description

Needless to say, as anything else in LaTeX, these lists can be customized to your liking.

[8] Here we are using compact versions of the standard lists, which tend to produce to much "air"

A.7 Source code

Large chunks of code should be placed in an appendix, but smaller pieces can be listed in the main part. We here demonstrate two ways of doing this.

In Listing **??** we have included code from a separate file.

Kode A.2: Recursive solution of Towers in Hanoi

```
/*************************************************************************
 *  Compilation:  javac Hanoi.java
 *  Execution:    java Hanoi N
 *
 *  Solves the Towers of Hanoi problem on N discs. The discs are labeled
 *  in increasing order of size from 1 to N and the poles are labeled
 *  A, B, and C.
 *
 *************************************************************************/

public class Hanoi {

    // move n smallest discs from one pole to another, using the temp pole
    public static void hanoi(int n, String from, String temp, String to) {
        if (n == 0) return;
        hanoi(n-1, from, to, temp);
        System.out.println("Move disc " + n + " from " + from + " to " + to);
        hanoi(n-1, temp, from, to);
    }

    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        hanoi(N, "A", "B", "C");
    }
}
```

In Listing **??** we have copied and pasted from the same file:

Kode A.3: Core of the recursive solution of Towers in Hanoi

```
// move n smallest discs from one pole to another, using the temp pole
 public static void hanoi(int n, String from, String temp, String to) {
     if (n == 0) return;
     hanoi(n-1, from, to, temp);
     System.out.println("Move disc " + n + " from " + from + " to " + to);
     hanoi(n-1, temp, from, to);
}
```

A.8 Cross-references and bibliography

As mentioned earlier, all non-text elements should be numbered, and should be referenced to at least once in the text. This is what is called *cross-referencing*, and is easily accomplished. First we need to attach a label to the element: `\label{type:name}`. Then we use this label in the reference: `\ref{type:name}`. The reference is only a number, so we usually add the element

type as a capitalized prefix, for instance like his: `Figure \ref{fig:lamport}}`, which produces: Figure A.1.

When you need a reference to an item in your bibliography (books, articles, web sites etc), you need to use one or more "databasefiles, which are plain texts files with bibliography items formatted according to certain rules. These files have the extension `.bib`, and must be included in the main file. An example of a correctly formatted bibliography item is found in **??**.

Kode A.4: BibTex entry

```
@book{perelman97mht,
author = {Perelman, Leslie and Barrett, Edward},
title = {{The Mayfield Handbook of Technical and Scientific Writing}},
year = {1997},
edition = {1},
publisher = {McGraw-Hill, Inc.},
address = {New York, NY, USA},
}
```

This format is called *bibtex*, and all the academic search engines, including *Google Scholar*, exports to this format. When referencing, you use this command: `\cite{perelman97mht}` and you get: [**?**].

To include a newline added reference, you run the following sequence:

```
pdflatex main
bibtex main
pdflatex main
```

Bibtex generates the final bibliography only from the references in the document (and not from all the items in the `.bib` files).

The different scientific communities have their own guidelines for how to format the entries in the bibliography, and how to format the references in the document. You decide which style to use with the command `\bibliographystyle{somestyle}` in the main file.

There are several tools for creating and maintaining bibliography databases that export bibtex files, both standalone programs, plugins to editors and browsers, and cloud based services[9].

A.9 Index

You may also make an index page. For instance, in this chapter, every time the word LaTeX occurs, I have put the command `\index{LaTeX}` close to the occurrence. The index page is produced by including `\printindex` in the main file, and running the following sequence:

```
pdflatex main
makeindex main
pdflatex main
```

The result is an entry on the index page, something like this: "LaTeX, 27, 28, 30, 31, 35".

[9] For instance, check out `zotero.org`

A.10 Fonts

Fore every LaTeX distribution, there is a default set of fonts. It is possible to customize this setup. Don't[10].

You can do it locally like this (but use it with extreme care):

Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetuer quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.

However, size, weight, style and font family may be manipulated using the following standard commands.

Font size First of all, you decide in the preamble the default font size for your document. The `documentclass` takes the parameters `10pt`, `11pt`, or `12pt`. Locally, you can change the style by the following commands, that resizes the font *relatively* to the default size:

tiny
scriptsize
footnotesize
small
Default: normalsize
large
Large
LARGE
huge
Huge

Font weight (Font series) You can locally change the font weight:

Default: Medium
Bold

Font style (Font shape) You can locally change the font style:

Default: Normal (Upright/Roman)
Italic
Slanted
SMALL CAPS

[10] It's by all means possible, but if you get this urge, you most likely suffer from a stroke of extreme procrastination ... ah ... well, then, check out the very last part of the preamble (that is everything before `begindocument` in the main file).

Font family You can locally change the font family:
Default: Roman (serif)
Sans serif
`Typewriter (monospace)`

A.11 Best practice

- First: Focus on *content* and *structure*
- Later: Decide on layout and style
- Use mostly the default settings
- If you need special functionality, look for packages covering your needs
- If you do not find suitable packages, make your own macros
- Learn by 1) asking fellow students, 2) google and cut'n paste, and 3) by sending me an email or come to my office
- Compile frequently
- Commit frequently to your versioning system[11]
- Run spell checks when things start to get complete[12]
- Last: perform minor fine-tuning (typically to sort out bad placements of floats). Remember that every fix you apply may affect the subsequent layout.

[11] SVN is a good choice, or use any of the many free online services.

[12] Most editors provide built-in spell check functionality (which ignores the mark up commands). On Linux platforms you have the ispell and aspell command line tools which can be configured for LaTeX. There are also stand-alone tools around.