

# Varmesumsestimering for norsk landbruk

Interaktive kart for et kupert  
og langstrakt land



Lene Halling  
Carsten Andersen  
Kjersti Nordskog



**Tittel**

Varmesumestimering for norsk landbruk – Interaktive kart for et kupert og langstrakt land

*Accumulated Degree Days for Norwegian Agriculture – Interactive Maps between Fjords and Mountains*

**Studieretning**

Master of Geoinformation Management – Technology Management (MTM)

**Projektperiode**

September 2015 – januar 2016

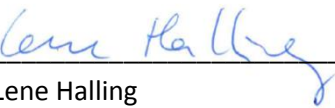
**Projektgruppe**

Gruppenummer 3

**Projektgruppedeltagere**

  
\_\_\_\_\_

Kjersti Nordskog

  
\_\_\_\_\_

Lene Halling

  
\_\_\_\_\_

Carsten Andersen

**Projektveileder**

Lars Brodersen

**Opplagstal: 6**

**Sider: 86**

**Bilag: 7**

Forsidebilde: Kjersti Nordskog



## Forord

Denne rapporten dokumenterer vårt masterprosjekt som utgjør 4. semester i studiet *Master of Technology Management (MTM) – Geoinformation management* ved Aalborg universitet. Rapporten bygger på forarbeidet som ble gjort i henholdsvis andre semester (*Klima på kart – Geokommunikasjon i Landbruksmeteorologisk tjeneste (LMT)*) og tredje semester (*Varmesum og nedbørsmengde på kart – Geokommunikasjon i Landbruksmeteorologisk tjeneste (LMT)*).

MTM-studiet har gått over fire semestre, og om lag halve tiden er dedikert til kursundervisning, mens resten brukes til prosjektarbeid. Alle tre prosjektene er utført av Lene Halling, Carsten Andersen og Kjersti Nordskog, som er studenter ved Aalborg universitets MTM-studium 2014-2016.

Utgangspunktet for at vi gikk i gang med dette arbeidet var at NIBIOs landbruksmeteorologiske tjeneste uttrykte et ønske om å utbedre tjenesten ved hjelp av et økt fokus på kart, og vår oppgave har siden starten tatt sikte på å finne gode og realistiske løsninger for nettopp dette.

Vi har fått betydelig hjelp av Berit Nordskog og Tor-Einar Skog ved NIBIO på Ås, Norge. Berit og Tor-Einar har vært positive og hjelpsomme gjennom alle de tre semestrene dette prosjektet har pågått. Dette er vi meget takknemlige for. Deres konstruktive innspill og ideer har flere ganger hjulpet oss videre når vi har stått fast, eller ikke helt visst hvilken retningen arbeidet helst burde ta for å komme videre. Vi har også hatt en tett dialog med flere landbruksrådgivere gjennom tre semestre, og er svært takknemlige for bidragene vi har fått fra dem.

Takk også til Ole Einar Tveito ved Meteorologisk Institutt, Ingvild Nystuen ved NIBIO og Morten Fuglsang ved Aalborg universitet for tips og innspill. Sist, men ikke minst takk til vår veileder Lars Brodersen ved Aalborg universitet for klare og direkte kommentarer til alt vi har sendt hans vei.

## Resymé

Denne prosjektrapporten beskriver prosessen med å designe og bygge opp en prototype for visning av varmesum på webkart i Landbruksmeteorologisk Tjeneste (LMT), hvor varmesummen er interpolert mellom målestasjoner og korrigert for terreng høyder. Målgruppen for løsningen er landbruksrådgiverne i Norge. Målet er at rådgiverne skal kunne estimere varmesummen på steder hvor den ikke er målt, og at rådgiverne vil bruke kartene i sin formidling av klimastatistikk overfor medlemmene.

Vi har brukt blant annet Brodersens designmodell til å treffe de nødvendige valg vedrørende løsningens utseende, innhold og funksjonalitet. Vi har i tillegg bygd opp et flytdiagram til å skissere løsningens logiske sammenheng, og har dessuten brukt diagrammet som grunnlag for planlegging av utviklingstiltakene, og for samtaler i gruppen om løsningens oppbygning og virkemåte. Vi har undersøkt om datagrunnlaget i LMT er godt nok til å kunne interpolere mellom værstasjonene, og funnet ut at stasjonstettheten i de fleste tilfelle er tilstrekkelig til å kunne gjøre dette. Likevel vil det være tilfeller hvor målt temperatur bare bør høydekorrigeres utover terrenget fordi avstanden til nærmeste stasjon er for stor. Gjennom en analyse av de innlagte stasjonshøydene i LMT har vi funnet enkelte feil og mangler som må rettes før dataene kan brukes i denne løsningen.

Målet var å lage prototypen på basis av LMTs eksisterende arkitektur og ved hjelp av Open Source programvare. Via søk på forskjellige utviklerfora på nettet har vi funnet tekniske komponenter som kan utføre de enkelte prosessene. I samarbeid med NIBIO har vi fått utviklet en webservice som ivaretar uttrekk av data fra LMTs database til bruk for interpolasjonen. Interpolasjon og høydekorrigering skjer ved bruk av algoritmer i GRASS, og vi mener at automatisk avvikling av den samlede prosessen kan skje gjennom bruk av PyWPS. Resultatet er et rasterlag som fargelegges ved hjelp av en monokrom fargeskala hvor forskjellen vises med varierende svertningsgrad. Til fargeleggingen brukes en normalisert verdiskala som tar høyde for den store variasjonen i verdiomfanget i celleverdiene. Resultatet settes opp som en webservice og vises i en kartklient basert på Open Layers, hvor kartet legges inn i eksisterende stasjonssider i LMT.

Prototypen vi har laget fungerer ikke helt automatisk, men via den har vi kunnet demonstrere at kombinasjonen av de valgte eller tilsvarende komponenter kan brukes til å produsere det ønskede resultatet. Test av prototypen hos utvalgte brukere har vist at det finnes potensiale i varmesumskartet, og vi har skissert videre utviklingsmuligheter.

I løpet av prosjektet har vi flere ganger justert mål og ambisjoner for løsningen, og rapporten inneholder ikke løsninger på alle de problemstillingene vi har formulert. Vi har likevel dokumentert arbeidet underveis i prosessen, og NIBIO har derfor et godt grunnlag for et videre arbeide med prosjektet.

Proessen har lært oss at utvikling av en løsning som denne krever innhenting av kompetente utviklingsressurser. Vi har også lært at vi kan bruke vår kunnskap og erfaring med geografisk informasjon, modeller og verktøy til å delta i og bidra til utvikling av tekniske komponenter. Vi kan analysere brukerbehov, innhente ny kunnskap og spesifisere ønsket utvikling av tilpasninger, og teste om produktet har den ønskede effekt på brukernes oppgaveløsning.

## Summary

This report describes the process of designing and building a prototype for visualizing the interpolated and height-adjusted heat sum on a web-based map in Agrometeorology Norway (LMT). Our system is designed for agricultural advisors in Norway, and the goal of the project is to enable the advisors to estimate the heat sum in places where there are no measurements of temperature. It is our hope, that the advisors will use the heat sum maps to convey climate statistics to their members.

We have used Brodersen's design model to make the necessary decisions regarding the interface, content and functionality of the system. We have also constructed a flow chart to visualize the logical processes and dataflow. This chart was used as a basis for planning development activities and for internal discussions of the architecture and functionality of the system. Further, we have examined the data of LMT in order to find out if the density of the weather-stations is sufficient to carry out interpolation of the heat sum values between the stations. We have found that in most parts of Norway there is a sufficient weather-station density, but there are also places where only a height correction should be carried out. Through analysis of the existing station heights in LMT we have found a number of errors and missing data that need to be corrected before all weather-stations can be used in this system.

The goal of the project was to make the prototype based on LMT's existing architecture and by using Open Source software. By searching different developer forums on the internet, we have identified technical components to carry out the individual parts of the process. In cooperation with NIBIO we have developed a webservice to carry out the extraction of data from LMT for use in the interpolation process. Interpolation and height correction will be performed using GRASS-algorithms, and we believe that the whole process can be automatized using PyWPS. The result of this will be a raster layer that must be coloured using a monochrome colour-scale, where colour density is used to differentiate between levels. The colourization will also make use of a normalized value range in order to compensate for a potentially large variation in the cell value ranges of the resulting rasters. The raster map will be presented as a webservice and will be shown in a map client based on Open Layers. In production we intend the map to be presented on the existing webpages in LMT.

The prototype we have made is not fully automated, but via the prototype we have been able to demonstrate that the combination of the chosen (or similar) components can produce the desired result. A test of the system showed that the users see a potential in the heat sum maps, and we have identified further development possibilities.

In the course of the project, we have adjusted goals and ambitions for the system and this report does not contain solutions for all the problems identified. Nevertheless, we have documented our work and our findings during the process, to pass on to NIBIO for use in further development.

We have learned from this process that this kind of development requires skilled developers. We have also learned that we can use our knowledge and experience in geoinformation together with acquired models and tools to participate in and contribute to the development of GIS components. We are able to analyze users' needs, gain new knowledge and specify necessary development and adjustment, as well as to carry out tests to find out whether the developed functions have the desired effect for the users' work process.

# Innholdsfortegnelse

Forord.....	1
1 Innledning.....	6
1.1 Bakgrunn .....	6
1.2 Problemanalyse .....	9
1.3 Hovedhypoteser .....	10
1.4 Plan for løsning.....	10
1.5 Avgrensning.....	11
1.6 Leseveiledning .....	13
2 Teori og metoder.....	14
2.1 Utilfreds – tilfreds.....	14
2.2 Forståelseshorisont og trepartsmodell .....	14
2.3 Samtalemodellen.....	16
2.4 Flytdiagram.....	18
2.5 Designmodellen.....	19
3 Løsningsdesign.....	20
3.1 Problembeskrivelse .....	20
3.2 Løsningsdesign via designmodell .....	24
3.3 Modellering av dataflyt .....	29
3.4 Teknisk arkitektur .....	29
4 Valg og utprøving av tekniske komponenter .....	32
4.1 Datauttrekk fra LMT .....	32
4.2 Interpolasjon .....	41
4.3 Høydekorreksjon .....	45
4.4 Presentasjon på webside.....	50
4.5 Automatisering av prosessen .....	54
4.6 Prototypen.....	59
5 Test av prototype .....	69
5.1 Slik utførte vi testen .....	69
5.2 Overvåket brukertest .....	70
5.3 Tilbakemeldingene .....	71
6 Konklusjon.....	72



6.1	Løsningsdesign.....	72
6.2	Tekniske komponenter.....	72
6.3	Brukertest.....	73
6.4	Hovedkonklusjon.....	73
7	Perspektivering.....	75
7.1	Fremtidsscenarier for LMTs varmesumskart.....	75
7.2	Kostnader og ressurser.....	77
7.3	Samfunnets nytteverdi.....	77
7.4	Eksportpotensiale.....	78
8	Refleksjon og læring.....	79
8.1	Hvordan gjøre ting man ikke kan.....	79
8.2	Hva forventet vi å lære – og hva lærte vi faktisk?.....	80
8.3	Hensiktsmessighet.....	80
8.4	Samarbeidet i gruppen og samarbeidet med andre.....	81
8.5	Rollefordeling.....	81
8.6	Metode.....	81
9	Kildekritikk.....	83
9.1	NIBIO.....	83
9.2	Forholdet til landbruksrådgiverne.....	83
9.3	LMT som datakilde.....	83
9.4	Nettbaserte kilder.....	84
10	Litteratur.....	85
Bilag	.....	87
I.	Ordforklaring.....	87
II.	Designmodellen.....	89
III.	Flytdiagrammet vi har forholdt oss til under arbeidet.....	90
IV.	Flytdiagram som viser hvordan et endelig resultat kan se ut.....	91
V.	Flytdiagram som viser prosessen for den foreliggende prototypen.....	92
VI.	Prototype brukertest for Brunlanes.....	93
VII.	Prototype brukertest for Landvik.....	94

# 1 Innledning

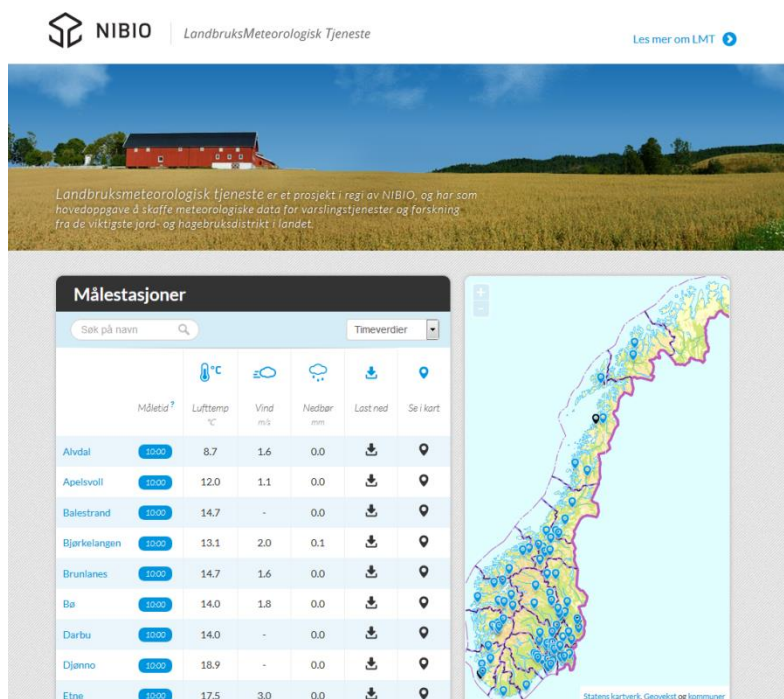
I dette kapittelet vil vi presentere det som er bakgrunnen for prosjektet gjennom totalt tre semestre, og hva vi har funnet i prosjektene på andre og tredje semester. Denne kunnskapen er bakgrunn for alt arbeidet som gjøres i fjerde og siste semester. Videre presenterer vi problemanalysen og hovedhypotesen som ligger til grunn for dette avsluttende masterprosjektet. Til sist i kapittelet kommer en plan for løsning og henvisning til hvordan resten av rapporten er bygget opp.

## 1.1 Bakgrunn

NIBIO<sup>1</sup> (Norsk institutt for bioøkonomi) er et norsk forskningsinstitutt som utvikler kunnskap om blant annet mat- og planteproduksjon, miljø, arealbruk, foretaks og samfunnsøkonomi i Norge.

Instituttets målsetning er å «bidra til matsikkerhet, bærekraftig ressursforvaltning, innovasjon og verdiskaping innenfor verdikjedene for mat, skog og andre biobaserte næringer»<sup>2</sup>.

En av NIBIOs tjenester heter Landbruksmeteorologisk tjeneste (LMT). Dette er en webbasert tjeneste som har som målsetning å forsyne norsk landbruk med meteorologiske data om blant annet lufttemperatur, jordtemperatur, nedbør, bladfuktighet og globalstråling gjennom et nettverk av omlag 80 meteorologiske målestasjoner som er plassert i flere av de viktigste landbruksområdene i Norge. LMT brukes i hovedsak av aktører knyttet til jord- og landbruk i Norge, men leverer også data til nasjonale etater som Jernbaneverket og Meteorologisk institutt og andre forskningsinstitusjoner. Samtidig brukes dataene fra LMT i flere av NIBIOs egenutviklede prognoser, for eksempel grovfôrmodellen, nitrogenprognoser og vanningsinformasjon gjennom tjenesten VIPS (Varsling Innen PlanteSkadegjørere).



Figur 1: Skjermdump fra [lmt.nibio.no](http://lmt.nibio.no) 12. september 2015

<sup>1</sup> I våre foregående rapporter ble NIBIO omtalt som Bioforsk. Dette skyldes at forskningsinstituttene Bioforsk, NILF og Skog og landskap fusjonerte 1. juli 2015 og skiftet navn til NIBIO.

<sup>2</sup> Om NIBIO, URL: <http://www.nibio.no/infoside/om-nibio> Konsultert 12. september 2015

LMT leverer data på time og døgnbasis. Noen av stasjonene er i drift hele året, mens andre kun leverer data i vekstsesongen<sup>3</sup>. Hovedsakelig er målsetningen at dataene som målestasjonene produserer skal brukes som beslutningsstøtte i forbindelse med jord- og landbruk. Per i dag presenteres dataene i form av tabeller og grafer på [www.lmt.nibio.no](http://www.lmt.nibio.no) på undersiden til hver enkelt målestasjon. Det eneste kartet på sidene per i dag er et oversiktskart som viser plassering og fordeling av stasjonene på landsbasis.

### **Bakgrunnen for prosjektet**

NIBIO hadde et ønske, og teori om, at dataene som høstes av LMT kunne få tilleggsverdi ved å presenteres i kartformat. På dette tidspunktet satt ikke NIBIO selv på kompetanse til å utføre arbeidet, og derfor oppsto idéen om å gjøre dette til en masteroppgave i MTM i samtale mellom ansvarlig forsker for LMT, Berit Nordskog og MTM-student Kjersti.

Kjersti tok idéen med til Aalborg universitet, og en studentgruppe bestående av fire personer ble dannet for å jobbe med idéen. Etter kun kort tid ble gruppen redusert til tre personer som har arbeidet med oppgaven gjennom totalt tre semestre. Under presenteres prosessen som har blitt gjennomgått frem til dette fjerde og avsluttende semester.

### **Kunnskap fra andre semesteroppgave**

Rapporten «Klima på kart - Geokommunikasjon i landbruksmeteorologisk tjeneste (LMT) presenterte resultatet av det innledende arbeidet – problemidentifikasjon. Det lå lite føringer for arbeidet fra NIBIOs side – vi sto selv fritt til å definere hva vår oppgave skulle være.

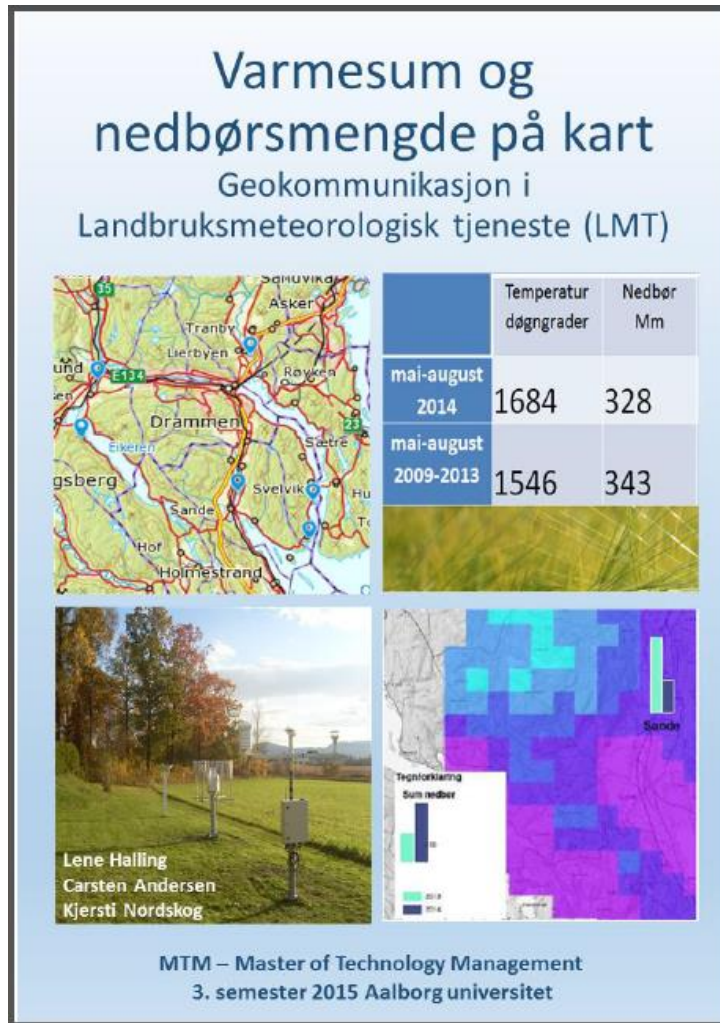
Vi brukte litt tid å bli kjent med selve tjenesten, og utførte i løpet av semesteret også intervjuer med flere typer brukere av LMT (forskere, bønder, landbruksrådgivere) og en webanalyse basert på data hentet fra Google Analytics om hvordan siden ble brukt.

Konklusjonen på intervjuene og webanalysen sammenfalt i stor grad. De fleste brukerne var interesserte i data fra kun én, eller et fåtall, målestasjoner. Kartet på forsiden av LMT (som vist i Figur 1) ble i liten grad brukt, og mange var ikke klar over at kartet kunne zoomes i, eller at det gikk an å klikke på stasjonspunktene for å komme til dataene. Brukerne oppgav likevel at de ikke var fremmede for å bruke andre karttjenester på nett. Landsdekkende sammenstillinger av værdata ble i hovedsak oppfattet som uinteressant – LMTs brukere var kun lokalt og regionalt interesserte. Brukerne var i stor grad fornøyd med måten dataene ble presentert på i dag, og hadde vanskelig for å forestille seg hvordan informasjon på kart kunne gjøre dem bedre i stand til å løse sine oppgaver og gi tjenesten en merverdi. Likevel uttrykte flere av dem interesse for datahistorikk, normaler og avvik og mente disse kunne være interessante og egnet for kartpresentasjon. Resultatet av prosjektet var at vi ikke kom frem til et reelt problem brukerne har med LMT som vi kunne løse ved hjelp av kart eller geovisualisering (Halling, et al., 2015a).

---

<sup>3</sup> OM LMT, URL: <http://lmt.nibio.no/about> Konsultert 12. september 2015

## Kunnskap fra tredje semesteroppgave



Figur 2: Prosjektrapportens forside

varmesum også kan estimeres for områder hvor temperatur ikke har blitt målt. I tillegg ville vi legge inn søyler på stasjonsnivå, som sammenlignet nedbør så langt inneværende år, med tidligere år. Også nedbør og fuktighet er viktig for estimering av for eksempel sprøytetidspunkt. Vi utarbeidet *mock-uper* som viste hvordan en slik løsning kunne bli seende ut, og presenterte disse for landbruksrådgiverne vi hadde opprettet kontakt med. Se eksempel på en slik *mock-up* i Figur 3. Tilbakemeldingene fra rådgiverne var bra, de likte i idéen og trodde de ville ha nytte av en slik løsning i sitt daglige virke (Halling, et al., 2015b). Dette er bakgrunnen vi tar med oss inn i dette fjerde og siste semesteret av MTM-studiet og som ligger til grunn for vårt avsluttende masterprosjekt.

På tredje semester jobbet vi videre med funnene fra andre semester. Ettersom en tydelig problemstilling ikke hadde utkrystallisert seg på annet semester, fortsatte vi å bearbeide funnene vi hadde fra brukerundersøkelsen. Vi kom frem til at det trolig var mest hensiktsmessig å lage en karttjeneste tilpasset landbruksrådgiverne, da disse utgjør en stor andel av LMTs brukere og har relativt ensartede oppgaver å løse ved hjelp av tjenesten. Idéen om å visualisere varmesum kom på basis av at dette er en datatype som LMT per i dag er alene om å tilby på dette nivået, det er av stor viktighet og betydning for mange landbruksrådgivere. Varmesum brukes for å estimere både høstetidspunkt for ulike typer avling, og for å beregne tidspunkt for gjennomføring av tiltak mot enkelte skadegjørere, spesielt skadedyr. Vi ville bruke vår GIS-kompetanse ved å interpolere varmesum mellom målestasjonene og korrigere for naturgitte forhold, slik at

### Beregn varmesum og nedbørsmengde <http://www.vips-landbruk.no/weather/wa707s.iso>

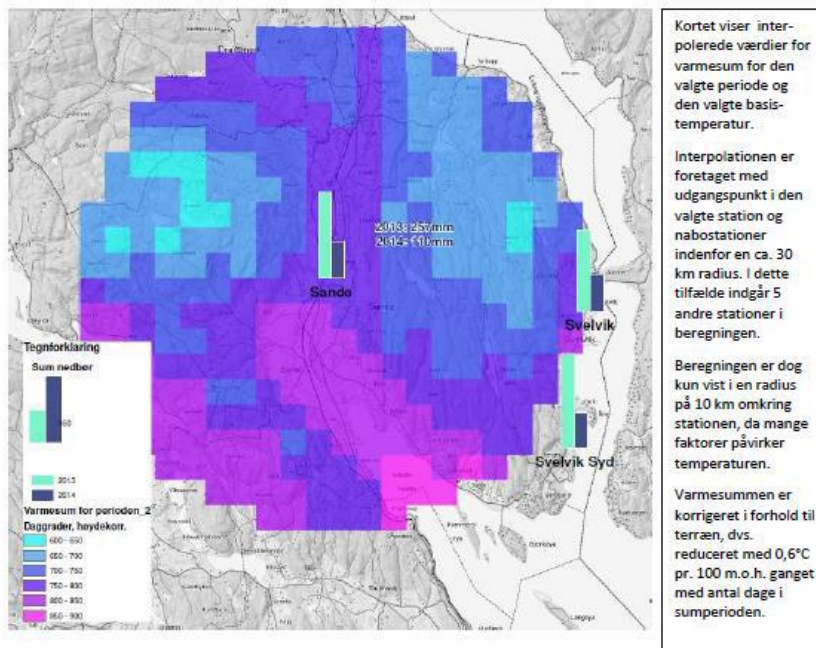
Denne funksjonen beregner varmesum (graddager) og nedbørsmengde mellom to datoer for en valgt klimastasjon. Varmesum er summen av differansen mellom registrert dønnmiddelmiddeltemperatur og en valgt basistemperatur. Nedbørsmengden er summen av registrert nedbør i millimeter (mm) for den valgte periode.

**Beregn varmesum og nedbørsmengde**

Klimastasjon	Sande	Fra og med dato (dd-mm-åååå)	01-05-2014
Basistemperatur	0.0 °C	Til og med dato (dd-mm-åååå)	30-06-2014

Sammenlign med nedbør fra:

Samme periode i 2013       Vis kun beregninger  
 Samme periode i de seneste 5 år      < 300 m høyde  
 Samme periode i et normalår



#### Resultater

**Varmesum:** Varmesum, med basistemperatur 0,0, beregnet ut fra middeltemperatur i 2m høyde for Sande klimastasjon for perioden fra og med 01.05.2014 til og med 30.06.2014 ( 61 dager ) er : **825 graddager**

**Nedbørsmengde:** Total nedbør beregnet for Sande klimastasjon for perioden fra og med 01.05.2014 til og med 30.06.2014 ( 61 dager ) er : **110 mm**

Figur 3: Mock-up av løsningsforslag i tredje semesters rapport

## 1.2 Problemanalyse

For å nå frem til løsningsforslaget, som er utgangspunktet for prosjektarbeidet på fjerde semester skal det overordnet sett skje følgende prosesser:

- Geografisk interpolasjon ut i fra målepunktenes plassering.
- Høydekorleksjon av de interpolerte verdiene ved hjelp av en digital terrengmodell (DTM).
- Visualisering av resultatene ved hjelp av en intuitiv fargeskala som brukeren lett kan lese konkrete tallverdier/verdiintervallet ut av.
- Usikkerheten i beregningene visualiseres ved bruk av celledørrelse, samt begrensning i utstrekning i en radius omkring stasjonene.
- Visning av nedbørssum som søylediagram på stasjonens posisjon, og sammenlikning av denne med tidligere år/normalår.

Vi har likevel identifisert noen problemstillinger som vi må finne ut av før den endelige løsningen kan designes. Disse gjennomgås i kapittel 3.1.

Derfor jobbet vi videre med design av løsningen, slik at den dekker behovene brukerne har med hensyn til form, innhold og interaksjon.

Det kommer også noen mer tekniske utfordringer ved oppsett av en løsning som skal kjøre hele prosessen ut fra brukerens oppgitte input-parametre. Helt overordnet består oppgaven i å lage en webbasert GIS-løsning som kan integreres i noen av LMTs eksisterende nettsider, og som kan generere geovisualiseringer av de valgte parametrene (blant annet varmesum) ut i fra brukerens input.

Gjennom utdannelsen har vi stiftet bekjentskap med forskjellige teknologier til oppsett av webgis-løsninger, og basert på vår generelle kjennskap til GIS-området vet vi at det finnes mange softwaremuligheter vi kan bruke. Området utvikles stadig, og det som var den beste løsningen i går er kanskje ikke det i dag. Dette er grunnen til at det kan være vanskelig å finne oppdaterte og gjennomarbeidede kilder (for eksempel bøker) om web-GIS. Ofte må man konsultere brukerfora på internett for å finne løsninger på problemer som har oppstått.

Målet for dette prosjektet er å lage en fungerende prototype som omfatter geovisualisering basert på værdata fra LMT. For å klare dette må vi arbeide frem mot en løsning hvor flest mulig av problemstillingene er behandlet innenfor rammene til tilgjengelige/valgte softwarekomponenter, og ut i fra de tilgjengelige ressurser.

Et utviklingsprosjekt vil ofte ha begrensede ressurser til rådighet, og med vår nåværende kunnskap er det umulig å estimere omfanget av de enkelte delprosesser. Vi må være i stand til å justere målet og ambisjonene underveis i prosessen. For å sikre at løsningen har verdi for brukerne er det viktig at måljusteringen/prioriteten foregår i dialog med brukere og oppdragsgiver.

### 1.3 Hovedhypoteser

Hvis vi arbeider med å løse de problemer og utfordringer vi har identifisert i designet av geovisualiseringen i LMT, vil vi kunne skissere hvilke prosesser som skal skje, og i hvilken rekkefølge.

Hvis vi deretter undersøker og tester funksjonene i flere forskjellige softwarekomponenter som kan gjøre det vi ønsker, vil vi kunne lage en skisse til teknisk oppsett av en prototype. Prototypen vil vi kunne bruke til å demonstrere og teste det samlede produktet overfor brukerne. Tilbakemeldingene vi får kan innarbeides i løsningen, eller overleveres som anbefaling, for det endelige produktet.

Hvis vi dokumenterer løpende underveis i prosessen og evaluerer det arbeidet vi utfører, samt prioriterer og justerer målet for prosjektet i dialog med oppdragsgiver og brukere, vil vi nå frem til et resultat som har verdi for LMT og dets brukere i deres oppgaveløsning.

### 1.4 Plan for løsning

Prosjektarbeidet på fjerde semester tar for seg siste delen av samtalemодellen (se Figur 6), og består av å utvikle et løsningsforslag til prototype, teste denne, konkludere og perspektivere det samlede prosjektforløpet.

Et vellykket utviklingsløp forutsetter god dialog mellom bestiller, produsent og bruker (Brodersen, 2014) se også trepartsmodellen i Figur 4. For å sikre en god start på utviklingsprosessen hadde vi et

innledende møte med NIBIO som er bestiller i denne sammenhengen. Her ble det avklart hvilke ressurser vi kunne trekke på, og hvilken dialog det skulle være gjennom arbeidsprosessen, samt at vi avstemte mål og ambisjoner for prosjektet.

Vi vil bruke designmodellen til å sørge for at alle aspekter ved løsningens design og bruk blir vurdert. Designmodellen er utarbeidet av Lars Brodersen (2008), og er en metode for informasjons- og løsningsdesign. Da kan vi ta en beslutning om de designmessige utfordringene og sikre at løsningen er så hensiktsmessig som mulig med tanke på fremtidig bruk.

Når vi har tatt beslutningene om løsningens design og virkemåte, har vi et grunnlag for å identifisere de tekniske prosessene som skal inngå. Etterpå kan vi gå i gang med å finne ut av hvilke data og tekniske komponenter som skal inngå i de forskjellige delprosessene. Trolig vil det likevel bli nødvendig med løpende justeringer av designet etter hvert som muligheter og begrensninger i de valgte komponentene blir klarlagt.

Ingen av oss i gruppen har noen større erfaring med GIS-utvikling, men under utdannelsen og i jobbsammenheng har vi stiftet bekjentskap med flere verktøy til geografisk analyse, geoprosessering, databaser og liknende. Ved hjelp av denne kunnskapen kan vi undersøke andre, tilsvarende løsninger og innhente erfaringer fra våre nettverk. Vi regner med å finne en fungerende kombinasjon av komponenter som kan utføre den samlede prosessen. Her vil vi gjøre bruk av den vitenskapelige metode, altså observere og innhente informasjon, stille opp hypoteser som vi tester (for eksempel gjennom å teste en mulig komponent) og til slutt konkludere på hvorvidt komponenten kan brukes eller ikke. Dersom den ikke kan brukes, vil vi prøve en annen, og sette opp en ny hypotese.

Dokumentering underveis i prosessen vil trolig også gi et godt grunnlag for dokumentasjon av løsningen som vi overdrar til NIBIO etter at MTM-studiet er over.

Når vi har kommet frem til en fungerende prototype vil vi teste denne på utvalgte brukere. Disse kan fortelle oss hvordan det oppleves å bruke prototypen, og om de får de svarene som løsningen er tenkt å skulle levere.

## 1.5 Avgrensning

Som et resultat av prosjektet på tredje semester hadde vi identifisert noen problemstillinger, og fått skissert en rekke muligheter for en Web-GIS løsning som viser varmesum på LMT-sidene. For å kunne organisere og konkretisere problemet var vi tidlig nødt til å ta noen avgjørelser om hva vi ønsket å fokusere på og hvordan vi skulle legge opp arbeidet, slik at prosjektet lot seg gjennomføre rent praktisk både hva angår tidsbruk og gruppedeltakernes interesser, evner og kunnskap.

De avgjørelsene det her er snakk om har vært bestemmende for hvordan arbeidet har blitt organisert, hva vi har valgt å gjøre - og kanskje særlig hva vi har valgt å IKKE gjøre. Vi har lagt den vitenskapelige metode til grunn for avgjørelsene våre i de tidligere prosjektene og til dels også i denne siste delen av prosjektet, men mange av avgjørelsene som nå skal beskrives, ble ikke tatt på grunnlag av oppsatt hypotese, test og konklusjon. I stedet har disse avgjørelsene blitt tatt av gruppen i fellesskap etter samtaler seg i mellom, og samtaler med andre ressurspersoner (for eksempel fra NIBIO). Grunnlaget for disse avgjørelsene har vært hva vi gruppedeltakere selv har hatt lyst til å lære mer om, eller ønsket å fokusere på. Vi har også sett an hva vi har hatt tilgjengelige av ekstern hjelp og eksterne ressurser.

### 1.5.1 Fokusområde

Gjennom egne idéer, samtaler med veiledere, NIBIO og med LMTs brukere hadde vi fått mange gode innspill til hva som kunne være fint, praktisk og nyttig i en varmesum-på-kart-løsning. Det er dessuten mange potensielle oppgaver i en slik løsning. Vi kunne for eksempel valgt å fokusere oppgaven på å finne en god interpoleringsmetode – den som gav det aller beste og mest nøyaktige resultatet. Vi har tidligere i MTM-studiet hatt fag som vektlegger hvordan et budskap kommuniseres i en geografisk setting. Dette er også noe vi kunne ha lagt mer vekt på i vår løsning.

Etter å ha diskutert i gruppen kom vi frem til at vi ville legge fokus på en løsning av oppgaven hvor vi mente at våre kunnskaper om GIS både kunne komme godt til nytte, og hvor vi fikk utfordret oss selv, våre kunnskaper og den forståelsen av faget vi allerede har. Vi valgte derfor å fokusere på å få til å sette opp løsningen fra start til slutt, at dataene skal flyte igjennom de ulike komponentene og at det skal foreligge et kart som viser varmesum for brukeren som søker å løse sin oppgave basert på oppgitte parametre. Forbedringer av de ulike komponentene som inngår i løsningen, enten den interpoleringstekniske biten eller den kartografiske fremstillingen, kan utbedres på et senere tidspunkt dersom det er ønskelig.

### 1.5.2 Valg av programvare til utførelse av prototypen

Det virker mest hensiktsmessig at vi i første omgang velger å løse oppgavene innfor de teknologier vi kjenner fra tidligere. Gjennom vårt arbeid, og tidligere gjennomførte fag på MTM-studiet, er vi kjent med et knippe Open Source-løsninger (PostGIS, Geoserver, Open Layers) som vi mener kan inngå i en slik løsning som den vi skal utvikle. Å bruke teknologi vi allerede kjenner til og kan håndtere, sparer oss mye tid, ettersom vi slipper å orientere oss om hvilke andre alternativer som eksisterer, lære oss hvordan disse virker, for så å konkludere med at de kanskje ikke er noe bedre alternativer i løsningen vår likevel. Vi velger i første omgang å bruke teknologi vi er kjent med, og som vi tror kan løse oppgavene vi skal løse. Dersom dette senere viser seg å ikke stemme likevel, vil vi stedet oppsøke andre teknologier.

### 1.5.3 Visualisering av nedbørssum

I de geovisualiseringer som var resultatet av arbeidet på tredje semester viste vi varmesum for en valgt periode, korrigert for høyde, i form av et rasterlag. I tillegg til dette viste vi nedbørssum for samme periode som søylediagram, hvor også nedbørssummen for samme periode året før var vist. Dette fikk oss til å drøfte om brukernes viktigste behov var å finne absolutte tall for varmesum og nedbør, eller om behovet heller er å sammenligne for eksempel inneværende års verdier med tidligere år.

Vi kom frem til at det er behov for begge typer beregninger til forskjellige formål, men at den hybridløsningen vi hadde produsert ikke var hensiktsmessig, ettersom det formodentlig ville forvirre brukeren at man for den ene parameteren (nedbør) kan sammenligne med tidligere år, men ikke for den andre parameteren (varmesum). Gjennom drøftelser i gruppen kom vi også frem til at vi skulle lage den enklest mulige løsningen, som vil være å beregne absolutte tall for en gitt periode.

Ettersom visualiseringen av nedbøren bare ville utgjøre én søyle i søylediagrammet, forsvant dermed også grunnlaget for å bruke søylediagram til visualisering i det hele tatt. Det vil antagelig fortsatt være behov og interesse for å kunne sammenligne varmesum og nedbør med tidligere år, og det er antagelig ikke særlig vanskelig å tilføye når grunnoppsettet er på plass. Vi tar altså ikke dette med i vårt prosjekt.



## 1.6 Leseveiledning

I kapittel 2 gjennomgår vi teori og metoder, inkludert modeller, som er brukt i prosjektet.

Kapittel 3 beskriver designmessige problemstillinger, refleksjoner og beslutninger. Dette munner ut i konseptuelle skisser av den logiske og tekniske løsningsarkitekturen.

I kapittel 4 gjennomgår vi de tekniske delprosessene i detalj, og beskriver hvilke tekniske komponenter vi har testet for å løse konkrete deloppgaver i prosjektet. Her beskriver vi også hvilke anbefalinger vi har nådd frem til, og om det er noe spesielt man skal være oppmerksom på i et eventuelt videre arbeid. Kapitlet avsluttes med en beskrivelse av prototypen vi har konstruert, og hvordan denne er forskjellig fra en ideell/endelig løsning.

Testen av prototypen er beskrevet i kapittel 5. Her beskriver vi hvilken testmetode som er brukt, hvordan testbrukerne er valgt ut, og hvilke resultater testen har gitt.

Konklusjonene fra prosjektet er samlet i kapittel 6. Vi konkluderer på hovedhypotesene, men kapitlet inneholder også konklusjonene fra delkapitlene 3, 4 og 5.

I rapportens kapittel 7 beskriver vi prosjektets perspektiver, blant annet relasjon til omverden og utviklingspotensiale.

I kapittel 8 reflekterer vi over den læring vi har oppnådd gjennom prosjektet og hva vi kan bruke den til.

Rapporten avsluttes med et kritisk blikk på kildene vi har brukt i prosjektet (kapittel 9), samt oversikt over referanser og litteratur. Til slutt kommer rapportens bilag.

Rapporten er bygget opp slik at den beskriver progresjonen som har vært gjennom prosjektets forløp. Den reflekterer hvordan vi har tilegnet oss kunnskap underveis gjennom kildestudier, testing av software og hvordan vi har brukt denne kunnskapen til å ta avgjørelser underveis. På denne måten har vi beveget oss fra et abstrakt konsept, til en praktisk teknisk løsning. Forløpet er ikke nødvendigvis skildret helt kronologisk, da det har vært et gjensidig samspill mellom design av løsningskonseptet og testing av ulike tekniske muligheter. Vi har prøvd å samle disse beskrivelsene etter overordnet emne eller problemstilling.

Første bilag er en ordforklaring over en del fagterminologi som brukes i rapporten.

## 2 Teori og metoder

I dette kapittelet vil vi redegjøre for noen av de teoriene, og i forlengelsen av dem, de metoder, som ligger til grunn for arbeidet med prosjektet gjennom hele semesteret.

### 2.1 Utilfreds – tilfreds

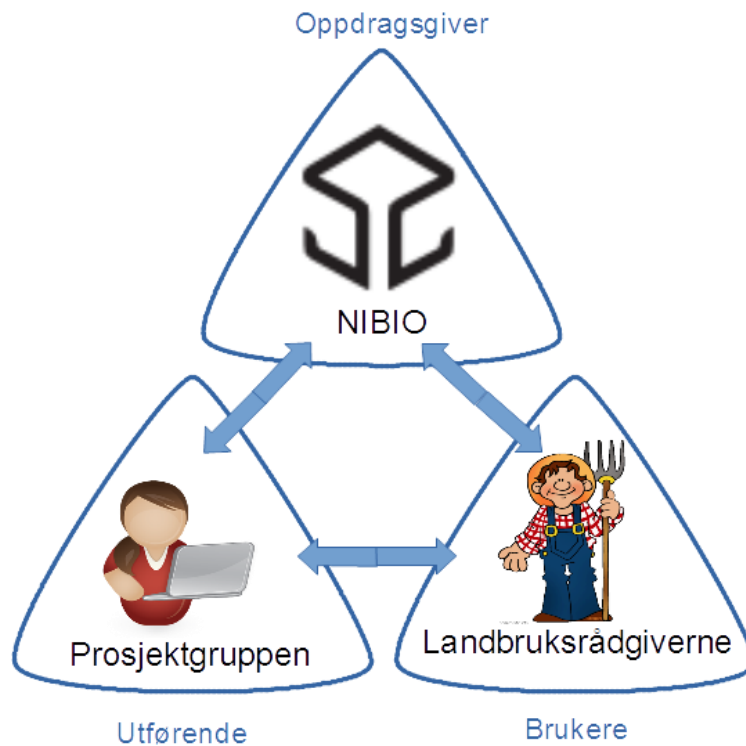
Vi gikk inn i 4. semester med en utilfreds tilstand som har vært startskuddet for alt etterfølgende arbeid. I følge Brodersen (2014) er det erkjennelsen om at man har en utilfreds tilstand som er den utløsende faktor for ethvert prosjekt. I vårt tilfelle begynner prosjektet på dette semesteret med å erkjenne denne utilfredse tilstanden: *"LMTs brukere kan ikke finne varmesummen på andre steder enn der den er målt av en LMT-målestasjon"*. Hvordan vi ser for oss at vi kan komme fra en utilfreds tilstand til en tilfreds tilstand, mener Brodersen er den forandringsprosessen som setter selve prosjektet i gang. En tilfreds tilstand vil i vårt tilfelle være at *"LMTs brukere kan finne varmesummen på steder som også ligger (relativt) langt borte fra en målestasjon"*.

Fordelene med å definere en tilfreds og utilfreds tilstand som utgangspunkt for prosjektet, om det enn kan virke banalt, er at man oppnår en idébevissthet og at det spisser fokuset i prosjektet. Samtidig får man "avslørt" om ikke alle deltakerne i prosjektet er enige om hva som er dagens situasjon/ hva som er det reelle problemet, og hva det er man ønsker å oppnå og ha som fokus i prosjektet. En slik felles oppfatning av hvor man befinner seg, og hvor man skal, kaller Brodersen forståelseshorisont.

### 2.2 Forståelseshorisont og trepartsmodell

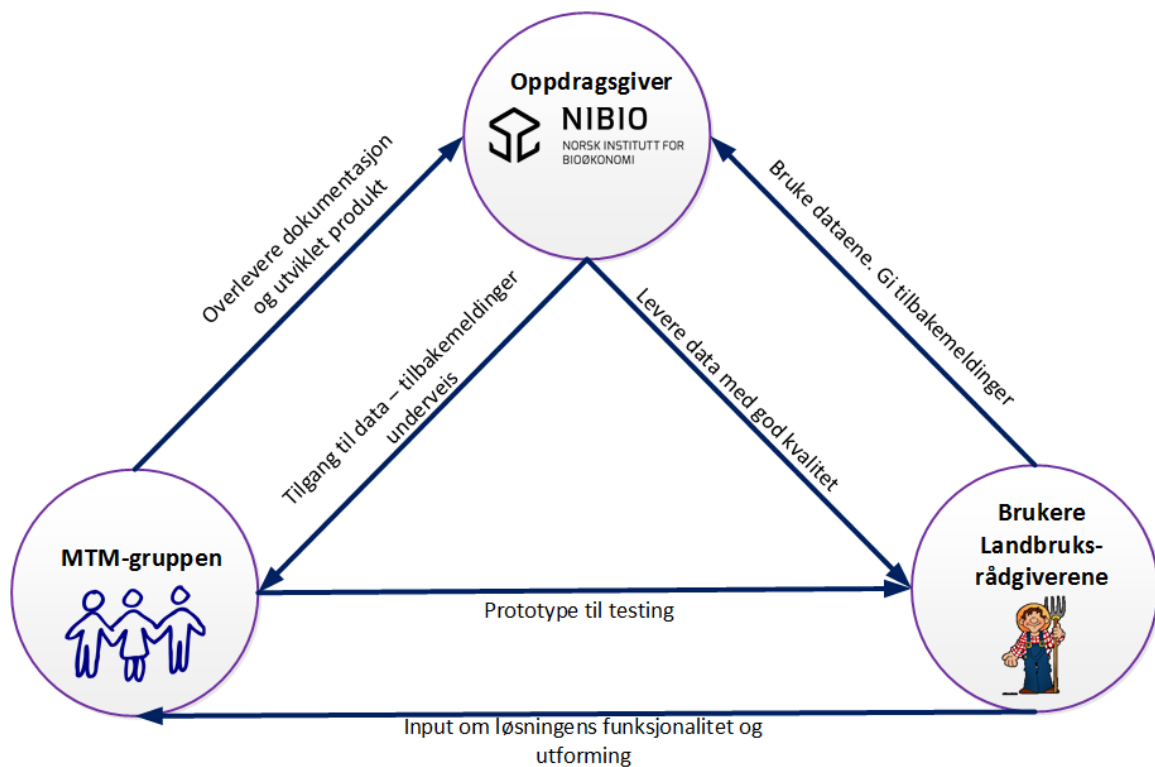
Forståelseshorisonten er summen av alt man vet og kan, de evnene man har og de erfaringene man har gjort seg tidligere. I tillegg spiller følelser, psykologiske prosesser og liknende inn og påvirker hvert individs forståelseshorisont (Brodersen, 2014). I dette prosjektet er det flere forskjellige forståelseshorisonter som skal avklares og samkjøres. At dette skjer er av avgjørende betydning for at prosjektet skal lykkes. Dersom prosjektets parter ikke er enige om hvilket problem det er som skal løses, eller hva som skal til for at problemet ansees løst, sier det seg selv at det både er vanskelig å styre prosjektet underveis, og at det skal mye til at alle partene blir fornøyde.

I alle prosjekter er det tre parter man skal forholde seg til (se Figur 4). Det er som regel alltid en oppgavestiller (som er den som betaler) en produsent (som produserer og utfører selve arbeidet) og en bruker (Brodersen, 2014).



Figur 4: Trepartsmodellen i vårt tilfelle inkluderer prosjektgruppen, landbruksrådgivere og NIBIO.

For å vise hva den konkrete kontakten mellom de ulike aktørene i trepartsmodellen består av, har vi spesifisert hvordan denne arter seg i vårt prosjekt:



Figur 5: Dette relasjonsdiagrammet viser hva kontakten mellom aktørene i trepartsmodellen består av

Her er det flere forståelseshorisonter som skal avklares og passe sammen. Prosjektgruppen, som skal produsere tjenesten, må avstemme sin forståelseshorizont med brukernes. I vårt tilfelle har vi hatt relativt mye kommunikasjon med brukerne av LMT i prosjektene på andre og tredje semester. Det var gjennom å bli kjent med brukerne, og deres forståelseshorizont, at vi sammen kom frem til at et kart som viser varmesum også på steder hvor denne ikke er målt kunne være en god idé. Basert på vår kjennskap til brukerens forståelseshorizont gikk vi i gang med å designe en løsning. Når denne foreligger som en funksjonell prototype kan vi la brukerne teste den, og igjen få korrigert vår egen forståelseshorizont med brukerens. Ved å gjøre dette er det større sjanse for at vi lager et produkt brukerne faktisk vil ønske å bruke.

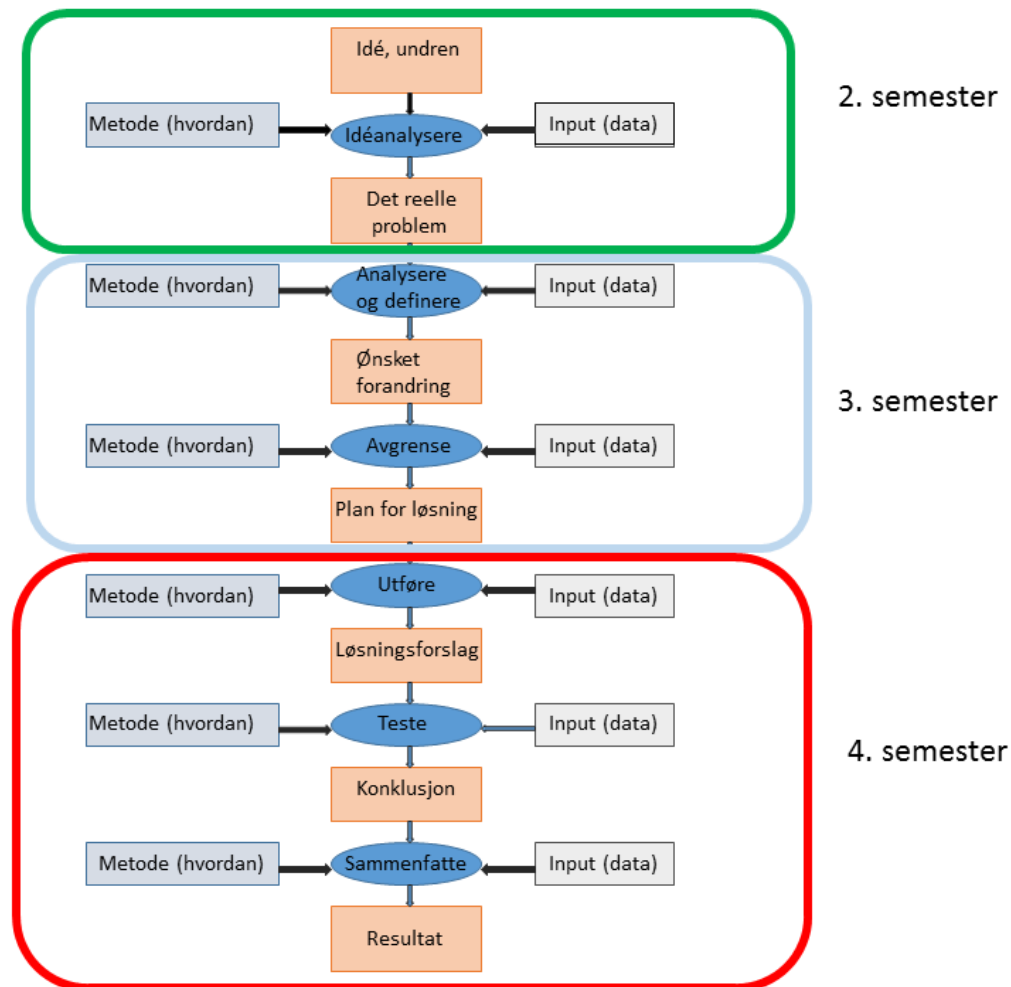
Det er også viktig å avstemme forståelseshorizontene mellom produsentgruppen og oppgavestilleren, som i dette tilfellet er NIBIO. Også her har samarbeid gjennom to semestre gjort at vi har fått bedre kunnskap om hverandres forståelseshorisonter. Når prosjektet startet opp visste ikke NIBIO hva vi kunne, og heller ikke konkret hva de ønsket å få ut av samarbeidet (annet enn å få økt verdi fra dataene de allerede samlet inn, gjennom å tilføre et "geografisk element"). Gjennom å holde flere møter, og gå metodisk til verks, kom vi frem til en forståelseshorizont som var såpass overlappende at vi var enige om hvilket problem som skulle løses, og for hvem. Vi opplevde også at vi ble enige om hvilket resultat vi ønsket å oppnå, hvilken teknologi vi skulle ta i bruk for å nå dette målet, og mer eller mindre hva som var et realistisk kompleksitetsnivå for oss å strekke oss mot i denne oppgaven.

For oss som har vært i produsentgruppen og som skulle utføre selve arbeidet med å få løsningen på plass, har også forståelseshorisonter vært et gjentakende tema. Kjennskap til hverandre og oppgaven gjennom to foregående semestre hjelper jo selvsagt, men vi har likevel opplevd at vi har hatt ulik oppfatning og forståelse for hvordan oppgaven konkret skal løses.

Det er selvsagt ønskelig at alle de tre partene i prosjektet er tilfredse med resultatet når prosjektet er ferdig.

### 2.3 Samtalemodellen

Vi ser på prosjektarbeidet som beskrives i denne rapporten som siste del av et større prosjekt. Arbeidet vi utførte på andre og tredje semester var forberedende og tilretteleggende for oppgaven som utføres i dette prosjektet. Bakgrunnen for en slik tanke er samtalemodellen som vist i Figur 6.



Figur 6: Samtalemodellen viser hvilke deler av prosjektet som skal utføres i hvilket semester.

### 2.3.1 Kort om samtalemodellen

Denne modellen er utviklet av Lars Brodersen (2014), og er tenkt å skulle fungere både som et planleggingsverktøy og et gjennomføringsverktøy i ulike prosjekter. De blå ovalene i modellen er prosesser. De rosa firkantene er en tilstand som fremkommer som resultat av prosessen og blir grunnlaget for den neste prosessen. Videre må det alltid annen input til for at prosessen skal finne sted, og en valgt metode forteller hvordan prosessen skal utføres. Ved hver rosa figur har man også muligheten til å kontrollere at man er på riktig kurs. Hvis man skulle finne at man ikke er på riktig kurs kan man hoppe et trinn tilbake og gjøre prosessen på nytt.

### 2.3.2 Samtalemodellen i dette semesterets prosjekt

Figuren viser hvordan vi på annet semester begynte med en idé/undren, og systematisk gikk i gang med å analysere denne for å komme frem til en reell problemstilling, som så ble utgangspunktet for arbeidet på tredje semester. Her definerte vi den ønskede forandring, avgrenset problemstillingen og utarbeidet en plan for løsning ved hjelp av metode og input som vi avdekket underveis i prosessen. Figuren viser også at fjerde og innværende semester skal begynne med å gjennomføre planen for løsning som resulterer i en prototype, at denne skal testes og konkluderes på, før man sammenfatter prosessen og et resultat foreligger. Det er denne prosessen vi skal gjennomføre i dette prosjektet, og dokumentere i denne rapporten.

I de to foregående semestrene har vi også opprettet fullstendige semesterspesifikke samtalemodeller som har guidet oss gjennom prosjektprosessen helt fra idé/undren til konklusjon. Det har vært en god arbeidsmetodikk å anse hvert semesters prosjekt som et selvstendig prosjekt som har en begynnelse og slutt, av praktiske hensyn. Dette har vært en selvstendig «modell-inni-modellen». I fjerde semester har vi valgt å ikke utarbeide en selvstendig samtalemodell for prosjektet. Da det er praktisk gjennomføring og prototype-bygging som er fokuset på prosjektet anser vi det ikke nødvendig å bruke fullstendig modell-inni-modell tilnærming. Vi legger derfor heller andre modeller og teorier til grunn for den praktiske gjennomføringen av prosjektets første del, men skal likevel være tro mot samtalemodellens metodikk i å tilføre metode og inputdata til hvert enkelt prosessledd, og komme med et løsningsforslag som testes og konkluderes på, før en sammenfatting av prosessen gir et endelig resultat. Samtalemodellen bygger på den vitenskapelige metode, og vi har lagt denne til grunn for arbeidet de foregående semestrene. Dette ønsket vi å fortsette med også i MTM-studiet og prosjektets avsluttende semester.

Mesteparten av denne rapporten handler om boksen som heter «utføre». Det meste av tiden dette semesteret har gått med til nettopp dette, og er beskrevet i kapitlene 3 og 4. Denne utføringen har resultert i et løsningsforslag, som vi omtaler som prototype i kapittel 4.6. Ved hjelp av metode og input har denne blitt testet av brukere, vi har beskrevet dette i kapittel 5. Vi har konkludert i kapittel 6, konklusjonskapittelet. Sammenfatningen av prosjektet har resultert i denne rapporten, som inneholder den dokumentasjon og de anbefalinger vi overdrar til NIBIO.

## 2.4 Flytdiagram

For at vi selv skulle få en oversikt over hvilke prosesser som logisk og teknisk sett ville inngå i løsningen vi så for oss, laget vi tidlig i prosessen et flytdiagram. Dette flytdiagrammet skulle i første omgang sikre overenskomst for gruppens interne forståelseshorison. Vi ville forsikre oss om at når vi snakket om hva de ulike komponentene vi arbeidet med skulle gjøre, hadde vi en klar felles forståelse. Dette ville sikre oss mot dobbeltarbeid og misforståelser, særlig ettersom det var snakk om teknologi vi ikke hadde veldig god kjennskap til før prosjektet begynte, og derfor til dels måtte utforske. Samtidig ville dette sikre at vi fikk med alle de logiske deler av prosjektet, og at vi ikke ved prosjektets slutt av en eller annen grunn manglet en "puslespillbrikke". Flytdiagrammet har gjennomgått mange revisjoner, og har forandret seg mange ganger etter hvert som vi har fått ny forståelse av hva oppgaven skulle inneholde, hvordan vi tror løsningen vil fungere i praksis, og hva de ulike teknologiske komponentene vi identifiserte var i stand til å gjøre rent konkret. Flytdiagrammet viste oss både hvordan løsningen var tenkt å fungere rent logisk - og vi oppdaget også at ved å gi en betegnelse til hver enkelt "boks" var det enklere å kommunisere og planlegge det praktiske arbeidet. Gruppens medlemmer kunne jobbe parallelt med ulike deler av løsningen, uten å komme i veien for hverandre. Flytdiagrammet som viser hvordan vi tenkte løsningen ligger i bilag III.

For enda bedre å holde oversikt over denne prosessen, utarbeidet vi en tabell som til enhver tid reflekterte elementene i flytdiagrammet. Her kunne vi skrive inn detaljer omkring prosessen, hvilken hypotese vi jobbet etter, hvordan testen av hypotesen hadde gått, hva som er status på hvert enkelt tidspunkt, og eventuelt en konklusjon. Slik ble modellen og tabellen også en måte å rapportere fremdrift på, samtidig som man lett holdt oversikt over hvilke deler av prosessen som var ferdige, hvilke som var underveis (og eventuelt hvilke problemer man hadde støtt på) og hvilke som ikke var påbegynt. Gruppen bestemte seg for å forsøke å bruke flytskjema og tabell som et prosjektstyringsverktøy, og se om det egnet seg, og i praksis fungerte slik som idéen opprinnelig var.

I praksis viste det seg at selve flytdiagrammet var godt egnet til bruken vi hadde sett for oss, og at det gjorde løsningen enkel og oversiktlig å kommunisere om, både oss imellom, og med utenforstående parter (for eksempel NIBIO eller veileder). Det ble løpende endret og utbedret, i takt med vår forståelse av løsningen. Tabellen viste seg å bli for omstendelig å oppdatere, og gikk derfor ut av bruk. Flytdiagrammet i bilag III viser prosessen vi forholdt oss til mens vi arbeidet med løsningen. I bilag IV er et flytdiagram som viser hvordan et endelig resultat kan se ut. I bilag V vises prosessen som den foreligger i prototypen.

På denne måten hadde vi en fleksibel prosjektstyring, hvor vi underveis evaluerte og justerte på målene. Selv om vi ikke bruker en agil utviklingsprosess formelt og systematisk i dette prosjektet, har utviklingsforløpet likevel lignet en agil prosess.

## 2.5 Designmodellen

I tredje semester brukte vi designmodellen for å skaffe oss en overordnet oversikt over den løsning vi da skulle frem til. Vi har tatt utgangspunkt i den samme modellen dette semesteret og videreført den i større detalj, og med de endringer vi bestemte oss for på begynnelsen av semesteret. Ettersom denne siste prosjektfasen stort sett dreier seg om implementering av en fungerende prototype basert på resultatene fra forrige prosjekt, har vi denne gangen brukt mye mer tid og innsats på å videreutvikle og diskutere innholdet i designmodellen.

Designmodellen skal være en generalisert gjengivelse av prosedyren med å utforme et system (Brodersen, 2008), og viser hvilke hensyn og egenskaper ved systemet som vi må ta stilling til. På denne måten fungerer modellen som en hjelp til å sørge for at man ikke utelater eller glemmer viktige elementer ved løsningen. Ved hjelp av modellen har vi kunnet ta stilling til ulike hensyn som brukerens utgangspunkt og bruk av funksjonen, formgivning og arkitektur, så vel som selve beregningen og det innhold den resulterer i.

Via designmodellen har vi utformet vår løsning både som en grafisk figur, og i et tilhørende dokument som gir en verbal beskrivelse av de ulike deler av modellen. På denne måten har vi gått gjennom de ulike prosjektfaser, fra prosjektets forankring og verdigrunnlag, til prototype til resultat. Til en viss grad har vi også ansett modellen som et hjelpemiddel i prosjektstyringen, ettersom de ulike deler av modellen også gjenspeiler en prosjektfase.

I likhet med flytdiagrammet har designmodellen fått enkelte mindre justeringer underveis, men vi synes likevel at den har stått seg godt og beholdt sin gyldighet gjennom hele prosjektet. Designmodellen i sin helhet blir omtalt i rapportens kapittel 3.2, og figuren ligger i bilag II.

## 3 Løsningsdesign

Etter tredje semesters prosjekt hadde vi en formening om hva vår anbefalte løsning i hovedsak burde inneholde, og hva den skulle gjøre. Det betyr selvfølgelig ikke at prototypen var ferdig designet – den mentale forestillingen vi hadde var overordnet og mangelfull. Vi måtte derfor i denne siste fasen definere detaljene i de elementene som til syvende og sist skulle settes sammen til en prototype.

### Hypotese for løsningsdesign:

Vi har identifisert noen grunnleggende problemstillinger i forhold til løsningen vi har planlagt å lage. Hvis vi arbeider systematisk med disse problemstillingene vil vi få et bedre grunnlag for å trekke overordnede beslutninger om løsningens funksjonalitet.

Hvis vi bruker designmodellen til å gjennomgå alle aspekter av løsningens design, vil vi kunne treffe de øvrige valg som tilpasser løsningen best mulig til brukerens oppgaveløsning, og som samtidig tar hensyn til de ressurser vi har tilgjengelig. Heretter vil vi kunne designe den logiske dataflyt i løsningen, samt utarbeide et forslag til løsningens tekniske design.

### 3.1 Problembeskrivelse

Vi har i løpet av prosjektarbeidet identifisert noen problemer som vi må undersøke og finne ut av før den endelige løsningen kan designes:

#### Interpolasjon:

- Det er stor variasjon i avstanden mellom målestasjonene i LMT-nettverket. Det kan gi utfordringer ved valg av interpolasjonsmetode.
- Ved interpolasjonen må vi huske på at stasjonene ligger på ulikt antall meter over havet.

#### Korreksjon av resultatet:

- Det er også andre faktorer, ifølge litteraturen, som kan inngå i korreksjon av verdiene for å få mest mulig nøyaktig resultat, for eksempel avstand til kyst, terrenghelning og –orientering (*aspect*).

#### Visualisering:

- Hvis resultatet skal fargelegges automatisk, må det velges en metode for det som rommer de store variasjonene som kan forekomme i datatilfanget, uten at brukeres villedes ved tolkningen av dataene.

#### 3.1.1 Interpolasjon av varmesum mellom stasjoner i LMT

LMTs stasjonsnettverk består av godt og vel 80 målestasjoner som er plassert i de viktigste landbruksområdene i Norge. Mange av stasjonene i nettverket ligger så langt fra nærmeste nabostasjon at det kanskje ikke er mulig å bruke disse som interpoleringsgrunnlag. Dette vil i så fall medføre at vår varmesumsløsning ikke vil være i stand til å vise data for store deler av landet. Vi ville derfor først analysere avstandene mellom LMTs værstasjoner for å finne ut om det i det hele tatt gir noen mening å interpolere når stasjonene er så spredte.



### 3.1.2 Analyse av stasjonstetthet i LMT

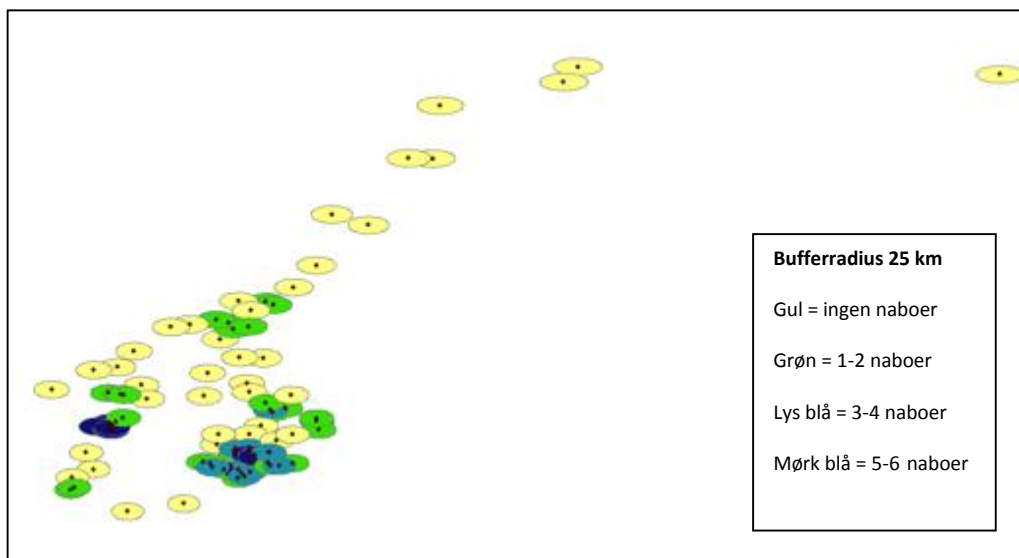
Ved hjelp av analyseverktøy i ArcGIS gjorde vi en analyse av stasjonstettheten i LMT-nettverket, og fant frem til følgende avstandsverdier (Tabell 1). Utgangspunktet for analysen er at vi mener at det bør være minst tre stasjoner innen en viss søkeradius for at interpolasjon gir noen mening.

Avstandsmål	
Gjennomsnittlig avstand mellom stasjoner	31 km
Gjennomsnittlig avstand til 3 nærmeste stasjoner	62 km
Maksimal avstand til 3 nærmeste stasjoner	594 km
Antall stasjoneruten naboer i 25 km radius	39 stk.
Antall stasjoneruten naboer i 40 km radius	16 stk.

Tabell 1: Analyse av avstander mellom stasjonene i LMT

Tabellen viser at man i gjennomsnitt må bruke en søkeradius på 62 km for å kunne hente data fra tre nabostasjoner, og at den maksimale søkeradius er på nesten 600 km dersom alle stasjonene skal få minst tre nabostasjoner med.

Vi lagde dessuten et kart som viser hvor i Norge stasjonene ligger henholdsvis tett på og langt fra hverandre (Figur 7).



Figur 7: Kart over LMTs værstasjoner med 25 km buffer fargelagt etter antall nabostasjoner

Kartet viser – ikke overraskende – at stasjonstettheten er stor i det sørlige Norge, som også har høyest befolkningstetthet, og mest landbruksjord, og at det i Nord-Norge kan være svært langt mellom stasjonene.

Vi forela problematikken for en ekspert hos Meteorologisk institutt i Oslo for å høre hvor stor søkeradius de anså for å være forsvarlig. I Meteorologisk institutts tjenester brukes en søkeradius på opptil 200 km, noen ganger enda mer (Tveito, pers. medd.). Med så stor søkeradius vil vi bare ha noen få tilfeller hvor det ikke finnes nok nabostasjoner til å kunne utføre en forsvarlig interpolasjon, skjønt vi vil jo gjerne at løsningen skal kunne håndtere også slike tilfeller. Det vil derfor være nødvendig å sette opp en rutine i samleprosessen som undersøker om det finnes nok nabostasjoner med gyldige målinger for den valgte perioden før interpolasjonen kjøres. Og hvis det ikke kan interpoleres, kan det likevel være ønskelig om man kunne foreta en høydekorleksjon bare ut fra den målte verdi på den valgte stasjon. Disse funksjoner er bygget inn i det logiske flytskjemaet (bilag III).

Vi utførte dessuten en analyse av hvor stor påvirkningen fra henholdsvis interpolasjonen og høydekorleksjonen var i et mindre testområde, med det formål å avklare om vi eventuelt skulle droppe interpolasjonen og heller fokusere på høydekorleksjonen, dersom denne skulle vise seg å ha en vesentlig større påvirkning på de endelige resultatene enn interpolasjonen. Analysen kan ses i sin helhet på LMTs testserver<sup>4</sup>. Konklusjonen var imidlertid ikke entydig – særlig i de lavtliggende områdene hadde høydekorleksjonen relativt mindre påvirkning på sluttresultatet enn interpolasjonen (noe som ved ettertanke ikke burde være overraskende), og vi valgte derfor å holde fast på interpolasjonen som en del av løsningen.

### 3.1.3 Bruk av supplerende målinger

Vi vet at det måles temperatur i hele landet, gjennom Meteorologisk institutts værstasjoner. Dette er autoritative data som meteorologisk institutt deler med omverden helt gratis, blant annet gjennom Web servicen "WSKlima" som er tilgjengelig på [eklima.no](http://eklima.no)<sup>5</sup>. Trolig ville vår varmesumsløsning ha blitt bedre dersom også dataene fra [eklima.no](http://eklima.no) var med i løsningen. Vi har likevel valgt å ikke ha disse dataene med inn i prototypen vi utvikler, fordi det i utgangspunktet er enklere å forholde seg til leveranser fra en enkelt leverandør - i dette tilfelle LMT. Å tilrettelegge for, og samkjøre data fra to ulike leverandører, levert på to ulike formater gjennom to forskjellige tjenester, og tilpasse dataene slik at de blir direkte sammenliknbare og kan vises i en felles presentasjon mener vi vil ta uforholdsmessig mye tid og arbeid i denne fasen av arbeidet. Vi har derfor valgt å utelate dataene som tilbys fra denne tjenesten i prototypen.

### 3.1.4 Håndtering av forskjellige stasjonshøyder ved interpolasjon

Norge er et kupert land, og selv om mye av landbruksjorden ligger lavt i terrenget, er det store forskjeller i LMT-stasjonenes høyde over havet. Vi vet at temperaturen avtar med 0,6°C pr. 100 m over havet (Køltzow, et al., 2014). Når temperaturene er målt på forskjellige høyder er de ikke sammenliknbare, og en direkte interpolasjon mellom de målte verdier vil ikke gi riktig resultat. En påfølgende høydekorrigering ved hjelp av DTM vil heller ikke bli riktig.

Meteorologisk institutt kjente også til dette, og kunne fortelle at i deres beregninger blir problemet håndtert ved at stasjonenes målte temperatur korrigeres ned til havnivå før man interpolerer mellom dem. Deretter korrigeres resultatet opp til terrengnivå igjen ved bruk av DTM (Tveito, pers. medd, 2015). Dette virket å være en god metode, og det var heller ingen større tekniske problemer med implementeringen av den. Vi har derfor brukt denne metoden i løsningen.

---

<sup>4</sup> <http://lmttest.nibio.no/mtm/VedleggTilRapport/Hoydeanalyse.pdf>

<sup>5</sup> [www.eklima.no](http://www.eklima.no) - Meteorologisk institutt - konsultert 4. oktober 2015

### 3.1.5 Korreksjon for øvrige forhold

I de foregående oppgavene har vi nevnt at de interpolerte dataene i tillegg til å bli korrigert for høyde, kunne blitt korrigert for avstand til hav/kyst, samt innsjøer og fjorder, da dette trolig vil påvirke temperaturen. Det kan være grunn til å tro at det samme vil gjelde for helningsgrad og helningsvinkel fordi dette kan påvirke solinnstrålingen, og dermed også temperaturen. Vi tok tidlig avgjørelsen om å ikke ta med disse elementene inn i analysen. Det var flere grunner til dette: vi ønsket heller å fokusere på de ulike delelementene i oppsettet av Web-GIS løsningen, hvor selve analysen kun er en enkelt del. I samråd med GIS-eksperter i NIBIO (Nystuen, pers. medd.), og på meteorologisk institutt i Oslo (Tveito, pers. medd.), ble det dessuten klart at slike analyser er meget krevende, og derfor kun utføres i enkeltstående tilfeller og begrenset grad per i dag. Meteorologisk institutt oppgir for eksempel at de ikke korrigerer for annet enn høyde i sine temperaturkart på [senorge.no](http://senorge.no). Å lage gode og korrekte korrigeringer som tar høyde for alle disse faktorene er altså et meget omfattende arbeid, og trolig grunnlag for helt egne studier og prosjekter. Vi går derfor ikke videre inn på dette i vår oppgave.

### 3.1.6 Fargeskala

Interpolasjonen og den etterfølgende høydekorrigeringen vil resultere i en rasterfil, som uten bruk av stylingsparametere bare vil vise et ugjennomsiktig gråtonebilde. Vi måtte derfor definere et sett med visningsparametere, via en SLD-fil, som kunne presentere rasterverdiene på en mer intuitivt forståelig måte. Det er også viktig for brukerne å kunne se grunnkartet «gjennom» det interpolerte rasteret, så vi må i tillegg sette opp en viss transparens i rasterbildet.

Avhengig av lengden på den valgte perioden, hvilke årstider perioden dekker og valgt basistemperatur, vil omfanget i dataverdiene for varmesummen (celleverdiene) kunne variere fra nær null til i prinsippet flere tusen døgngader. Dette innebærer at vi måtte finne en metode for styling som kunne håndtere store variasjoner i verdispenn fra den ene beregningen til den andre. Vi måtte dermed finne en metode og en fargeskala som gir et lesbart og forståelig resultat uansett hvor stort eller lite verdispenn skalaen skal dekke.

Vi har tidligere funnet ut følgende:

- Varmesummen er kvantitative data, men kan i visualiseringsammenheng betraktes som rangordnede data (fra lavest til høyest).
- Bruk av en fargeskala med forskjellige farger er ikke velegnet til visualisering av kvalitative data eller rangordnede data (jf. (Brodersen, 2008)).
- Bruk av svertningsgrad kan være anvendelig til visualisering av rangordnede data. Svertningsgrad kan benyttes for kun én farge. Dette er heller ikke helt uproblematisk, men det finnes gode eksempler hvor dette er brukt (Brodersen 2008).
- NIBIO har gitt tilbakemelding om at en *stretched* skala (trinnløs fargeskala) er forvirrende; de ønsker noen klart avgrensede intervaller, så de lett kan se hvilket intervall en celle tilhører.
- Rekkevidden av variabelen svertningsgrad på arealobjekter ligger mellom 7 og 11 trinn (Brodersen, 2008).

Hvis fargelegging av rasteret skal skje automatisk og ut fra ovenstående prinsipper, ser vi bare to mulige metoder:

1. Vi lager en skala på inntil 10 trinn og med faste intervaller, som dekker det forventede maksimale omfang av rasterverdier (for eksempel 2000 graddager), slik at hvert trinn dekker 1/10, i dette tilfellet 200 graddager.
2. Vi bruker en skala med inntil 10 trinn, hvor hvert trinn dekker 10 % av verdispennet i det aktuelle rasteret.

Fordelen med metode 1 er at en gitt farge eller svertningsgrad alltid svarer til samme verdi, så en fast bruker enkelt kan tolke resultatet, fordi han/hun kjenner skalaen. Ulempen er likevel at hvis det er valgt en forholdsvis kort periode som analysegrunnlag, vil alle verdiene i rasteret ligge innenfor det samme intervallet, og man kan dermed ikke se variasjonen i varmesummen rundt det valgte stasjonen. Det vil dessuten kunne forekomme verdier som ligger utenfor det forventede omfang, og dermed utenfor skalaen.

Fordelen med metode 2 er at man alltid vil kunne se en variasjon, uansett hvor liten den er. Ulempen kan være at brukeren forledes til å tro at det er en større variasjon enn det virkelig er når alle trinn er i bruk, også når omfanget i verdiene er lite. Resultatet av ovenstående drøftelser er at vi foreløpig velger metode 2, som er det vi vil teste i prototypen. De tekniske detaljer beskrives nærmere i kapittel 4.4.4

## 3.2 Løsningsdesign via designmodell

På dette stadiet hadde vi tatt en del overordnede beslutninger, men nå skulle vi bestemme helt konkret hvilke «skruer og muttere» som prototypen skulle bestå av.

Vi har brukt designmodellen til å utarbeide detaljene i så vel prototypens ide og verdigrunnlag som i de mer tekniske elementene den skal bestå av. Vi startet dette arbeidet på et tidlig stadium i prosessen, men arbeidet også delvis videre med den parallelt med vår testing av løsningsforslagene.

### 3.2.1 Designmodellen

I avsnittet under har vi beskrevet modellen vi kom frem til i tekst. Punktene refererer til de ulike fasene og bestanddelene designmodellen består av. Selve designmodellen er vedlagt som bilag II.

#### **1: Ide**

*I LMT finnes det allerede en funksjon for å beregne varmesum med basis i målingene fra en gitt stasjon. Denne varmesummen består av en tabell som gjelder for det aktuelle punktet som stasjonen står i. Vår ide er at denne funksjonen kan forbedres og utvides ved hjelp av geovisualisering. Ved å interpolere varmesummer mellom flere stasjoner, kan landbruksrådgiverne få beregnet /visualisert varmesum på steder hvor denne ikke er målt. Vi mener også at vi ved å høydekorrigere det interpolerte rasteret, kan levere et troverdig resultat som tar hensyn til variasjoner i temperatur i ulik høyde over havet.*

#### **2: Verdimodellere**

*En høydekorrigert interpolering av varmesum bygger på noen sentrale verdier: Den vil gi landbruksrådgiverne mer detaljerte temperaturoversikter. Den vil også tilføre ny informasjon til varmesummen. Dette vil i sin tur føre til en forbedret situasjon for rådgiverne, fordi de med dette verktøyet kan gi bedre informasjon til medlemmene. Interpolasjonen over et område rundt værstasjonen vil dessuten kunne vise brukerne varmesummen for utvalgte områder utover selve stasjonspunktet.*

### **3: Tenkt prototype**

Prototypen vår tar utgangspunkt i de eksemplene vi lagde i tredje semester. Oppgaven går nå ut på å designe en metode for produksjon av interpolert temperatur, med utgangspunkt i eksisterende data i LMT.

LMT bruker en PostGreSQL-database og Open Layers som webverktøy, og vi vil bruke en arkitektur som er kompatibel med disse. Prototypen skal benytte Open Source-verktøy. Prototypen skal bestå av et brukergrensesnitt og brukerdiallog, de prosesser som må til for å generere resultatet, og en utforming og presentasjon av det interpolerte kartet.

#### **4 a og b: Formmodellering**

Formmodelleringen går ut på å bestemme hvilken arkitektur vi skal bygge prototypen på (verktøy/apparat), og dessuten hvilken utforming vi ønsker å gi interaksjonen, prosessen og resultatet.

##### **4 b: Form:**

Fra tidligere semestre vet vi at mange av brukerne ikke har rettet noen spesiell oppmerksomhet mot kartet på LMTs forsiden, og at mange ikke engang er innom forsiden – de går i stedet rett til stasjonssiden for sin nærmeste værstasjon. Dette kan ses som en konsekvens av nåværende LMTs form, altså at kartet er lite brukt fordi det ikke befinner seg på sidene som brukes. På bakgrunn av dette velger vi derfor å legge selve interpolasjonsløsningen på hver enkelt stasjons side, som ofte er brukernes faktiske inngangsside.

##### Apparat:

Til å begynne med antok vi at vi måtte gå inn i LMTs tabeller og kartlegge hvilke data som fantes, og lage et databaseview som inneholdt de nødvendige data, samt geometri i et egnet koordinatsystem. NIBIOs webutvikler foreslo imidlertid på et tidlig tidspunkt at vi kunne henvende våre databaseforespørsler mot en webtjeneste som han tilbød seg å lage. Vi starter dermed med en webtjeneste mellom vårt system og LMT/PostGreSQL. Vi ville fremstille de øvrige LMT-dataene som et geodatalag ved hjelp av Spatial view i PostGIS og OpenLayers. Interpolasjonen skulle skje automatisk ved hjelp av prosess tjenester basert på åpen GIS-programvare. Resultatet skulle sendes til Geoserver/WMS-tjeneste, og visualiseres ved hjelp av et SLD-oppsett. Webkomponenter som GetFeatureInfo, pan og zoom og så videre skulle settes opp i OpenLayers, og resultatet sendes til klimastatistikkfanen på LMTs stasjonsside - siden forespørselen kommer fra.

##### Visualisering/form:

Varmesummen visualiseres med en standard fargeskala som er lett å forstå, og som kan fungere i alle potensielle verdispenn.

##### Øvrige elementer:

Resultatsiden må i tillegg inneholde metadata. Det må genereres en tegnforklaring som viser varmesumverdiene. I tillegg er det behov for noen forklarende tekster som beskriver inputparametrene for beregningen (start- og sluttdato, stasjonsnavn og basistemperatur), en

enkel veiledning for navigering i kartet, og en tekst som forteller hva varmesumkartet beskriver – og ikke beskriver.

### **5 a og b: Innholdsmodellering**

Løsningen skal bestå av et rasterkart med interpolerte temperaturer. Interpolasjonen skal høydekorrigeres. Interpolasjonen må ta hensyn til at stasjonene befinner seg på ulik høyde over havet, noe som innebærer at varmesummene må reduseres til havnivå før interpolasjon. Justering tilbake til terrenghøyde gjøres i en høydekorreksjon mot en DTM.

Rasteret skal ha celler på 1x1 km, for å markere usikkerheten i kartet. Av samme grunn skal interpolasjonen bare vises i en gitt avstand fra målestasjonen.

Det skal genereres en tegnforklaring over den symbologi som skal settes inn i kartet. Kartet skal vises i en fornuftig målestokk/zoomnivå i forhold til det klipte, interpolerte rasteret.

I en egen «faktarute» ved siden av kartet skal det oppgis metadata som beskriver innholdet, og som kan angi noe om nøyaktigheten i beregningen, hvilke forbehold som gjelder og så videre.

Kartet bør vises oppå et bakgrunnskart, for eksempel et topografisk kart over Norge, så brukeren har mulighet til å orientere seg rundt den valgte værstasjonen. Likeledes bør værstasjonene vises i kartet som punktobjekter, med en påskrift som angir stasjonsnavnet.

### **6 a og b: Interaksjonsmodellering**

**6 a:**      Interaksjonsmodellere:

Vi har her tatt utgangspunkt i våre tidligere samtaler med rådgiverne, og satt opp de situasjoner og spørsmål vi anser som relevante.

Funksjonen er beregnet på en rådgiver i landbruksrådgivningen, som søker grunnlag for å gi informerte råd til medlemmene om hvorvidt forholdene ligger til rette for såing, sprøyting, vanning, høsting et cetera for gitte avlingstyper. Brukeren vil starte funksjonen på valgt stasjonsside i LMT, under fanen «Klimastatistikk». Vi har tatt utgangspunkt i en rådgivers oppgaver, og satt opp en liste over de spørsmål en rådgiver kan forventes å stille.

Forventet spørsmål	Våre svar	Inkluderes
Hva er varmesummen for en gitt periode på et gitt punkt/mindre område (i nærheten av en værstasjon)?	Forventet svar: et tall, som i utgangspunktet bare gjelder for stasjonspunktet.	Ja
Hvor stort er avviket fra normalen?	Svar kan ikke leveres – stasjonene har for ulik historikk, og normaldata er ikke tilgjengelige.	Nei
Er det korrigeret for at min eiendom ligger på en nordskråning/i nærheten av en isbre/ved havet/etc?	Svar i disclaimer/metadata om tjenesten: Dette er for kompliserte spørsmål for tjenesten, og det er per i dag vanskelig å beregne slikt (Tveito, pers. medd.).	Nei

Tabell 2: Forventede spørsmål og svar til løsningen

Vi har dermed «bestemt» hvilke spørsmål som skal kunne stilles til, og besvares av funksjonen. Det spørsmål som skal besvares før kjøring av funksjonen er beregning av varmesum. Aktuelle parametere for å stille dette spørsmålet er valg av stasjon, valg av periode, og valg av basistemperatur. Dette skjer gjennom en meny på stasjonssiden.

#### 6 b: Interaksjon:

Start av funksjon: Brukeren velger stasjon, periode og basistemperatur. Dette skjer i en meny som ligner dagens oppsett (se Figur 8. )

Figur 8: Brukergrensesnitt i den eksisterende varmesumberegningen i LMT

Ulike brukere kan ha individuelle behov for svar i løsningen. Det er derfor viktig å gi dem en mulighet for å kunne interagere med resultatkartet. Kartet skal derfor ha mulighet for klikkbar info om varmesum fra en rastercelle (Get Feature Info). Det skal dessuten også gis muligheter for zoom- og pan-funksjoner som tillater navigering i kartet, innzooming til særlig interessante områder og så videre.

#### Feilhåndtering

Det kan forekomme stasjoner som har NULL-verdier i målingsfeltene. Dette kan komme av ulik oppstart av målinger, eller at noe har feilfunget i en kortere periode. I så fall kan en interpolasjon av stasjonene rundt bli meningsløs. Vi bør ha en tilbakemelding til bruker om dette, slik at bruker eventuelt kan korrigere perioden, eller velge en annen stasjon. Dersom det

ikke skulle være andre stasjoner nær nok til å interpolere mot, bør vi også tilby at vi høydekorrigerer et raster som kun inneholder den eventuelt ene temperaturen for valgt stasjon, slik at denne kan tilpasses topografien.

### 7: Informasjonsforedle

Når vi har fått tak i måledataene fra LMT, bør alle verdier reduseres i henhold til stasjonens høyde over havet, slik at interpolasjonen kjøres på havnivå (Tveito, pers.medd.). Etter interpolasjonen må resultatet korrigeres tilbake til reell høyde, ved hjelp av en formel for temperaturfall i forhold til stigning.

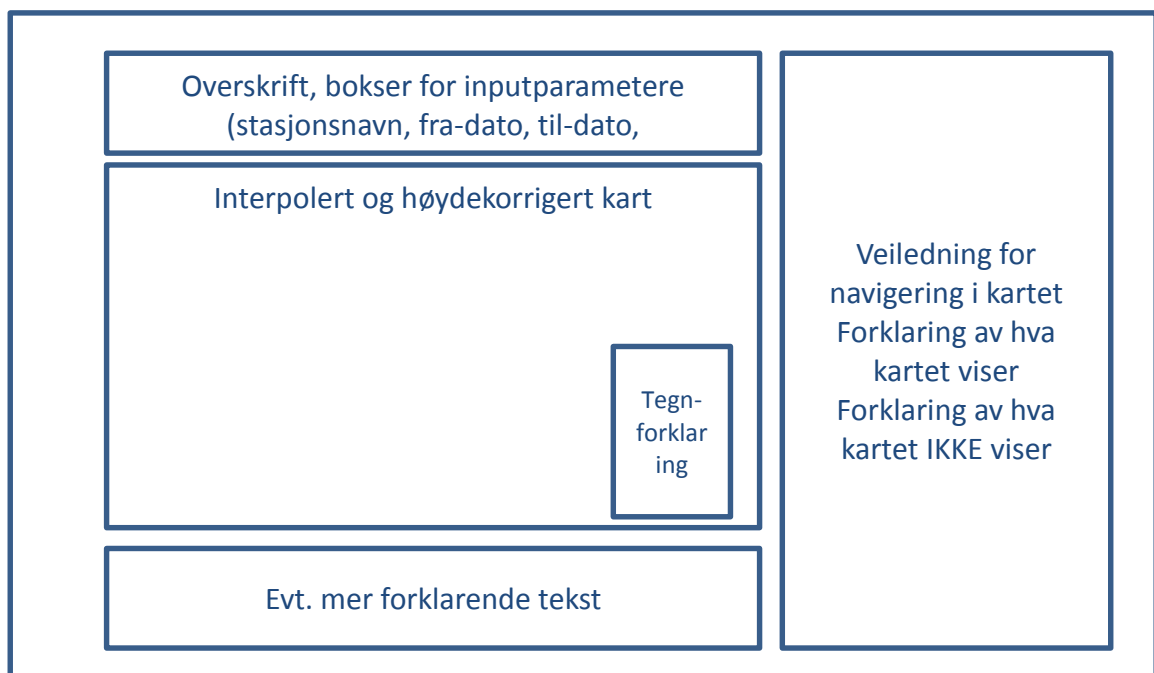
Rastercellene vil i utgangspunktet ha den oppløsning vi har i DTMen vi velger å bruke, i utgangspunktet 50x50 m. For å understreke usikkerheten i den beregningen vi utfører, vil vi regne om rastercellene til 1000x1000m, for å skape et «grovere» uttrykk.

For at rasterkartet og bakgrunnskartet skal passe best mulig sammen må de ha det samme koordinatsystemet, for eksempel UTM sone 33 (EPSG:32633).

Som videre foredling av resultatet, vil vi utforme en presentasjon av kartet som kan vises i en webside, og med en fargeskala som kan håndtere den store variasjonen vi vil få i verdipennet. Kartet skal settes inn på stasjonssiden via en IFRAME. Til dette skal det tilføyes forklarende tekster og tegnforklaringer.

### 8: Resultat

Resultatet skal være et kartutsnitt som viser interpolert varmesum, som brukeren kan zoome og panne i, og som har tilgang til FeatureInfo. Det bør være mulig å laste ned kartbildet som en fil. Kartet skal ha en tilhørende tegnforklaring, og vi skal vise metadata og disclaimere på et godt synlig sted i vinduet.



Figur 9: Wireframe over hvordan kartsiden kan se ut



## **9: Distribuere**

*Distribusjon av kartet gjøres via oppsett av kartutsnittet i en webside, som settes inn på stasjonssiden under fanen klimastatistikk. Dette er den websiden funksjonen startes fra, og hvor parameterne legges inn, og det er her resultatet bør komme tilbake. En eventuell videre distribusjon til medlemmene gjøres av rådgiveren via dennes vanlige kanaler og verktøy.*

Gjennomgang av designmodellen har hjulpet oss til å få overveid og avgjort de beslutninger vi måtte ta for å kunne finne ut av hvordan løsningen skal skrues sammen. Det vil si hvilke prosesser som skal skje i hvilken rekkefølge for at de data vi henter ut fra LMT kan bli til informasjon på det varmesumkartet vi gjerne vil lage.

### **3.3 Modellering av dataflyt**

Selv om vi nå hadde et godt og presist bilde av hva prototypen skulle bestå av, og hvilke behov den skulle fylle, hadde vi fortsatt ikke en helt presis beskrivelse av akkurat hva som skjer i hvilken rekkefølge, ved hjelp av hvilke verktøy og komponenter. For å utvikle dette måtte vi gå dypere inn i særlig punkt 7 (Informasjonsforedle) i designmodellen, og se detaljert på hvilke prosesser som må/kan utføre de operasjonene vi har spesifisert, og hvordan data flyter gjennom prosessen.

#### **3.3.1 Logisk flytdiagram**

Vi bestemte å lage et flytdiagram som kunne håndtere programlogikken gjennom hele prosessen. I flytdiagrammet satte vi opp alle interaksjoner og operasjoner vi hadde kommet frem til, og beskrev sammenhengen og avhengighetene mellom dem.

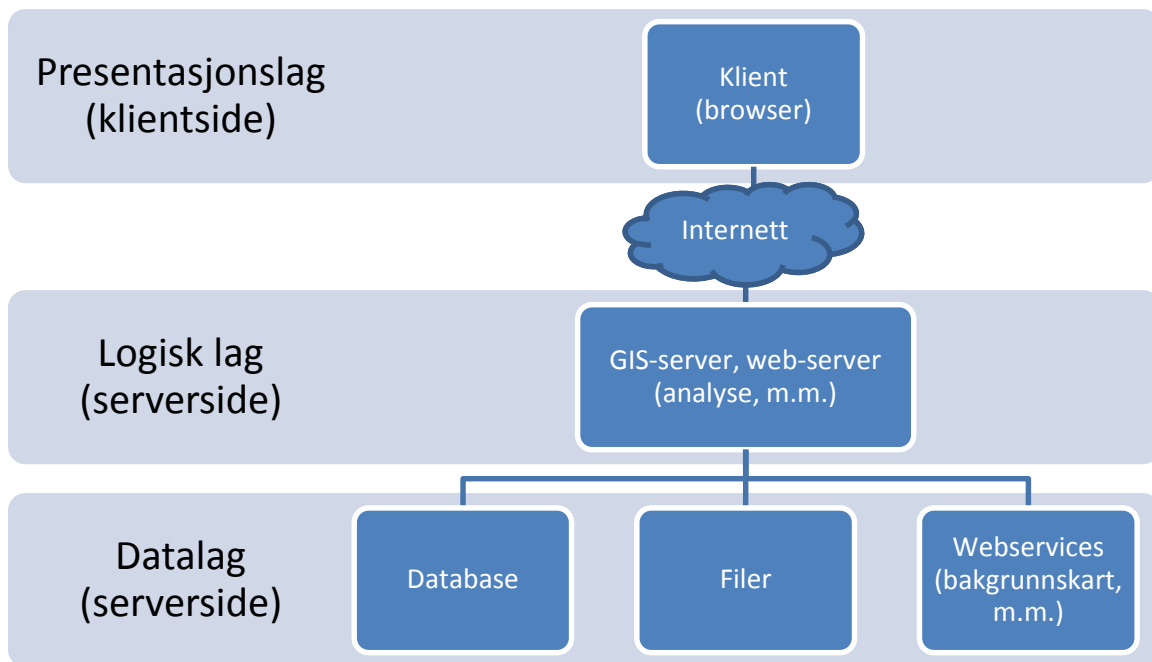
Etter flere gjennomganger har vi kommet frem til en grafisk fremstilling av alle nødvendige prosesser og interaksjoner, og den logiske flyten mellom dem, se bilag III.

Boksene i flytdiagrammet kan grovt sett deles inn i fire hovedkategorier: Brukerinput, utvalg av data fra LMT, geoprosessering og presentasjon. Mellom disse kategoriene foregår det en transport av data, og eventuelle kontroller for at neste prosess kan gjennomføres. Det var dessuten ved utarbeidelsen av flytdiagrammet at vi identifiserte de forhold som kunne føre til feil i beregningen, eller til at beregning ikke kunne utføres.

### **3.4 Teknisk arkitektur**

Etter å ha laget et utkast til hvordan dataflyten og de logiske prosesser skulle settes sammen gikk vi i gang med å se på hvilke komponenter som skulle utføre de enkelte delene av den samlede prosessen. Ettersom vi ikke har mye erfaring i oppsett av webløsninger undersøkte vi hvilke anbefalinger vi kunne finne i litteraturen om webGIS-løsninger.

I følge Fu & Sun (2011) består et webGIS av en klient-serverstruktur, som regel i en tredelt lagstruktur med et datalag, et logisk lag og et presentasjonslag. Avhengig av graden av interaksjon og GIS-funksjonalitet kan det logiske laget romme en eller flere servere (Figur 10).



Figur 10 Tredelt lagstruktur for webgis

Kommunikasjonen mellom de enkelte lag og komponenter skjer ved hjelp av standarder for kommunikasjon på nettet, for eksempel http (protokoll for utveksling av informasjon), HTML (dokumentformat/språk) og URL (standard for angivelse av adressen på informasjonen) (Fu & Sun, 2011). GIS-komponentene arbeider også seg imellom med forskjellige geografiske utvekslingsformater, for eksempel shape-formatet, GeoJSON, GeoTIFF, som det er nødvendig å kjenne til for å kunne lage en fungerende og hensiktsmessig dataflyt i løsningen.

I designet av en webGIS-løsning skal man også ta stilling til fordelingen av oppgaver mellom server og klient. Fu & Sun opererer med tykke og tynne klient-oppsett, hvor større eller mindre deler av oppgavene løses på henholdsvis server eller klientsiden. I denne sammenheng påpeker de at trenden i øyeblikket går mot tykkere klient-oppsett, hvor klientsiden løser større/mer avanserte deler av de samlede oppgavene gjennom bruk av for eksempel plugins og avansert javascript. Her skal man selvsagt vurdere begrensinger (for eksempel båndbredde), hvis det kreves at det overføres store mengder data til klienten.

En annen trend er at flere og flere geografiske analysemetoder finnes tilgjengelige som webservices, som såkalte Web Processing Services (WPS). Fordelene med å bruke disse servicene er at man slipper å programmere funksjonaliteten selv, og det er enklere å skifte ut en service, hvis det har blitt utviklet bedre versjoner (Fu & Sun, 2011).

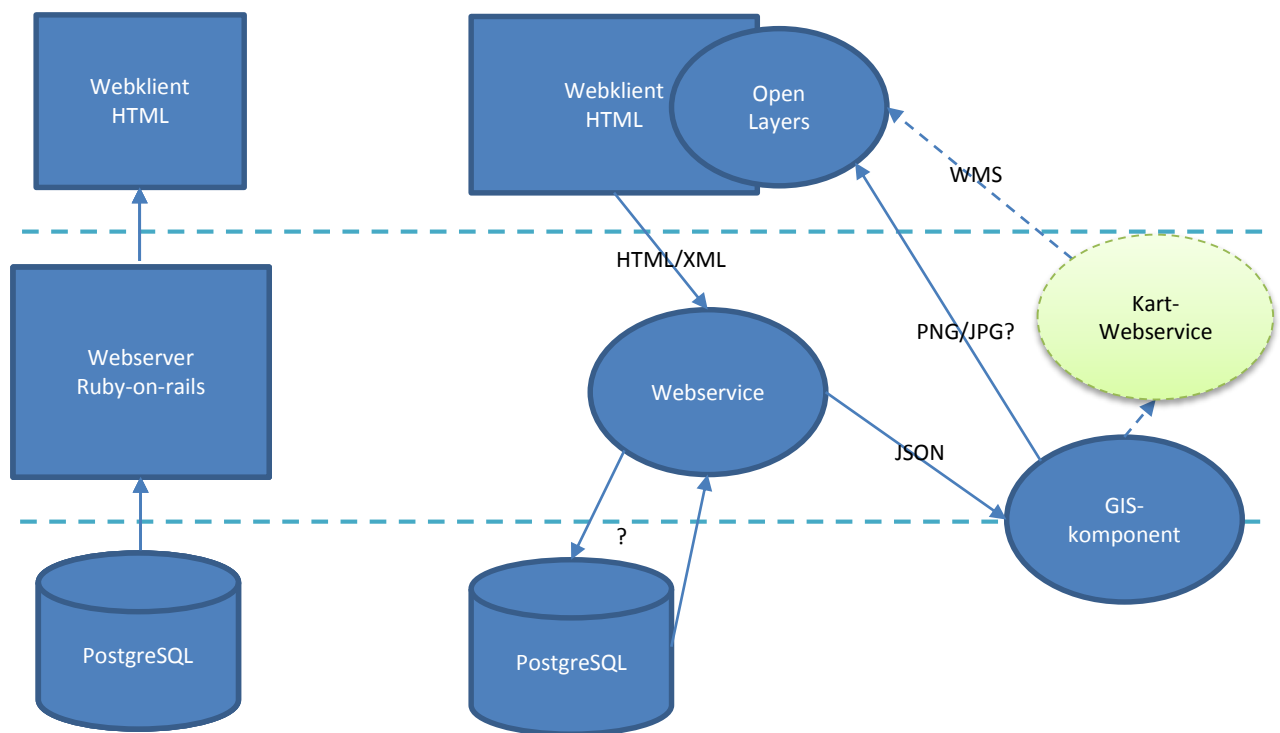
For å komme frem til en fungerende webGIS-løsning som oppfyller brukerens behov, må vi være oppmerksomme på følgende:

- Hvilke data fra hvilke datakilder skal inngå i løsningen, og hvordan skal disse være formatert?
- Hvilke softwarekomponenter kan ivareta de forskjellige prosessene på henholdsvis serversiden og klientsiden, og hvilke prosesser kan eventuelt løses av webservicer?

- Hvordan kan de forskjellige komponentene kommunisere med hverandre?
- Hvordan kan den samlede prosessen automatiseres, sånn at brukerens input automatisk utløser den ønskede geovisualisering?

### 3.4.1 Prinsippskisse over systemarkitektur

Etter å ha gått gjennom og analysert de ulike funksjonene vi skulle utarbeide, og de mulige problemstillingene vi måtte løse i prototypen, laget vi en oversikt over hvordan komponentene burde henge sammen og samspille. Utgangspunktet for våre valg var den eksisterende plattformen i LMT. Modellen (Figur 11) viser sammenhengen i prosessene med utgangspunkt i den eksisterende websiden, innhenting av data fra PostgreSQL-basen, overføring av data til GIS-komponenten for analyse, og overføring av det formaterte resultatet tilbake til websiden.

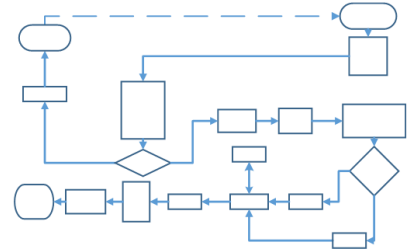


Figur 11: Prinsippskisse over mulig systemarkitektur. Komponentene til venstre er arkitekturen i dagens LMT. Figuren til høyre viser den mulige arkitekturen vi så for oss for prototypen. Ut fra denne modellen kan vi identifisere de overordnede komponenter som må inngå i løsningen.

## 4 Valg og utprøving av tekniske komponenter

I dette kapittelet har vi lagt inn små oversiktsbilder av flyttdiagrammet. Miniatyrene vil bli brukt til å markere hvor i prosessen de ulike delprosessene befinner seg.

LMT-løsningen er basert på Open Source-software, og det er NIBIOs ønske at vi så langt som mulig baserer våre programkomponenter på det samme. Det er derfor innen dette miljøet vi vil søke etter løsninger til utarbeidelse av prototypen. Prosjektgruppens primære erfaring ligger i bruk av proprietær desktop software, og vi har på studiet blant annet brukt ArcGIS til løsning og automatisering av avanserte oppgaver. Vi visste derfor av erfaring at de funksjoner vi ville bruke finnes, og at de kan settes sammen til mer komplekse prosedyrer ved hjelp av modell- og programverktøy. Vi hadde imidlertid mindre erfaring med åpen programvare, men antok likevel at de samme operasjonene lot seg løse etter lignende prinsipper som vi allerede kjente til.



### Problemstilling

Vi skal lage en webløsning som på et kart kan vise varmesummen for en brukervalgt periode omkring en brukervalgt værstasjon, hvor varmesummen er interpolert mellom nabostasjoner og korrigert for terrengforskjeller med  $-0,6$  grader pr. 100 m stigning. Vi skulle dermed finne frem til programvare som kunne utføre følgende typer operasjoner:

- Uttrekk av varmesum og andre relevante data fra LMT, herunder stasjonenes koordinater
- Interpolasjon av varmesummen mellom stasjoner
- Høydekorreksjon via rasterberegning
- Presentasjon av resultatet på et kart på en stasjonsside

Det var dessuten et krav at komponentene skulle kunne inngå i en automatisering av den samlede prosessen.

### Hypotese for valg av tekniske komponenter

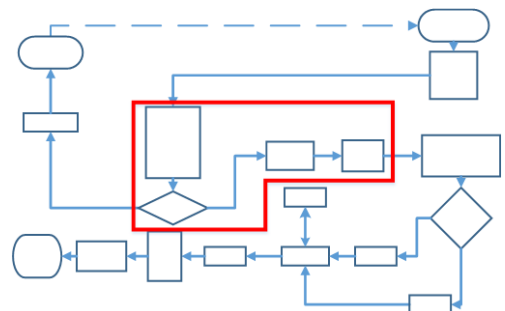
Hvis vi undersøker hvilke Open Source GIS-komponenter som kan løse disse oppgavene, kan vi sette opp en liste med de programmer og funksjoner vi vil teste ut.

Hvis vi deretter tester de ulike GIS-komponentene i detalj, kan vi finne fram til de som egner seg best til å løse de ulike delprosessene, som en del av en samlet automatisk prosess.

### 4.1 Datauttrekk fra LMT

I dette avsnittet vil vi beskrive hva vi har gjort for å trekke ut data fra LMTs database i et format som kan brukes i interpolasjonen, som er neste trinn i prosessen.

Databasen i LMT er en PostgreSQL-database, det vil si en database som i utgangspunktet ikke inneholder geometri, men bare numeriske koordinatangivelser i utvalgte felter. På bakgrunn av våre erfaringer med databaser antar vi likevel at vi gjennom en SQL-basert syntaks kan trekke ut



varmesumdata for LMTs værstasjoner i form av geografiske punktobjekter. Dette uttrekket skal kunne brukes direkte som input i en GIS-komponent som håndterer geoprosesseringen.

I vår første samtale med NIBIO gikk vi gjennom hvordan vi så for oss at den samlede prosessen kunne se ut på et logisk plan, altså hvilke operasjoner som skulle utføres og i hvilken rekkefølge (se bilag III). NIBIOs webutvikler tilbød seg å lage en webservice som kunne trekke ut data på grunnlag av en rekke inputparametere. Vi avtalte med NIBIO at vi skulle utarbeide en spesifisering for hvilke data som skulle trekkes ut, og i hvilke formater.

I denne prosessen har vi hatt et forløp som kan ligne et standard utviklingsforløp. Vi som geodataadministratorer hadde en bestillerrolle, vi spesifiserte krav og hadde ønsker til et produkt fra en produsent, vi testet og evaluerte resultatet og omsatte dette til nye krav og ønsker til produktet i en iterativ prosess. Derfor vil vi beskrive denne prosessen litt mer detaljert enn de øvrige tekniske delprosesser.

#### 4.1.1 Spesifisering av data til bruk i interpolasjonen

Vi utarbeidet en spesifisering vi mente kunne brukes til å lage webservicen. Utgangspunktet for arbeidet var vår forestilling om hvilke data vi ville få bruk for i de senere prosessene, og vårt kjennskap til LMT-data fra tidligere arbeid med systemet. På dette tidspunktet hadde vi ikke fått tilgang til LMTs testserver enda, og kjente ikke strukturen i databasen. Underveis i arbeidet med webservicen oppdaget vi en del nye forhold rundt databasen, som vi går gjennom i neste avsnitt.

Spesifikasjonen av hvilke datafelter og formater vi antok skulle brukes for å kunne utføre interpolasjonen og resten av prosessen fremgår av Tabell 3:

Navn (forslag)	Datatype	Beregning (konseptuell)	Beskrivelse
StationID	Integer		Stasjonens ID
StationName	Varchar		Stasjonens navn (lokalitet)
PeriodStart	Date		Startdato (brukervalgt)
PeriodEnd	Date		Sluttdato (brukervalgt)
BasisTemp	Float		Basistemperatur (brukervalgt)
NullCount	Int	Count * WHERE Date BETWEEN PeriodStart and PeriodEnd and value = NULL	Antall dager i perioden, hvor verdien er NULL/NoData.
DaysCount	Int	PeriodEnd – PeriodStart	Antall dager i perioden
StationZ	Float		Stasjonens høyde over havet i meter. Det er forutsatt at denne verdien finnes som felt i stasjonstabellen.

Varmesum	Float	(SUM(GnsDøgnTemp – BasisTemp) WHERE Date BETWEEN PeriodStart and PeriodEnd)	Her forutsetter vi at det finnes et felt, alias 'GnsDøgnTemp', som lagrer gjennomsnittsdøgntemperaturen. Hvis denne i stedet skal beregnes ut fra timeobservasjonene, skal 'formelen' endres.
VarmesumSL	Float	(SUM(GnsDøgnTemp – BasisTemp) WHERE Date BETWEEN PeriodStart and PeriodEnd)  + (StationZ /100 * 0,6 * (PeriodEnd – PeriodStart))	Beregnet varmesum for perioden, korrigert ned til havnivå (SeaLevel).
The_Geom	Geom		Stasjonens geometri (x,y)

Tabell 3: Spesifikasjon av datafelter i uttrekk fra LMT til bruk i interpolasjonen.

Vi antok at de grønne feltene allerede fantes i LMT.

De gule feltene er brukervalgte input fra LMT-siden.

De gråmarkerte felter er ikke egentlig nødvendige for beregningen, men vil kunne brukes i presentasjonen som *labels*, metadata, og liknende. Faktisk er bare feltene VarmesumSL og The\_Geom, altså varmesummen korrigert til havnivå og stasjonenes geometri helt nødvendige for interpolasjonen.

Dataene i de hvite feltene beregnes i webservicen, og skal inngå i blant annet interpolasjonen.

Under vil vi gå gjennom hvilke vurderinger som lå til grunn for de enkelte felter.

#### StationID:

Dette er en viktig nøkkel som brukes i mange sammenhenger på LMTs webside, for eksempel i følgende URL: [http://lmt.nibio.no/weatherstations/12/chart?log\\_interval=1](http://lmt.nibio.no/weatherstations/12/chart?log_interval=1), hvor 12 angir at det er snakk om stasjonen Balestrand. Vi mener at denne nøkkelen er fornuftig å ha med, i tilfelle den viser seg nødvendig, for eksempel til integrasjon med målte verdier for den enkelte værstasjon.

#### StationName:

Stasjonsnavnet er først og fremst tenkt brukt i visualiseringen, for eksempel som *label* på værstasjonspunktene.

#### PeriodStart, PeriodEnd og BasisTemp:

De brukervalgte parameterne inngår som grunnlag for selve datauttrekket, men vil også være bra å ha med i selve uttrekket, så de kan brukes som metadata og kontroll av om data er korrekt inntastet.

#### NullCount:

Ikke alle værstasjoner i LMTs stasjonsnett samler data hele året, og det er forskjell på når de

enkelte stasjoner settes i drift. Det kan dessuten være dager med utfall av datainnsamling eller overføring. På grunn av dette kan det forekomme dager med NULL-verdier i gjennomsnittsdøgnverdiene. Vi mener at før interpolasjonen utføres, bør det være en validering av stasjonsdata. Da kan vi eventuelt sortere bort stasjoner med manglende data og ha med en opptelling av antall NULL-verdier per stasjon for den valgte periode.

#### DaysCount:

Antall dager som skal inngå i uttrekket ved høydekorrigeringen (som utføres etter interpolasjonen). Denne verdien kan egentlig beregnes når som helst i prosessen, så lenge start- og slutt-tidspunkt er kjent.

#### StationZ:

Stasjonens høyde over havet, brukes til å omregne varmesummen til havnivå, samt kontroll og metadata.

#### Varmesum og VarmesumSL:

For hver stasjon beregnes en varmesum ut fra oppgitt basistemperatur og beregnede døgnverdier. Denne verdien inngår ikke i interpolasjonen. På grunn av høydeforskjellene mellom stasjonene er de ikke direkte sammenlignbare. Derfor beregnes varmesummen ved havnivå (SeaLevel: derav feltnavnet VarmesumSL) til bruk i interpolasjonen, men varmesumverdien kan likevel brukes som label på stasjonene og til å sammenligne målt og beregnet verdi.

#### The\_Geom:

The Geom er standard feltnavn for geometrikolonnen i PostGIS. Vi antok at stasjonstabellen ville inneholde geometri via PostGIS, og at geometriene derfor ville bære koordinatene (x,y) i uttrekket. Det viste seg at dette ikke var tilfelle, og det kommer vi tilbake til senere.

Vi vurderte i første omgang å bygge inn en "utvalg via radius"-algoritme i webservicen, fordi interpolasjoner ofte bruker en radius for å angi hvilke punkter som skal inngå i beregning av verdien for et enkelt punkt. Vi valgte likevel å ikke ta dette med, ettersom datamengden ikke er så stor (ca. 80 stasjoner), og det sannsynligvis enkelt kan gjøres som en del av interpolasjonen.

### 4.1.2 Webservicen – 1. iterasjon

På grunnlag av spesifikasjonen over, utarbeidet NIBIOs webutvikler Tor-Einar Skog et førsteutkast til en webservice. Denne ble utviklet i PHP, og hentet kun verdier for den valgte stasjonen. Formatet for URLen som skulle brukes til å kalle opp servicen så slik ut:

<http://lmttest.nibio.no/mtm/varmesumws/varmesum.php?weatherStationId=5&periodStart=2014-03-01&periodEnd=2014-06-15&basisTemp=10>. Her inngikk stasjons-ID samt parametrene PeriodStart, PeriodEnd og basisTemp i kallet. Webutvikleren hadde valgt json/text som outputformat, og resultatet så ut som vist i Tekstboks 1.

```
{{"periodStart":"2014-03-03","periodEnd":"2014-06-15","basisTemp":"0","weatherStationId":"5","weatherStationName":"\u00c5s","nullCount":"0","dayCount":"104","altitude":"94","latitude":"59.660468","longitude":"10.781989","linearHeatSum":"891.569573913043","linearHeatSumZAdjusted":"950.789573913044"}}
```

Tekstboks 1: Resultat fra webservicen, 1. iterasjon

I mellomtiden hadde vi fått tilgang til LMTs testserver og PostGres-databasen, og hadde derfor også mulighet til å undersøke kildekoden til webservicen. Kildekoden er på NIBIOs testside <sup>6</sup>.

```
<?php
/*
 * @copy; 2015 NIBIO
 * @author Tor-Einar Skog tor-einar.skog@nibio.no
 */
include_once("db.php");
$retVal=$_GET;
$varmesum =
getVarmesum($retVal["weatherStationId"],$retVal["periodStart"],$retVal["periodEnd"],$retVal["basisTemp"]);

$retVal["weatherStationName"] = iconv("ISO-8859-1", "UTF-8", $varmesum["name"]);
$retVal["nullCount"] = $varmesum["null_count"];
$retVal["dayCount"] = $varmesum["day_count"];
$retVal["altitude"] = $varmesum["altitude"];
$retVal["latitude"] = $varmesum["latitude"];
$retVal["longitude"] = $varmesum["longitude"];
$retVal["linearHeatSum"] = $varmesum["linear_heat_sum"];
$retVal["linearHeatSumZAdjusted"] = $varmesum["linear_heat_sum_z_adjusted"];

header('Content-Type: application/json; charset=utf-8');
//print_r( $retVal );
echo json_encode($retVal);
```

Tekstboks 2: Kildekoden i 'varmesum.php'

Varmesum.php er den filen som eksekveres via URL-kallet. Dette gjøres ved at adressen til filen angis i URLen. I varmesum.php er filen db.php inkludert, slik at denne kjøres under eksekvering av varmesum.php (se lyseblå markering i Tekstboks 2). Varmesum.php inneholder dessuten funksjoner for å lese inputparametrene som er angitt i URL-kallet, for eksempel

<sup>6</sup> <http://lmttest.nibio.no/mtm/VedleggTilRapport/ScriptsTilDataudtrek.pdf>



'weatherStationId=5&periodStart=2014-03-01&periodEnd=2014-06-15&basisTemp=10' og gjøre disse parametrene til variabler som skal brukes i uttreksfunksjonen (se gul markering i Tekstboks 2. Endelig inneholder varmesum.php funksjoner for å strukturere uttreksresultatet og skrive det som output i JSON-format (markert med henholdsvis grønn og lilla i Tekstboks 2.

På samme måte har vi forsøkt å tolke hva som skjer i filen db.php. Det meste av innholdet i denne kan vi gjenkjenne som en SQL-forespørsel med en struktur som svarer til det vi kjenner fra diverse desktop-GIS-programmer:

```
SELECT [COLUMNS, DATA]
FROM [TABLES]
WHERE [CONDITIONS]
GROUP BY, SORT BY [COLUMNS]
```

I SELECT-delen (markert med lyseblått i Tekstboks 3) hentes først feltene weatherstation\_id, name, latitude, longitude og altitude fra en tabell, som her kalles 'w'. I FROM-delen (markert med gult) er det angitt at tabellen 'w' tilsvarer tabellen 'weatherstation' og skjemaet 'public'. I forhold til våre spesifikasjoner er det altså her at de eksisterende feltene StationID, StationName og StationZ fra tabellen trekkes ut. I tillegg hentes verdiene fra feltene 'latitude' og 'longitude' - stasjonenes x- og y-koordinater. Posisjonen hentes ut som desimaltall i et koordinatsett og ikke som et geografisk objekt, slik vi hadde håpet. Dette måtte vi finne en løsning på, ettersom vi trengte geografiske objekter med attributtdata for å kunne utføre interpolasjonen.

Deretter beregnes verdiene linear\_heat\_sum, linear\_heat\_sum\_z\_adjusted, null\_count og day\_count, som svarer til Varmesum, Varmesum\_SL, NullCount og DaysCount i spesifikasjonen. Vi ble her oppmerksom på at det bare ble beregnet varmesum for en gitt dag hvis døgnmiddeltemperaturen er større enn basistemperaturen. Dette kan leses i uttrykket: CASE WHEN ov.value\_controlled - \$1 > 0.0 THEN ov.value\_controlled - \$1, hvor \$1 representerer basistemperaturen. Dette hadde vi ikke vært klar over tidligere, og vi hadde derfor lurt på om det kunne oppstå tilfeller hvor varmesummen ville bli negativ. Dette kan altså ikke skje. Det er et godt eksempel på at de forskjellige parter i et utviklingsprosjekt har forskjellige bilder av virkeligheten som er det nødvendig å få samkjørt. I dette tilfellet en selvfølgelighet for NIBIO som har arbeidet med disse verdiene og definisjonene i årevis, og en antagelse fra den annen parts side (MTM-gruppen) som ingen har tenkt på å undersøke nærmere.

```

<?php
include_once '.././conf.php';
function getVarmesum($weatherStationId,$periodStart,$periodEnd,$basisTemp)
{
    $heightCorrFactor = 0.6;
    $conn = getConn();
    $query = "SELECT ";
    " w.weatherstation_id, w.name, w.latitude, w.longitude, w.altitude, ";
    " SUM(CASE WHEN ov.value_controlled - $1 > 0.0 THEN ov.value_controlled - $1 ELSE 0 END) AS
linear_heat_sum, ";
    " SUM(CASE WHEN ov.value_controlled - $1 + (w.altitude * $2/100) > 0.0 THEN ov.value_controlled -
$1 + (w.altitude * $2/100) ELSE 0 END) AS linear_heat_sum_z_adjusted, ";
    " SUM(CASE WHEN ov.value_controlled IS NULL THEN 1 ELSE 0 END) as null_count, ";
    " EXTRACT(DOY FROM TO_DATE($4,'YYYY-MM-DD')) - EXTRACT(DOY FROM TO_DATE($3,'YYYY-MM-
DD')) AS day_count ";
    "FROM public.weatherstation w, lmt.observation_view ov ";
    "WHERE w.weatherstation_id= $5";
    "AND ov.weatherstation_id=w.weatherstation_id ";
    "AND ov.element_measurement_type_id = 1 ";
    "AND loginterval_id=3 ";
    "AND time_measured BETWEEN TO_DATE($3,'YYYY-MM-DD') AND TO_DATE($4,'YYYY-MM-DD') ";
    "GROUP BY w.weatherstation_id, w.name, w.latitude, w.longitude, w.altitude ";
;
//echo $query;
$result =
pg_query_params($query,Array($basisTemp,$heightCorrFactor,$periodStart,$periodEnd,$weatherStati
onId)) or die('Query failed: ' . pg_last_error());
$retVal = pg_fetch_array($result, null, PGSQL_ASSOC);
closeConn($conn);
return $retVal; }

```

Tekstboks 3: Kildekoden i filen 'db.php'. Variablenavnene \$1-\$5 refererer til parametrene markert med lilla i viste rekkefølge.

I WHERE-delen av forespørselen (markert med grønt i Tekstboks 3) settes noen betingelser for uttrekket, blant annet `ov.element_measurement_type_id = 1`, som angir at det bare skal hentes målinger av temperaturen, og `loginterval_id=3`, som angir at det er døgnverdier som skal hentes. Det er også her det skapes en kobling mellom de to berørte tabeller, nemlig stasjonstabellen (public.weatherstation) og observasjonstabellen (lmt.observation\_view), med weatherstation\_id som koblingsnøkkel.

Til slutt returneres resultatet av uttrekket på en form som kan hentes fra scriptet varmesum.php, og presenteres som det endelige resultatet (markert med grått).

Som nevnt over, kunne denne webservicen bare trekke ut varmesummen og de øvrige data for den ene stasjonen som var angitt i URL-kallet. For å kunne utføre en interpolasjon på datasettet måtte vi

likevel samle inn data for alle relevante værstasjoner i ett datasett, og vi antok at det ville forenkle prosessen dersom vi slapp å sende 80 separate kall til databasen.

I databasen kunne vi dessuten se at tabellen inneholdt stasjoner både i Sverige og Bosnia og Herzegovina. Disse var vi ikke interessert i, og spurte om det var mulig å filtrere ut disse fra webservicen, for eksempel ved å bruke feltet "show\_on\_website", hvor vi kunne se at bare norske, aktive stasjoner hadde verdien 'TRUE'.

På bakgrunn av denne tilbakemeldingen tilpasset NIBIOs webutvikler servicen, og vi fikk en ny versjon.

#### 4.1.3 Webservicen – 2. iterasjon

Den nye versjonen av webservicen kunne hente ut alle de ønskede verdier for alle stasjoner på en gang, og inputparameteren 'weatherstationID' var fjernet fra URL-kallet:

<http://lmttest.nibio.no/mtm/varmesumws/varmesum.php?periodStart=2014-03-03&periodEnd=2014-06-15&basisTemp=0>

```
[{"periodStart":"2014-03-03","periodEnd":"2014-06-15","basisTemp":"0","weatherStationId":"38","weatherStationName":"Ramnes","nullCount":"0","dayCount":"104","altitude":"39","latitude":"59.38081","longitude":"10.23923","linearHeatSum":"946.840810606061","linearHeatSumZAdjusted":"971.410810606061"}, {"periodStart":"2014-03-03","periodEnd":"2014-06-15","basisTemp":"0","weatherStationId":"22","weatherStationName":"Hjelmeland","nullCount":"0","dayCount":"104","altitude":"43","latitude":"59.22995","longitude":"6.14992","linearHeatSum":"964.460648715415","linearHeatSumZAdjusted":"991.550648715415"}, {"periodStart":"2014-03-03","periodEnd":"2014-06-15","basisTemp":"0","weatherStationId":"50","weatherStationName":"Tj\u00f8lling","nullCount":"0","dayCount":"104","altitude":"19","latitude":"59.04641","longitude":"10.12513","linearHeatSum":"985.58675","linearHeatSumZAdjusted":"997.55675"}, ...
```

*Tekstboks 4 Resultatet av webservicekall, versjon 2.*

Resultatet (utdrag som viser de tre første værstasjonene) er vist i Tekstboks 4.

Vi skal ikke gjennomgå alle endringene i php-filene i detalj, men de besto blant annet i å tilføye FOREACH... og WHILE... løkker, som leste alle stasjonsrecorder, og skrive de ønskede stasjoner til et *array*, det vil si en ordnet liste som utgjør et samlet resultat. I tillegg ble det tilføyd en betingelse om at start- og sluttdato skulle ligge i samme kalenderår. Det vil som oftest ikke gi noen mening å be om varmesum utover vekstsesongen, fordi mange av stasjonene er tatt ut av drift (NIBIO, 2015b), og det er lite landbruksaktivitet i vintersesongen. Denne begrensningen finnes likevel ikke i den nåværende varmesumberegningen i LMTs stasjonsside.

Outputformatet var fremdeles JSON, og resultatet ble lagret til en fil som deretter kunne åpnes i for eksempel QGIS som en tabell med attributtdata. Det var fremdeles ingen objektgeometri i dette formatet, og vi ville derfor ha måttet opprette geometrien på grunnlag av de angitte koordinatsett for hver målestasjon. Vi bestemte derfor at vi ville forsøke å få tilpasset webservice-scriptet slik at output-formatet ble GeoJSON, som er et JSON-format som også inneholder geometri.

#### 4.1.4 Webservicen – 3. iterasjon

JSON er et format hvor data er oppdelt i objekter. I GeoJSON finnes det i tillegg objekttyper vi kjenner fra GIS-bransjen, for eksempel punkt, linje, polygon og multiobjekt. Hvis det er flere geografiske objekter i en GeoJSON-fil utgjør de til sammen et objekt av typen *FeatureCollection*, som tilsvarer det vi kaller et lag eller en *feature class* i desktop-GIS-programmer. I et objekt av denne typen kan det være et ubegrenset antall såkalte «medlemmer», som hver kan være ett av ovenstående type geografiske objekter og ha et antall koordinatpar som egenskaper, alt etter objekttype. I tillegg kan hvert objekt ha andre egenskaper, som man i databaseterminologi kaller attributtdata. Data er organisert i *key-value pairs*, som består av feltnavn:verdi-par, for eksempel: «weather StationName»:»Hjelmeland» (GeoJSON Format Specification, The, 2015).

Et eksempel på oppbygningen av en GeoJSON-fil med hierarkiet mellom *FeatureCollection* og *feature* er vist i Tekstboks 5.

```
{ "type": "FeatureCollection", "features": [ { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ 10.23923, 59.38081 ] }, ...
```

*Tekstboks 5: Eksempel på strukturen i en FeatureCollection i GeoJSON-format*

Opgaven besto altså i å få endret de to php-scriptene så værstasjonene ble samlet i en *FeatureCollection*, og å få lagt verdiene fra *latitude* og *longitude* fra stasjonstabellen inn som koordinater til punktobjekter som kunne tilknyttes de øvrige attributtdata i den enkelte værstasjon. Ved søk på nettet, for eksempel med følgende søkeord: «php output geojson» fant vi flere interessante innlegg i diskusjonsfora som ga oss inspirasjon til selv å prøve å skrive disse endringene <sup>7</sup> (McBride, 2015).

Vi klarte å eksportere et GeoJSON-format med stasjonskoordinater formatert som geometriobjekter, men ikke å få dem til å henge sammen med de øvrige attributtdata for stasjonene, så vi måtte ha hjelp fra NIBIOs webutvikler til dette. En prøveinnlesing i QGIS viste også at datatypene måtte konverteres før outputet kunne brukes direkte. Dette er fordi alle data i utgangspunktet kom ut formatert som tekst, og derfor ikke kunne brukes som tallverdier i for eksempel interpolasjonen. Omformateringen kunne gjøres ved å tilføye *int* (heltall) eller *double* (desimaltall) foran de enkelte felter i outputformateringen.

Dessuten har de uttrukne dataene koordinater i lengde/breddegrader (EPSG:4326), mens vi gjerne vil ende opp med et resultat i UTM sone 33 (for eksempel EPSG:32633). Dette vil imidlertid kreve at det tilføres koordinater i denne projeksjonen for hver stasjon, med mindre det kan løses på annet vis. Vi noterer dette som et forhold som NIBIO bør være oppmerksom på.<sup>8</sup>

I samarbeid klarte vi på denne måten å få utarbeidet en webservice som ga et output i GeoJSON, og som kunne leses direkte inn i QGIS og brukes i den påfølgende geoprosesseringen. Resultatscriptene er lagt på NIBIOs testside.<sup>9</sup>

<sup>7</sup> <http://stackoverflow.com/questions/17775627/creating-a-geojson-in-php-from-mysql-to-use-with-mapbox-javascript-api>

<sup>8</sup> <http://lmttest.nibio.no/mtm/VedleggTilRapport/TilpassPG-tabell.pdf>

<sup>9</sup> <http://lmttest.nibio.no/mtm/VedleggTilRapport/ScriptsTilDataudtrek.pdf>

### 4.1.5 Oppsummering

Vi erfarte i løpet av prosessen at vi endte opp med et litt annet produkt enn det vi bestilte. Dette kom antagelig av vår (bestillerens) og webutviklerens (produsentens) ulike forståelseshorisonter. Vi hadde ikke god nok kunnskap om datastrukturen og logikken i LMT til å kunne spesifisere uttrekket presist nok, og utvikleren hadde ikke nok kunnskap om våre behov til å vite eksakt hva som skulle leveres. Ved å samtale om de konkrete utfordringene gjennom prosessen ble vi likevel klokere på hverandres områder, og endte opp med et brukbart produkt. En annen gang vil vi skrive oss bak øret at det er en god ide å bringe utviklerkompetansen inn allerede ved utarbeidelse av spesifikasjonene.

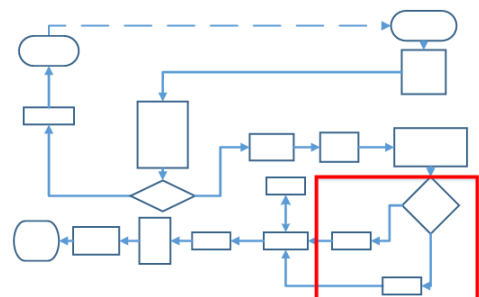
Samtidig var det et godt eksempel på hvordan de forskjellige kunnskapsområder kan utfylle hverandre – utvikleren kunne med sin ekspertise gjøre de nødvendige tilpasninger i scriptet, og vi kunne med vår kunnskap om geoinformasjon og GIS teste om det uttrukne data kunne brukes i den videre prosessen.

Det er nok tvilsomt om webservicen har funnet sin endelige form. Etter hvert som prosjektet eventuelt videreutvikles vil det antagelig oppstå flere ønsker til uttrekk eller formatering av data, som må spesifiseres og utarbeides i kommende iterasjoner.

## 4.2 Interpolasjon

I dette underkapittelet vil vi beskrive hvilke overveielser vi har gjort når det gjelder interpolasjon av varmesum mellom værstasjonene, deriblant både valg av beregningsmetode, håndtering av eventuelle feil og avvik i dataene, og test og valg av verktøy til oppgaven.

Når data er trukket ut av databasen er neste trinn i prosessen (jf. Flytdiagram i bilag III) å interpolere varmesummen mellom værstasjonene som nå foreligger som spredte geografiske punkter (input), så det dannes et output i form av et rasterbilde som dekker et område rundt den valgte værstasjonen.



Vi skulle velge en interpolasjonsmetode og et verktøy som kunne utføre beregningen. Målet var å lage en så god og troverdig interpolasjon som mulig, men vi ønsket først og fremst å få etablert en sammenhengende løsning på grunnlag av de relevante komponentene, så vi ville derfor ikke bruke for mange krefter på å finjustere interpolasjonsalgoritmen.

### 4.2.1 Valg av beregningsmetode og verktøy

I designfasen har vi hittil fattet følgende beslutninger vedrørende interpolasjonen:

- Vi har valgt at interpolasjonsresultatet skal vises med sentrum i den valgte værstasjon.
- Vi vil bare bruke LMTs egen stasjoner til interpolasjon; datauttrekk fra klima.no viste seg å være vanskelig tilgjengelig.
- Interpolasjonen skal vise varmesumverdier i absolutte tall, det vil si at det i første omgang ikke blir muligheter for å beregne verdier i forhold til tidligere år.
- Interpolasjonen skal bare vises i en gitt avstand fra målestasjonen.
- Hvis det ikke finnes nok stasjoner innen den valgte søkeradius skal den målte varmesummen bare høydekorrigeres, for eksempel ved at det produseres et konstant raster med den målte

varmesum korrigeret til havnivå, slik at høydekorreksjonen kan foretas på samme måte som for interpolerte resultater.

- Hvis dataene for en værstasjon inneholder NULL-verdier innenfor den valgte perioden, skal stasjonen filtreres ut før interpolasjonen, eller brukeren skal få en melding om hvor mange NULL-verdier som inngår i resultatet.

På grunnlag av våre erfaringer så langt og ovenstående beslutninger utarbeidet vi en kravliste til interpolasjonen.

Komponenten skal kunne:

- A. Interpolere attributtdata (varmesummen) på spredte punktobjekter med et rasterlag som output,
- B. Finne punktobjekter å interpolere mot via en søkeradius,
- C. Velge en minimumsgrense for antall punkter å interpolere mellom,
- D. Avgrense det resulterende rasterlaget, for eksempel via en radius eller en *bounding box*,
- E. Inngå i en automatisert prosessflyt, for eksempel via et script.

Vi besluttet å finne frem til og teste en rekke GIS-komponenter som kan utføre interpolasjonen, for å kunne sammenligne de enkelte komponenters funksjonalitet med de oppgaver vi ønsket at komponenten skulle løse. På denne måten ville vi komme frem til en anbefaling av den eller de beste komponentene til oppgaven.

På tredje semester fant vi anbefalinger i litteraturen om å bruke residual kriging til interpolasjon av temperaturvariabler (Ustrnul & Czekierda, 2005). Dette var utgangspunktet for våre søk etter programvare.

Som nevnt i kapittel 3.1.5 fikk vi noen råd og tips fra NIBIOs webutviklingsmiljø angående mulige kandidater og programplattformer vi burde se nærmere på.

- GRASS
- SAGA
- GSTAT
- HPGL
- R-project

(Nystuen, pers. medd.)

Vi valgte i første omgang å undersøke GRASS, SAGA og R-project via søk på nettet. Under denne gjennomgangen fant vi ut at QGIS har integrasjon mot funksjonsbibliotekene til både GRASS, SAGA og GDALOG, og ettersom QGIS dermed kunne brukes som testmiljø for utprøving av en rekke forskjellige algoritmer, valgte vi å fortsette med disse tre funksjonsbibliotekene.

## **Kriging**

Vi fant bare SAGA-algortimene Ordinary Kriging og Universal Kriging tilgjengelige via QGIS. I forsøkene på å teste disse støtte vi på begreper som *variogram-modell*, *nugget*, *sill* og *range*. Gjennom litteraturstudier fant vi ut at kriging oftest krever en innledende utarbeidelse av et semivariogram til å fastsette disse parametrene (Hengl, 2009).

Vi vurderte at dette ville bli for komplisert å sette opp i en automatisk prosess, og at bruk av kriging ville kreve at vi brukte mange ressurser til å sette oss inn i matematikken og statistikken omkring kriging. Vi valgte å fokusere på å få satt opp en sammenhengende, automatisk løsning fremfor å bruke tid på å finne en optimal algoritme til interpolasjonen, og lette derfor etter mer tilgjengelige alternativer.

### Inverse Distance Weighting

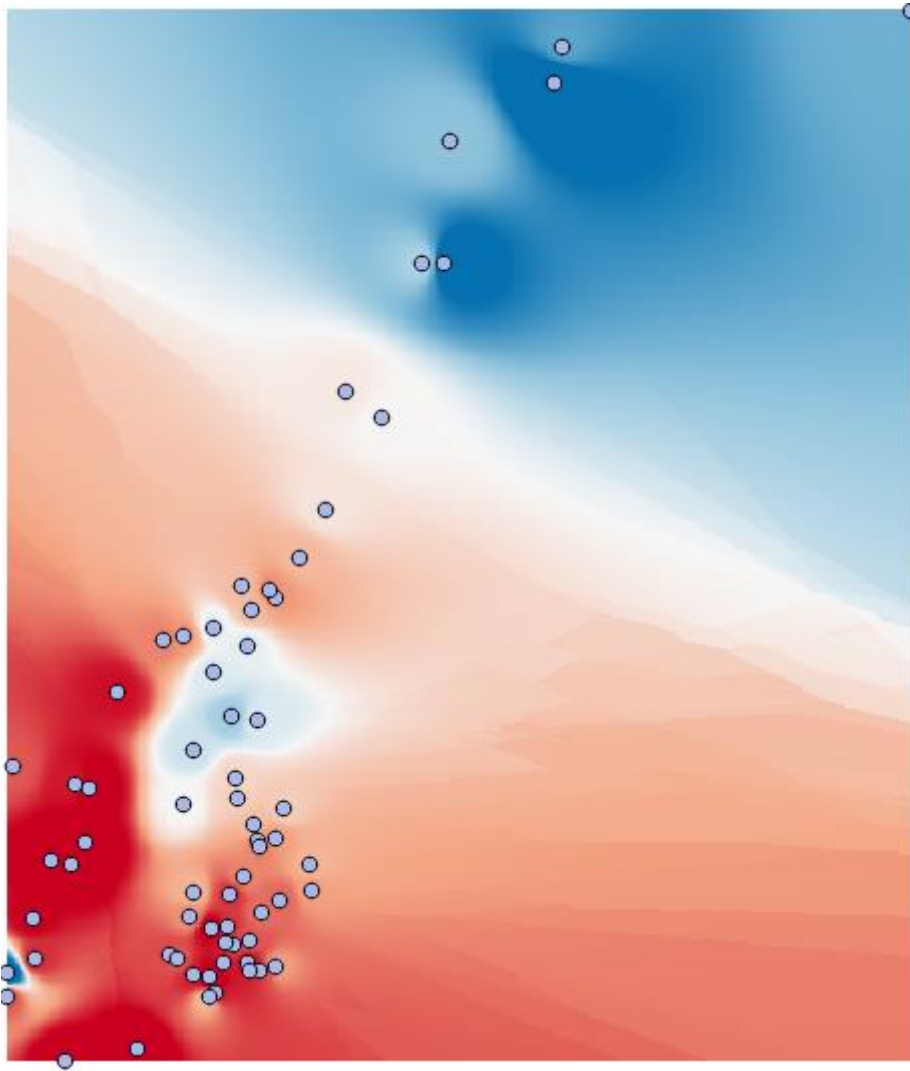
Hengl (2009) nevner at kriging er en avansert utgave av interpolasjonsmetoden Inverse Distance Weighting (IDW), og denne metoden har vi også sett nevnt andre steder som alternativ for interpolasjon av temperatur (Ninyerola, 2007). Uttrykket betyr «vekting etter invers avstand», og prinsippet bak IDW er at målepunkter som ligger nær et gitt punkt får større innflytelse på den interpolerte verdi i punktet enn målepunkter lenger bort. Prinsippet går igjen i «Toblers 1. lov om geografi» (Tobler, 1970). Det er den lineære avstand til punktet som inngår i vektingen, og avstanden kan inngå med større eller mindre vekt ( $\beta$ ). Til forskjell fra kriging fastlegges ikke  $\beta$  ut fra en analyse av de innbyrdes forskjeller i data (semivariogrammet), men fastsettes ut fra brukerens antagelser om data, anbefalinger, erfaringer med mer. Kunsten er med andre ord å fastsette  $\beta$  så den avspeiler den aktuelle, romlige autokorrelasjonen i data mellom punktene – i denne sammenheng forstått som den beregnede varmesum for hver av LMTs værstasjoner.

Vi fant IDW tilgjengelig via GRASS-algoritmen «v.surf.idw», som også er tilgjengelig via QGIS. I denne funksjonen kan man angi parametere som vist i Tabell 4.

Parameter	Forklaring
Number of interpolation points	Øvre grense for antall punkter som kan inngå i beregning av verdien for et gitt punkt.
Power parameter	$\beta$ -verdien, dvs. vektingsfaktoren for avstanden. Standardverdien er 2. Jo større verdi, jo større vekt fra målinger tett på punktet.
GRASS region extent	Bounding box for resultatet, altså rasterlagets utstrekning, men også for hvilke punkter som kan inngå i interpolasjonen. Det vil si at punkter som ligger utenfor BBOX tas ikke med.
GRASS region cellsize	Cellestørrelse i rasterlaget.

Tabell 4: Inputparametre i GRASS-algoritmen v.surf.idw

Test av algoritmen med et sett uttrukne data i GeoJSON-format ga et resultat som vist i Figur 12.



*Figur 12 Resultat av interpolasjon av varmesum fra GeoJSON datasett med alle LMTs-værstasjoner.*

Resultatet av våre forsøk med algoritmen har vist at den oppfylder kravene A, B og D som er nevnt ovenfor. Det kan ikke angis et minimumstall av punkter eller målinger for interpolasjonen, som anført i krav C. Til gjengjeld blir resultatet et rasterlag med værstasjonens målte verdi i hele laget hvis det ikke finnes noen nabostasjoner å interpolere med. Dette er et ønskelig resultat, siden man kan bruke dette laget som grunnlag for høydekorreksjonen. Krav C kan derfor fravikes.

Krav E handler om at algoritmen skal kunne inngå i en automatisert prosess. Bare i QGIS alene finnes det flere forskjellige måter å bruke IDW på hvor det kan angis forskjellige parametere. Vi har valgt å se bort fra en annen funksjon som er innebygd i QGIS, og som også bruker IDW til beregning, fordi QGIS-funksjoner ikke kan kalles via et API. GRASS har derimot mulighet for at funksjonene kan kalles via Python, og vi håper at vi ved bruk av disse kan automatisere interpoleringen. Dette behandles videre i kapittel 4.5 om automatisering.

Vi har dessuten undersøkt om interpolasjon fantes tilgjengelig som WPS-prosess, siden vi tenkte dette kanskje ville lette den automatiske kjøringen, men vi har ikke funnet en passende algoritme tilgjengelig på nettet.



#### 4.2.2 Stasjoner med NULL-verdier

Som nevnt over skal vi være oppmerksomme på stasjoner hvor det finnes NULL-verdier i døgnmiddeltemperaturen innenfor den valgte perioden, siden de ellers bare ville inngå i varmesumberegningen som verdien 0, noe som vil gi et galt resultat. Det enkleste vil være å frasortere stasjoner hvor NullCount-verdien er større enn 0, altså hvor det er minst én dag med NULL i døgnmiddeltemperaturen. Risikoen med dette er at man dermed reduserer datagrunnlaget (antall stasjoner) i interpolasjonen, og dermed kan få et mer upresist resultat for områder der disse stasjonene kunne ha bidratt med data. Det er altså et balanseforhold mellom datamengde og datakvalitet.

Vi har diskutert om man kan finne et passende maksimum for hvilket antall eller prosentandel av døgnverdiene som må være NULL før en stasjon velges fra interpolasjonen, men vurderer at dette krever en nærmere analyse av omfanget av NULL-verdier i data, og av i hvilken grad resultatet påvirkes av forekomsten av NULL-verdier. En slik analyse har vi av tidshensyn valgt å ikke lage i dette prosjektet. Det er likevel noe vi vil anbefale NIBIO å se nærmere på – eventuelt også for å informere om kvaliteten på den eksisterende varmesumberegningen.

Rent teknisk er det ikke noe større problem å frasortere stasjoner med en viss mengde NULL-verdier, siden vi får trukket ut denne verdien for hver stasjon gjennom webservicen, og filtreringen kan – når man har funnet en passende grense – enten bygges inn i webservicen, eller foretas ved hjelp av en SQL-basert syntaks før datasettet inngår i interpolasjonen.

#### 4.2.3 Oppsummering

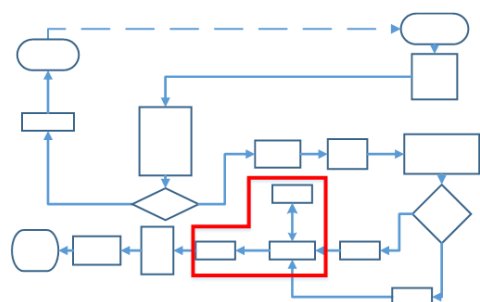
Samlet sett er det vår vurdering at GRASS' IDW-funksjon er en egnet metode til å interpolere varmesummen i vår prototype. I den endelige løsningen vil vi likevel fortsatt anbefale å bruke kriging, men det vil kreve bistand fra utviklere med erfaring i kriging og eventuelt meteorologi for å få angitt de rette parametere og dermed oppnå et troverdig resultat.

Resultatet av den prosessen vi har beskrevet ovenfor er at vi ikke endte med det redskap vi begynte med å lete etter. Vi har blitt klokere når det gjelder hvilke verktøy som finnes, og har lært at verktøyene krever forskjellige inputparametere, noe som påvirker definisjonen av den oppgaven vi skal løse ved hjelp av verktøyet. Og i tillegg har vi måttet innse at noen verktøy krever mer enn bare generell GIS-kunnskap for at man skal kunne bruke dem riktig.

Vi anbefaler NIBIO å foreta en nærmere analyse av forekomsten av NULL-verdier i døgnmiddeltemperaturen for å kunne fastsette en grense, så stasjoner som ville gi for store feil i resultatet for en valgt periode kan frasorteres.

#### 4.3 Høydekorreksjon

Resultatet av interpolasjonen er et rasterbilde med en celledørrelse på 1000 x 1000 meter, og som er avgrenset av en *bounding box* på en valgt avstand omkring den valgte stasjonen. Som nevnt i avsnittet om interpolasjon utføres denne på verdier av varmesummen som er korrigeret for



høyde over havet, så verdiene fra de forskjellige værstasjonene blir sammenlignbare. Dette tilsvarer at målingene er foretatt ved havnivå.

Neste trinn i prosessen var derfor å korrigere dette resultatet så det viser en verdi som tilsvarer målinger på det faktiske terrengnivå. For å kunne gjøre dette trengte vi et datasett som viser terrenghøydene i Norge, og vi måtte finne en metode og et verktøy som kunne foreta beregningen.

#### 4.3.1 Korreksjonsalgoritme

I datauttrekket utføres korreksjonen av varmesummen til havnivå for hver stasjon av denne formelen:

$$\text{Varmesum ved havnivå} = \text{Målt varmesum} + (\text{terrenghøyde}/100 * 0,6 \text{ °C} * \text{antall dager i perioden})$$

Uttrykket i parentesene kan kalles korreksjonsbidraget som altså legges til den målte varmesummen. For å korrigere resultatet tilbake til aktuell terrenghøyde etter at interpolasjonen er utført, må korreksjonsbidraget på samme måte trekkes fra varmesummen på havnivå:

$$\text{Målt varmesum} = \text{Varmesum ved havnivå} - (\text{terrenghøyde}/100 * 0,6 \text{ °C} * \text{antall dager i perioden})$$

I tredje semesters prosjekt kom vi frem til at et rasterkalkulasjonsverktøy som for eksempel MapAlgebra i ArcGIS kunne brukes til dette ved å utføre matematiske funksjoner på celleverdiene i to overlappende rasterlag. Det ene laget er resultatet av interpolasjonen, og det andre er en Digital Terreng Modell (DTM) eller deler av en slik. Vi undersøkte derfor hvilke tilsvarende funksjoner som finnes i Open Source GIS-verktøyene, og fant fram til tre algoritmer som så ut til å kunne løse oppgaven:

- r.mapcalculator fra GRASS,
- Raster Calculator fra GDALOGR, samt
- Raster Calculator fra SAGA.

I de to første algoritmene har man mulighet til å spesifisere cellestørrelse, noe som er nødvendig for å sikre en cellestørrelse på 1000 x 1000 meter i det resulterende rasterlag. I GRASS-algoritmen kan man dessuten definere en *bounding box*, det vil si resultatrasterets utstrekning, og vi har derfor foretrukket denne algoritmen i arbeidet med prototypen, selv om også RasterCalculator fra GDALOGR har vært brukt.

Et eksempel på en utfylt dialogboks med relevante inputparametere kan ses i Figur 24. Dialogboksen viser bruk av funksjonen i QGIS-grensesnittet. Etter planen skulle funksjonen også brukes i en automatisk kjøring av hele prosessen, men på grunn av problemer med det grunnleggende oppsettet av denne automatiseringen (som er beskrevet i kapittel 4.5) rakk vi aldri å arbeide med høydekorreksjonen på annen måte enn ved rent manuelt oppsett i QGIS.

#### 4.3.2 Terrenghøyder

Vi har også undersøkt hvilke eksisterende høydedata vi kan bruke til høydekorreksjonen. Det er utarbeidet en DTM for hele Norge som er basert på digitale, topografiske kart med høydekurver, som har en oppløsning på 10 x 10 meter. Det finnes avledete versjoner av DTMen med lavere oppløsning, for eksempel en utgave med oppløsning på 50 x 50 meter, som vi har brukt som input til høydekorreksjonen (Kartverket, 2015).

Det er lettest å utføre en rasterkalkulasjon på to eller flere rasterlag hvis rasterlagene har samme utstrekning (*bounding box*) og cellestørrelse. Det kan likevel tenkes at det er mulig å kjøre rasterkalkulasjon på lag med forskjellig utstrekning og cellestørrelse. Derfor har vi testet to forskjellige metoder for å finne ut hvordan vi best kunne tilrettelegge høydedataene for en prosess, hvor valg av inputdata skal kunne skje automatisk:

- Data fra en DTM klippes i et kvadratisk utsnitt rundt den valgte værstasjonen, og *resamples* til en cellestørrelse på 1000x1000 meter. Vi vet at dette er mulig å gjøre, og at det trolig gir god ytelse i beregningen, ettersom DTMen er generalisert og har samme oppløsning som interpolasjonsresultatet. Til gjengjeld krever det at det enten skal klippes i DTMen underveis i prosessen, eller at det ligger predefinerte utklipp av DTMen for hver av LMTs værstasjoner.
- Bruk av en samlet DTM lagret som rasterkatalog (.vrt) bestående av DTM-tiles på 100x100 km med den originale oppløsningen på 50 x 50 meter. Hvis denne kan brukes direkte i høydekorreksjonen, sparer man en prosess med klipping og *resampling*, og det vil være en fordel i forhold til automatiseringen. På denne måten kan høydekorreksjonen alltid bruke den samme DTM-filen.

Testen viste at begge metoder er brukbare til høydekorreksjon, og det er ikke noen nevneverdig forskjell i ytelsen. Det er derimot litt forskjell på resultatet, hvilket antakelig skyldes cellestørrelsen i de to utgavene av DTMen.

De største forskjellene merkes på steder med markante høydeforskjeller, og det er nettopp her det er stor forskjell på om man bruker en terrenghøyde som er et gjennomsnitt av høydene innenfor en celle på 50 x 50 meter eller 1000 x 1000 meter. Forskjellene ser likevel ut til utliknes over et større areal. På bakgrunn av analysen vil vi anbefale å bruke rasterkatalogen som datagrunnlag, særlig med tanke på automatiseringen av prosessen. Filtypen det er snakk om (.vrt) kan også leses av både GRASS og GDAL utenfor QGIS, hvilket er en fordel hvis et script skal kunne kalle GRASS-funksjonen (osgeo.org, 2015) .

### 4.3.3 Feilkilder

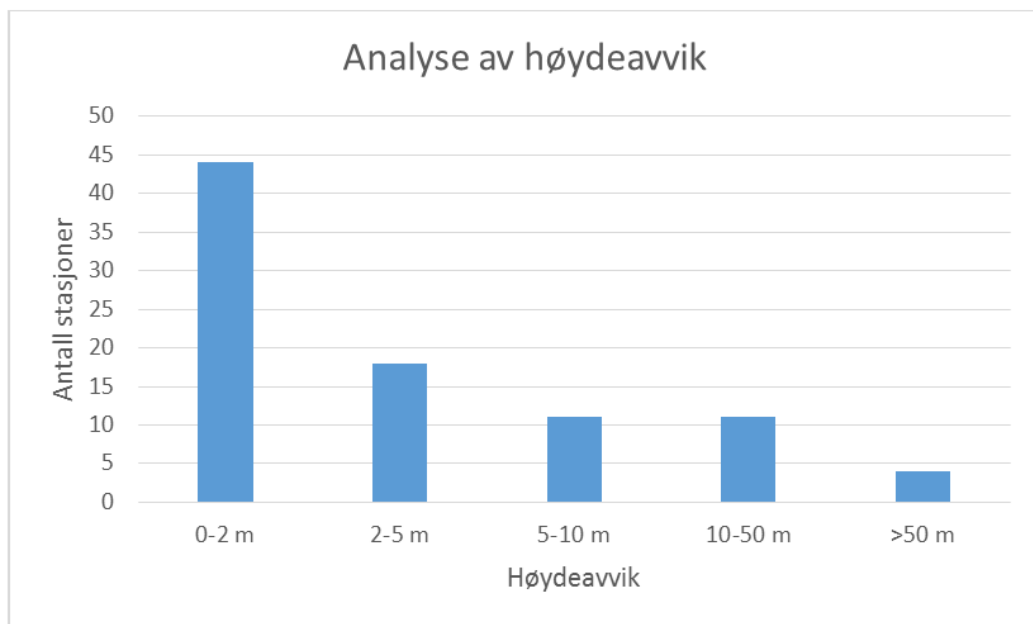
I arbeidet med høydekorreksjonen har vi blitt oppmerksomme på flere potensielle feilkilder. I de neste avsnittene vil vi gjennomgå disse, og beskrive hvilken innflytelse de potensielt kan ha på resultatet. Vi diskuterer også hvordan feilkildenes påvirkning av resultatet kan minimeres.

#### Stasjonshøyder

Den oppgitte terrengkote for værstasjonene i LMT, som hittil bare har vært brukt som orientering og metadata om stasjonen, blir med innføring av høydekorreksjonen plutselig en viktig faktor i beregningen. Dette drøftet vi allerede tidlig i prosessen med NIBIO, hvor det kom frem at kildene til høydeangivelsen var ulike og antagelig av varierende kvalitet. Vi avtalte at vi som en del av prosjektet skulle prøve å verifisere de angitte stasjonshøyder gjennom bruk av en DTM. Analyseresultatene på stasjonsnivå kan ses i på NIBIOs testside.<sup>10</sup> Et sammendrag av analysen vises i Figur 13.

---

<sup>10</sup> <http://lmttest.nibio.no/mtm/VedleggTilRapport/Hoydeanalyse.pdf>.



Figur 13: Analyse av forskjell mellom oppgitt stasjonshøyde og høydeverdi fra FKB<sup>11</sup>.

Analysen viser at det i hovedsak er en god overensstemmelse mellom den oppgitte terrengkoten og DTMenes høydeverdi. Rundt 50 % av stasjonene har et avvik på mindre enn to meter, og ca. 83 % ligger innenfor en toleranse på 10 meter. Et avvik på 10 meter gir en feil på 0,06 °C pr. dag, noe som tilsvarer maksimalt 11 °C over en hel vekstsesong (6 måneder). Dette utgjør i det fleste tilfelle mindre enn 1 % av det samlede resultat, så det er ikke snakk om en betydelig feilkilde sammenliknet med de øvrige usikkerhetene i målinger og beregninger.

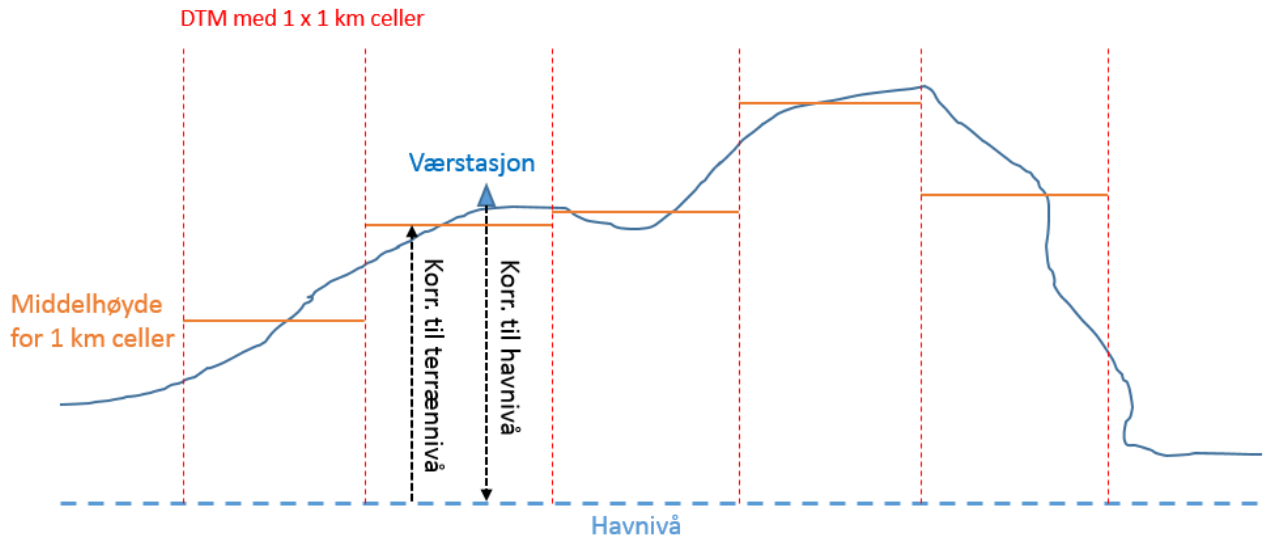
Til gjengjeld viser analysen også at det er 15 stasjoner med et avvik større enn 10 meter, inkludert fire stasjoner hvor avviket er større enn 50 meter. Her fremstår særlig stasjonene Rindal, Soknedal og Meldal med avvik på henholdsvis 295, 495 og 141 meter, noe som skyldes at det ikke er lagt inn noen høyde for dem i LMTs database. Det finnes imidlertid eksempler på stasjoner hvor høyde er oppgitt, men hvor det likevel er store avvik. Et eksempel på dette er Rennebu, hvor høyden er oppgitt til 294 meter, men målt til 212 meter i DTMen. Dette utgjør en forskjell på 82 meter. Et så stort avvik vil gi en feil i varmesummen på ca. 0,5 °C pr. dag, noe vi vurderer som en betydelig feil. Hva som er årsaken til en feil av denne størrelse, kan vi ikke si med sikkerhet, men det kan for eksempel være mulig at stasjonen har vært flyttet fra en lokalitet til en annen, og at høyden ikke er oppdatert samtidig.

Vi anbefaler at både x- og y-koordinater og terrenghøyde for alle stasjoner kontrolleres og korrigeres før høydekorreksjonen settes i endelig drift.

### Forskjellige terrenghøyder i beregningene

Korreksjonen av den målte varmesum ned til havnivå skjer ved bruk av værstasjonenes oppgitte høyde, mens korreksjonene tilbake til aktuelt terrengnivå skjer ved å bruke en høydeverdi fra en DTM. Problematikken er skissert i Figur 14.

<sup>11</sup> FKB -Felles KartdataBase: De mest høyoppløselige, autoritative vektorkartene i Norge



Figur 14 Terrengprofil som viser problematikken ved bruk av forskjellige høydegrunnlag til korreksjon.

Det kan oppstå en forskjell mellom varmesumsverdien som er målt for en gitt stasjon og den verdien som vises i 1 x 1 km-cellen stasjonen ligger innenfor. Dette skyldes måten systemet er satt opp på. Det er ikke noe metodisk feil i det, men man bør nok foregripe eventuelle misforståelser ved å opplyse brukerne om hvordan beregningen er foretatt, og hvorfor det kan være avvik mellom de viste verdiene, særlig hvis de målte verdiene vises som *labels*. Ellers er det en risiko for at brukeren avskriver resultatet som lite troverdig.

#### Antall dager som inngår i beregningene

I beregningen av det høydekorrigerte resultatet inngår også det antallet dager som uttrekket dekker. Dette er fordi høydeforskjellen påvirker hver enkelt av døgnmiddeltemperaturene som inngår i varmesumberegningen. Likevel inkluderes bare døgnmiddeltemperaturer som er over den valgte basistemperaturen i uttrekket, mens antall dager regnes ut ved en enkel beregning av forskjellen mellom angitt start- og sluttdato. Ettersom noen døgn i den valgte perioden potensielt kan bli utelatt fra beregningen, spørres det om antallet dager som inngår i høydekorreksjonen heller burde avspeile antallet dager hvor varmesumbidraget er positivt, altså større enn den valgte basistemperaturen

Dette er ikke noe vi har undersøkt konsekvensene av, men vi vurderer at problemet er forholdsvis lite, så lenge det primært søkes på datoer innenfor vekstsesongen, Da vil temperaturene i hovedsak vil ligge over en relevant basistemperatur. Vi vil likevel luften temaet overfor NIBIO, så problematikken kan drøftes og eventuelt undersøkes nærmere.

#### 4.3.4 Oppsummering

På bakgrunn av våre undersøkelser, med testing av forskjellige verktøy og datasett, anbefaler vi bruk av GRASS-funksjonen `r.mapcalculator` for å gjennomføre høydekorrigeringen på en samlet DTM med 50 x 50 meter oppløsning i `.vrt`-format. Vi har ytterligere fått klarlagt at det finnes flere feilkilder man bør være oppmerksom på, blant annet de oppgitte stasjonshøydene i LMTs database, terrengmodellens oppløsning og antall dager som inngår i høydekorreksjonen.

Vi anbefaler at NIBIO går igjennom og kontrollerer stasjonenes plassering, både i x, y og z-retning for å minimere feilene dette genererer i en eventuell høydekorreksjon. Vår analyse av stasjonshøydene

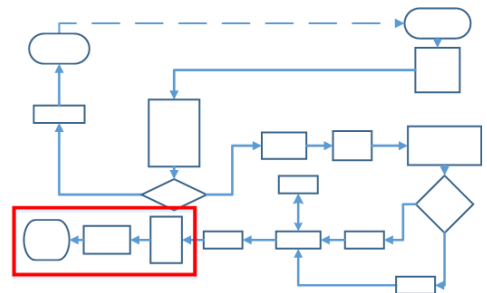
kan være et innspill til dette arbeidet Videre bør det undersøkes nærmere hvilken påvirkning generaliseringen har på resultatet, slik at det kan bestemmes når i prosessen det er mest optimalt å generalisere fra en 50 meter oppløsning, til den 1000 meters oppløsning vi skal vise i resultatet. En tidlig generalisering vil antakelig gi bedre ytelse ettersom datasettene reduseres i størrelse, men til gjengjeld risikerer man å miste noen nyanser i dataene som kunne gitt et mer presist endelig resultat. Dette er noe man kan utforske og justere når en etablert og funksjonell basisløsning foreligger.

#### 4.4 Presentasjon på webside

I kapittel 3 har vi beskrevet hvilken brukerflate vi tenker varmesumskartet kan ha i den endelige løsningen. I dette underkapittelet vil vi beskrive hva som skal gjøres rent teknisk for å oppnå det ønskede innholdet i brukerflaten.

I designfasen bestemte vi følgende i forhold til løsningens utseende:

- Resultatet av varmesumsberegningen skal presenteres som et rasterlag i et innebygget kart under klimastatistikk-fanen på hver stasjonsside i LMT.
- Rasterlaget skal fargelegges med en monokrom fargeskala i 10 intervaller, hvor hver enkelt dekker 10 % av verdiomfanget av den aktuelle varmesumsberegning, og som er adskilt ved svertingsgrad.
- Det skal genereres en tegnforklaring som viser sammenhengen mellom farger og celleverdier for det aktuelle rasterlaget.
- Det skal være mulig å få opplyst varmesummen som et tall for de enkelte celler i rasterlaget, for eksempel via en `GetFeatureInfo`-funksjon.
- Det skal dessuten også gis muligheter for zoom- og pan-funksjoner som tillater navigering i kartet, innzooming til særlig interessante områder og så videre.
- Resultatsiden må i tillegg inneholde en rekke metadata, en enkel veiledning for navigering i kartet, og forklarende tekster som forteller hva varmesumkartet beskriver – og *ikke* beskriver.



Vi antar, på bakgrunn av den erfaringen vi så langt har gjort oss, at Open Layers kan brukes til å fremstille et webkart som består av bakgrunnskart, resultatet av geoprosesseringen som et rasterlag og værstasjonene som punkter. Videre antar vi at fargeleggingen av rasterlaget kan skje ved å bruke en SLD-fil med normalisert skala som visualiserer alle resultater uansett verdiomfanget i dataene.

##### 4.4.1 Valg av kartklient

Resultatet av varmesumsberegningen skal sendes tilbake til LMT-siden forespørselen kom fra. Vi så for oss at kartet skulle settes inn som en *IFRAME*. Det finnes flere forskjellige teknologier/software for å lage kart som kan settes inn som et objekt i en *IFRAME*. Fra tidligere kjenner vi blant annet til Leaflet og Open Layers. Fordelen med å bruke denne type kartklient er at mange elementer allerede er innebygget i løsningen, for eksempel zoom- og panoreringsfunksjoner, og det er forholdsvis enkelt å sette opp bakgrunnskart og temalag. Teknologien rundt innsetting av en *IFRAME* er standard webteknologi, og vi vil ikke beskrive denne nærmere her.

Vi har satset på å bruke Open Layers til presentasjon av rasterkartet. Vi har også fått Open Layers anbefalt fordi det er en «moden» teknologi som inneholder mye funksjonalitet. Leaflet kunne ha vært et alternativ, men denne anses ikke som like «moden» enda, og inneholder ikke de samme mulighetene for skreddersøm som Open Layers<sup>12</sup>.

#### 4.4.2 Visning av varmesum som tallverdier

Vi ønsket å legge inn en `GetFeatureInfo` for rastercellene i dette kartet; dette ville gi rådgiverne en mer presis informasjon om eksakt varmesum i et gitt område. Ved å søke på Open Layers eksempelsider, ser vi at det bør være mulig å sette opp `GetFeatureInfo` også for rasterlag, men vi har ikke klart å legge dette inn på en måte som fungerer. Vi har dessuten søkt hjelp fra mer erfarne folk til å løse dette, men uten resultat. Det viser seg at `GetFeatureInfo` på raster ikke er helt enkelt i Open Layers 3. Vi anbefaler likevel at dette legges inn i den endelige løsningen.

#### 4.4.3 Visning av rasterlaget

I utgangspunktet antok vi at vi kunne ta tak i den resulterende tiff-filen som inneholder det interpolerte og høydekorrigerte rasteret, og sette denne direkte inn i et Open Layers-kart som inneholder bakgrunnskart og et punktlag som viser stasjonene. Det viser seg imidlertid at man vil trenge en god del kunnskap og erfaring med Open Layers 3 (eller tilsvarende webplattform) for å få dette til på en god og lesbar måte.

Våre forsøk på å sette inn rasteret som en tiff-fil resulterte i et «usynlig» kartlag – det lå der, men hadde ingen visningsparametere. For å utarbeide en egnet presentasjon av rasterverdiene måtte vi ha satt opp en CSS-definisjon i klienten. CSS tilsvarende på mange måter det samme som et SLD-oppsett i GeoServer, men dette var ikke noe vi hadde gjort før, og det var ikke tid til å finne en metode for visning av den planlagte prosentvise rasterskalaen ved hjelp av denne metoden. For å kunne løse presentasjonen til prototypen, valgte vi å bruke et SLD-oppsett i stedet.

#### 4.4.4 Fargeleggingsmetode

Vi antok at hvis vi bruker en monokrom skala med 10 forskjellige stigende svertningsgrader, hvor hvert trinn dekker 10 % av det aktuelle rasterets verdispenn, vil brukeren enkelt kunne tolke hvilket intervall en gitt celle tilhører, og hvordan varmesummen varierer omkring den valgte stasjon.

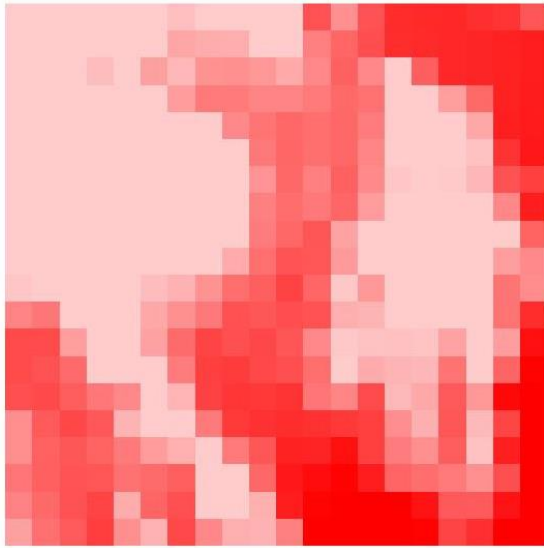
Vi har undersøkt om det finnes en SLD-mal for fargelegging av raster som støtter fargelegging etter fastsatte, prosentvise andeler av det samlede verdiomfang, og som også støtter bruk av ulik svertningsgrad innen den samme fargenyansen.

I GeoServers SLD-Cookbook har vi funnet den såkalte *alpha channel* –effekten, hvor alle verdier i rasteret fargelegges med samme fargekode, men hvor cellenes gjennomsiktighet (*opacity*) stiger eller avtar med voksende celleverdi. Da kan celler med de høyeste verdiene for eksempel settes røde med lav gjennomsiktighet, og derfor framtre som sterkt røde, mens celler med de laveste verdier ser svakt røde ut på grunn av høy gjennomsiktighet (Figur 15).

---

<sup>12</sup> <http://gis.stackexchange.com/questions/33918/should-i-use-openlayers-or-leaflet>

Konsultert 06.01.2015



*Figur 15: Eksempel på visning av variasjon via forskjellige svertningsgrader (opacity)*

Spørsmålet er om dette rasteret vil være anvendelig oppå et bakgrunnskart, eller om de ulike grader av gjennomsiktighet vil være forvirrende for brukeren.

Et alternativ kan være å definere fargene (via hexkode) for alle 10 intervaller, med den samme gjennomsiktigheten, som i Figur 16:



*Figur 16 Varmesumkart basert på SLD-oppsett med diskrete farger som dekker et likt intervall, og med lik transparens for alle fargene.*

Dette vil antagelig gi et mer lesbart kart. Dette eksemplet er ikke helt monokromt, men viser likevel en tydelig gradering av fargeverdier.



## Fargelegging etter prosentfordeling av verdier

Det bør være mulig å normalisere verdiene i et raster innen verdiene 0-255 via følgende syntaks:

```
<StyledLayerDescriptor xmlns="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0.0" xsi:schemaLocation="http://www.opengis.net/sld StyledLayerDescriptor.xsd">

  <NamedLayer>
    <Name>Brightness and contrast</Name>
    <UserStyle>
      <Title>SLD Cook Book: Brightness and contrast</Title>
      <FeatureTypeStyle>
        <Rule>
          <RasterSymbolizer>
            <ContrastEnhancement>
              <Normalize/>
              <GammaValue>0.5</GammaValue>
            </ContrastEnhancement>
            <ColorMap>
              <ColorMapEntry color="#008000" quantity="70"/>
              <ColorMapEntry color="#663333" quantity="256"/>
            </ColorMap>
          </RasterSymbolizer>
        </Rule>
      </FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

*Tekstboks 6: SLD-script som understøtter bruk av svertningsgrad (opacity)*

I følge SLD Cookbook som er en del av GeoServers dokumentasjon, er det mulig å normalisere verdiene i et raster fra 0 -255<sup>13</sup>, men style editoren i Geoserver kan ikke validere syntaksen. Det kan se ut som om normaliseringen ikke virker sammen med ColorMap-oppsettet.

### 4.4.5 Tegnforklaring

Det skal også ligge en tegnforklaring sammen med kartet. Denne genereres i vårt tilfelle automatisk av SLD-oppsettet i GeoServer. Ved å bruke en GetLegend-URL i et HTML-kall, får vi også plassert et eget objekt som inneholder tegnforklaringen i en IFRAME på websiden i LMT.

### 4.4.6 Metadata og forklarende tekster

Som nevnt ved gjennomgang av designmodellen skal løsningen også inneholde noen metadata og forklarende tekster. Som metadata kan vi nevne de brukervalgte parametrene som er utgangspunktet for beregningen. Disse vises allerede i dag på stasjonssiden når man har startet en varmesumberegning. De finnes dermed som variabler på siden, og kan hentes derfra inn i en html-tekst via et script, hvis de skal vises et annet sted på siden.

<sup>13</sup> [http://docs.geoserver.org/latest/en/user/downloads/raster\\_brightnessandcontrast.sld](http://docs.geoserver.org/latest/en/user/downloads/raster_brightnessandcontrast.sld) . Konsultert 11.01.2016.

De øvrige tekstene legges inn på siden som faste html-tekster ved siden av kartet. Det kan for eksempel være en veiledning i bruken av kartet, informasjon om nøyaktighet i beregningen og øvrig informasjon om hvilke faktorer som inngår i beregningen, og hvilke som ikke gjør det. Et forslag er vist i bilag VI.

#### 4.4.7 Oppsummering

Gjennom arbeidet med presentasjon av resultatet i en kartklient på stasjonssiden har vi funnet indikatorer på hvordan de forskjellige utfordringene kan løses rent teknisk, men må erkjenne at vår kunnskap i Open Layers 3 ikke er gode nok til å komme i mål med løsningen. Vi mener likevel fortsatt at Open Layers kan brukes til å løse oppgaven, kombinert med standard HTML-verktøy til oppsett av tekster og metadata, men at det trengs noen med mer utviklingskompetanse for å nå det ønskede resultatet.

### 4.5 Automatisering av prosessen

Fra oppstarten av prosjektet har det vært klart for oss at en av de vanskeligste oppgavene i designet og oppbygningen av kartkomponenten i LMT ville bli å sikre automatisk avvikling av de forskjellige trinn i prosessen i den riktige rekkefølge. Hvordan sikrer vi at alt går automatisk, fra brukerens inntasting av parametere i en stasjonsside i LMT, til visning av resultatet på et kart på samme stasjonsside?

Vi vet at geografiske rutiner kan kobles sammen og avvikles i en bestemt rekkefølge, for eksempel i ArcGIS ModelBuilder og et tilsvarende verktøy i QGIS. Vi antar derfor at også de algoritmene og rutinene vi har valgt som en del av den samlede prosess i LMT kan settes opp til automatisk kjøring i en bestemt rekkefølge, så output fra en delprosess brukes som input i den neste. Hvis vi søker etter eksempler på hvordan de valgte algoritmer er brukt i automatiske prosesser i andre, tilsvarende systemer, kan vi finne frem til en metode eller verktøy til å sette opp en automatisk avvikling av delprosessene i LMTs kartløsning.

Vi regnet med at dette ville kreve oppbygging av et script i et programmeringsspråk, for eksempel Python, som kunne holde styr på input og output fra de forskjellige prosessene, slik at output fra en prosess ble input i neste trinn.

Vi vil gjerne utvikle de følgende prosessene automatisk, og i nevnte rekkefølge:

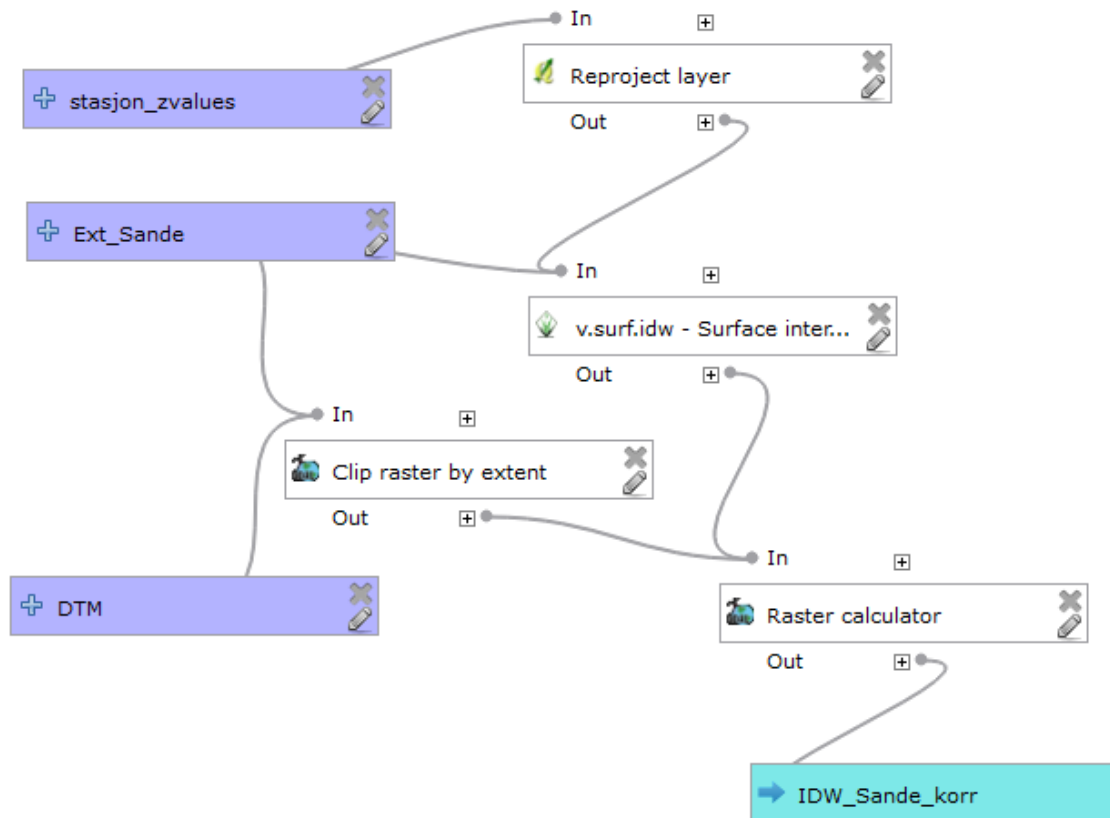
- Omgjøring av de inntastede brukerparametere til inputvariabler i webservice-kallet,
- Kall av webservice for uttrekk av data,
- Interpolasjon,
- Høydekorreksjon via DTM,
- Avgrensning av resultatet rundt den valgte værstasjon, samt
- Visning av resultatet i et webkart.

Her beskriver vi de refleksjoner og forsøk vi har gjort i løpet av prosjektet for å få avviklet prosessene automatisk, både enkeltvis og i sammenheng.

#### 4.5.1 Modellverktøy i QGIS

Det første vi tok fatt i var geoprosesseringen – interpolasjon og høydekorreksjon. Vi brukte i første omgang QGIS til å prøve ut hvordan de forskjellige algoritmene fungerte, og ble i den forbindelse oppmerksomme på at QGIS inneholder verktøyet *Graphical Modeler* (engelsk versjon), som fungerer

noenlunde på samme måte som Model Builder i ArcGIS. Det er et verktøy som man kan bruke til å bygge oppgaver sammen til en prosess av forskjellige operasjoner i en bestemt rekkefølge, og hvor man kan definere input og output til de forskjellige operasjonene. Ved hjelp av dette verktøyet satte vi sammen modellen som vises i Figur 17.



Figur 17 Testmodell laget ved hjelp av QGIS' verktøy Graphical Modeler

I ArcGIS *Model Builder* kan man eksportere modeller som Python-script. Denne funksjonen har også eksistert i QGIS frem til versjon 2.6, men den er for tiden under ombygging, og er derfor fjernet fra de senere versjonene av QGIS<sup>14</sup>.

Det viser seg også at disse scriptene bare fungerer så lenge de kjøres innenfra QGIS' brukergrensesnitt, så metoden kan ikke brukes til å kalle eller aktivere funksjonene uten at QGIS kjører<sup>15</sup>. Vi ble tidlig i prosessen frarådet å bruke QGIS som «GIS-motor», og har derfor ikke gått videre med å undersøke om det rent teknisk lar seg gjøre.

#### 4.5.2 GRASS og Python

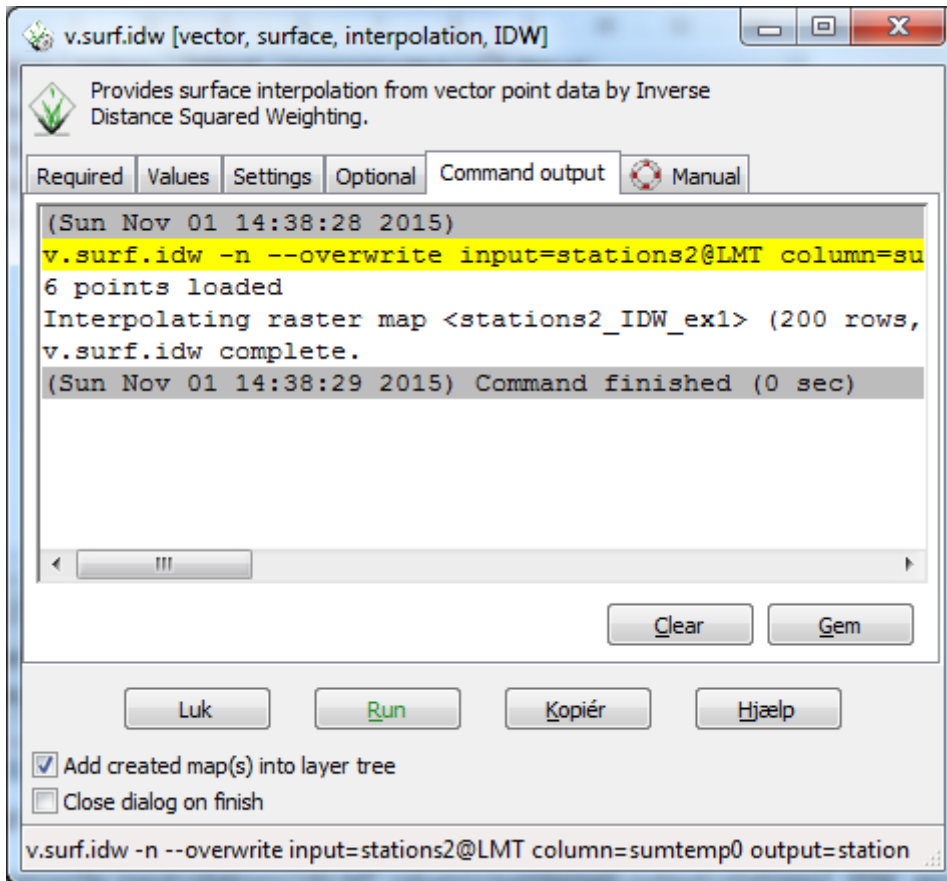
I det videre arbeidet tok vi utgangspunkt i at interpolasjonen skulle løses ved hjelp av GRASS-algoritmen *v.surf.idw*. GRASS-algortimene er ikke bare tilgjengelige via QGIS, men har sitt eget brukergrensesnitt, og viste seg også å ha et Python-API (Application Programming Interface). Det

<sup>14</sup> <http://gis.stackexchange.com/questions/121537/export-model-to-python-script-in-qgis-2-6>. Konsultert 5.1.2016.

<sup>15</sup> <http://gis.stackexchange.com/questions/29580/how-to-run-a-simple-python-script-for-qgis-from-outside-e-g-sublime-text?lq=1>. Konsultert 5.1.2016.

skulle dermed være mulig å kalle GRASS-funksjoner uten at et GRASS-grensesnitt kjører på serveren<sup>16</sup>.

Først testet vi om GRASS<sup>17</sup> kunne bruke et datasett som inneholdt værstasjoner med varmesummer til å utføre en interpolasjon via v.surf.idw-algoritmen. Vi satte vi opp funksjonen via en dialogboks (se Figur 18), som også kunne vise kommandoens oppbygning



Figur 18 Dialogboks som viser kommandosyntaks for v.surf.idw-algoritmen

Denne kommandoen testet vi deretter i GRASS' Python-konsoll med følgende syntaks:

```
grass.run_command(v.surf.idw, --overwrite, input=stations2@LMT, column=sumtemp0,
output=stations2_IDW_ex1, npoints=4)
```

Tekstboks 7: Kommando til aktivering af v.surf.idw-funksjonen i Python-syntaks.

Hensikten med dette var å finne ut hvordan kommandoene i GRASS skulle bygges opp for å kunne aktivere GRASS-funksjoner via Python-APIet. Avvikling i GRASS-brukerflaten gikk fint, men forsøk på å gjennomføre kommandoen via det innebyggede Python-shell resulterte i en feil. Videre forsøk basert på spørsmål og svar på diverse internettfora ga heller ikke noe brukbart resultat.

På dette tidspunktet bestemte vi oss for å søke råd hos en foreleser. I første omgang trengte vi råd om valg av klientsoftware til kartvisning, men som erfaren webutvikler kunne han se av vår

<sup>16</sup> <https://grass.osgeo.org/grass70/manuals/libpython/>. Konsultert 5.1.2016.

<sup>17</sup> I testen er anvendt GRASS 7.0 til Windows.

beskrivelse av problemstillingen at vi hadde behov for en *process wrapper*. Dette var et nytt begrep for oss, men ut fra beskrivelsen kanskje det vi trengte for å løse automatiseringsproblematikken (Morten Fuglsang, pers. medd.).

#### 4.5.3 PyWPS som *process wrapper*

Etter dette tipset valgte vi å undersøke muligheten for å bruke programmet PyWPS til å knytte sammen de ulike prosesselementene, slik at de kan avvikles fortløpende på grunnlag av brukerens input.

Vi har tidligere sett på muligheten for å bruke WPS-teknologi (Web Processing Services) til å utføre geoprosesseringen (interpolasjon og høydekorreksjon), ettersom flere og flere funksjoner blir tilgjengelige som WPS. Vi fant likevel ikke noen tilgjengelige prosesser som tilfredsstilte våre behov. På det daværende tidspunkt lette vi etter interpolasjon via kriging.

PyWPS fungerer slik at geoalgoritmene som skal kjøres samles i et Python-script på webserveren. Scriptet kalles ved hjelp av et URL-kall, som kan inneholde kommandoene:

- GetCapabilities (gir en oversikt over tilgjengelige prosesser),
- DescribeProcess (gir detaljer om den enkelte prosess) eller
- Execute (utfører en bestemt prosess).

I kallet kan man angi en rekke inputparametere og -data, deriblant lenke til filer med geografisk innhold (for eksempel vektor- eller rasterfiler)<sup>18</sup>. Kallet kan for eksempel komme fra en webgis-klient som Open Layers, og WPS-servicen kan deretter returnere geografiske data som vises på kartet i webklienten. En av fordelene med PyWPS er at man kan bruke algoritmer fra blant annet GRASS direkte i scriptet.

Oppsett av PyWPS er ikke trivielt, selv om det finnes gode tinn-for-trinn-veiledninger tilgjengelige på nettet. Programvaren kan installeres på servere med Windows OS, men er opprinnelig utviklet for Linux-server. Ved å bruke de tilgjengelige veiledningene (blant annet <sup>19,20</sup>), tok det likevel ikke så lang tid for NIBIOs serveradministrator å installere programmet på lmttest-serveren.

Ved hjelp av eksempler på script, veiledninger og spørsmål/svar i diverse fora på nettet har vi forsøkt å utarbeide et script som kunne kjøre en GRASS-algoritme via PyWPS og returnere en rasterfil som kunne vises i en webklient. Vi har likevel måttet innse at vi mangler grunnleggende kunnskap om muligheter, regler og begrensninger i Python og oppsett av PyWPS. Det har ikke lyktes oss å etablere en automatisk kjøring av de ønskede prosesser via PyWPS.

Vi har altså ikke klart å få et brukbart resultat ut av PyWPS-prosessen vi har prøvd å sette opp. Vi ble da anbefalt å bryte scriptene våre ned til enkeltkommandoer, for å teste kommandoene enkeltvis, og så gradvis bygge dem ut etter hvert. På den måten kan man finne ut hvilke kommandoer som får scriptet til å feile. Metoden har også vært nyttig til å finne de kommando-elementene som tilsynelatende ikke kunne kjøres fra scriptet, men vi har ikke funnet alternativer til de script-elementene vi har funnet på nettet, og har heller ikke klart å finne effektive metoder til å teste

<sup>18</sup> [http://wiki.rsg.pml.ac.uk/pywps/Main\\_Page](http://wiki.rsg.pml.ac.uk/pywps/Main_Page) Konsultert 02.12.2015.

<sup>19</sup> [http://wiki.ieee-earth.org/Documents/GEOSS\\_Tutorials/GEOSS\\_Provider\\_Tutorials/Web\\_Processing\\_Service\\_Tutorial\\_for\\_GEOSS\\_Providers/Section\\_4%3A\\_Provisioning%2F%2FUsing\\_the\\_Service\\_or\\_Application/Steps\\_for\\_Plymouth\\_Marine\\_Laboratory\\_implementation\\_of\\_Python\\_WPS](http://wiki.ieee-earth.org/Documents/GEOSS_Tutorials/GEOSS_Provider_Tutorials/Web_Processing_Service_Tutorial_for_GEOSS_Providers/Section_4%3A_Provisioning%2F%2FUsing_the_Service_or_Application/Steps_for_Plymouth_Marine_Laboratory_implementation_of_Python_WPS), konsultert 02.12.2015

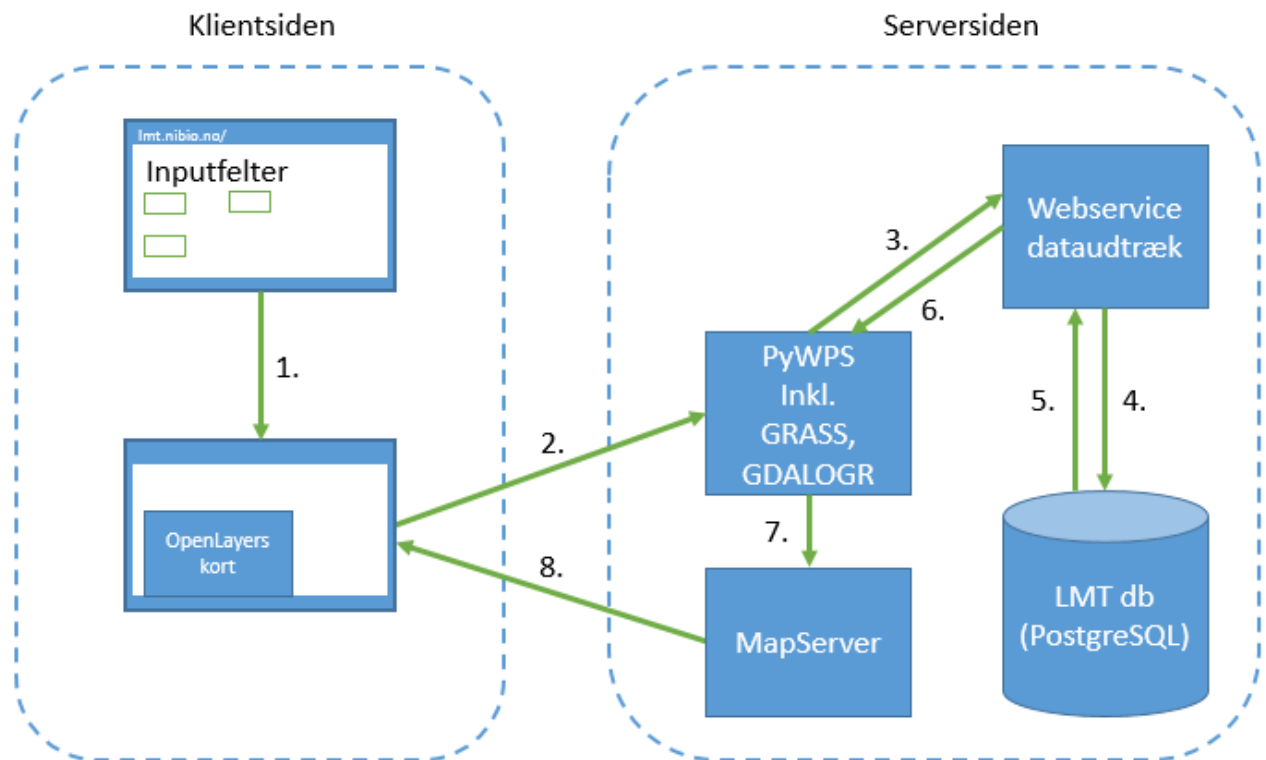
<sup>20</sup> <http://pywps.wald.intevation.org/index.html>, konsultert 02.12.2015

enkeltkommandoene, blant annet fordi vi ikke visste hvilket output vi kunne forvente, og hvor outputet ble lagt. Dokumentasjon fra arbeidet med PyWPS er samlet på NIBIOs testside.<sup>21</sup>

### Våre erfaringer og anbefalinger

Ved oppstarten av PyWPS-testene antok vi at vi kunne bygge opp et samlet script som kunne utføre uttrekk av data, interpolasjon og høydekorrigering, samt eksportere og vise resultatet via WMS, ved hjelp av eksempler på script og veiledning. Underveis justerte vi målet til å bare sette opp en prosess som kunne importere data i GeoJSON-format og utføre interpolasjonen. Selv om heller ikke dette lyktes, tror vi likevel at det lar seg gjøre hvis den riktige kompetansen trekkes inn i prosjektet. Det bør være noen som har erfaring med oppsett og bruk av PyWPS. Det finnes tross alt eksempler på fungerende løsninger hvor PyWPS brukes til å utføre geoprosessering og beregning<sup>22</sup>.

Ut fra våre antagelser om hvordan PyWPS kan brukes til å automatisere den samlede prosessen, har vi utarbeidet en ny skisse over den tekniske arkitekturen i løsningen (Figur 19).



Figur 19 Løsningsarkitektur for LMTs kartkomponent ved bruk av PyWPS

<sup>21</sup> <http://lmttest.nibio.no/mtm/VedleggTilRapport/PyWPS1.pdf>.

<sup>22</sup> [http://wiki.rsg.pml.ac.uk/pywps/Main\\_Page](http://wiki.rsg.pml.ac.uk/pywps/Main_Page). Konsultert 02.12.2015.

#### Forklaring til Figur 19:

1. Brukervalg av parametere på LMT-side utløser initiering av kartobjekt på siden.
2. OpenLayers utløser eksekvering av WPS-request inklusiv brukervalgte parametere.
3. PyWPS kaller webservice med de valgte parametere.
4. SQL-kall mot databasen med de valgte parametere.
5. Resultat retur til webservice.
6. Datauttrekk som GeoJSON inngår i interpolasjon og høydekorreksjon, styrt av PyWPS. DTM finnes tilgjengelig på serveren.
7. Resultat av geoprosessering (rasterbilde) sendes til WebMapServeren.
8. Svar på WPS-request omgjøres til en WMS og vises som kartlag i kartklienten.

#### 4.5.4 Den valgte værstasjon

Ut fra den skisserte løsningsarkitekturen vurderer vi at PyWPS kan håndtere prosessene 2 og 3, og 6-8, som beskrevet i bilag III, mens webservicen for datauttrekk beskrevet i kapittel 4.1 sørger for uttrekk fra databasen (prosess 4 og 5). Vi mangler likevel fortsatt å få «båret» den valgte stasjon og de brukervalgte parametrene med fra stasjonssiden i LMT og over til de øvrige prosessene (prosess 1). Stasjonens plassering (koordinat) skal blant annet brukes til å bestemme avgrensningen av det resulterende rasterlag omkring den valgte stasjonen.

NIBIOs webutvikler mener at det vil være enkelt å lage en kommando i JavaScript som fanger opp de brukervalgte parametrene og gjør dem til variabler som kan kalles og inngå i andre script. Denne oppgaven regner vi derfor med at NIBIO kan stå for selv. I samme rutine vil man antagelig også kunne hente stasjons-IDen eller koordinatet som en variabel. Det vil være nødvendig å bygge inn både de brukervalgte parametrene og den valgte stasjonens plassering i URL-syntaksen i WPS-kallet, og på den måten overføre dem til PyWPS, hvor de kan omdannes til brukbare variabler i geoprosesseringen.

#### 4.5.5 Oppsummering

Vi har gjennom våre undersøkelser av tekniske komponenter lært at det finnes flere forskjellige muligheter for å kalle geoprosesseringsalgoritmer fra Open Source funksjonsbiblioteker (GRASS, med flere) via Python-script, slik at prosessene kan avvikles i den riktige rekkefølgen, og motta relevante inputdata og -parametere. Resultatene tyder på at PyWPS kan settes opp til å styre den samlede prosessen, siden det der finnes direkte integrasjon til GRASS-funksjonene og til MapServer, som kan generere en WMS-tjeneste av resultatet av geoprosesseringen. Denne kan deretter vises som et webkart.

Vi må imidlertid innse at vi ikke har den tekniske kompetansen som kreves for selv å sette opp PyWPS på riktig måte. Likevel tror vi at i samarbeid med en erfaren WebGIS-utvikler ville vi kunne spesifisere og få utviklet denne komponenten.

#### 4.6 Prototypen

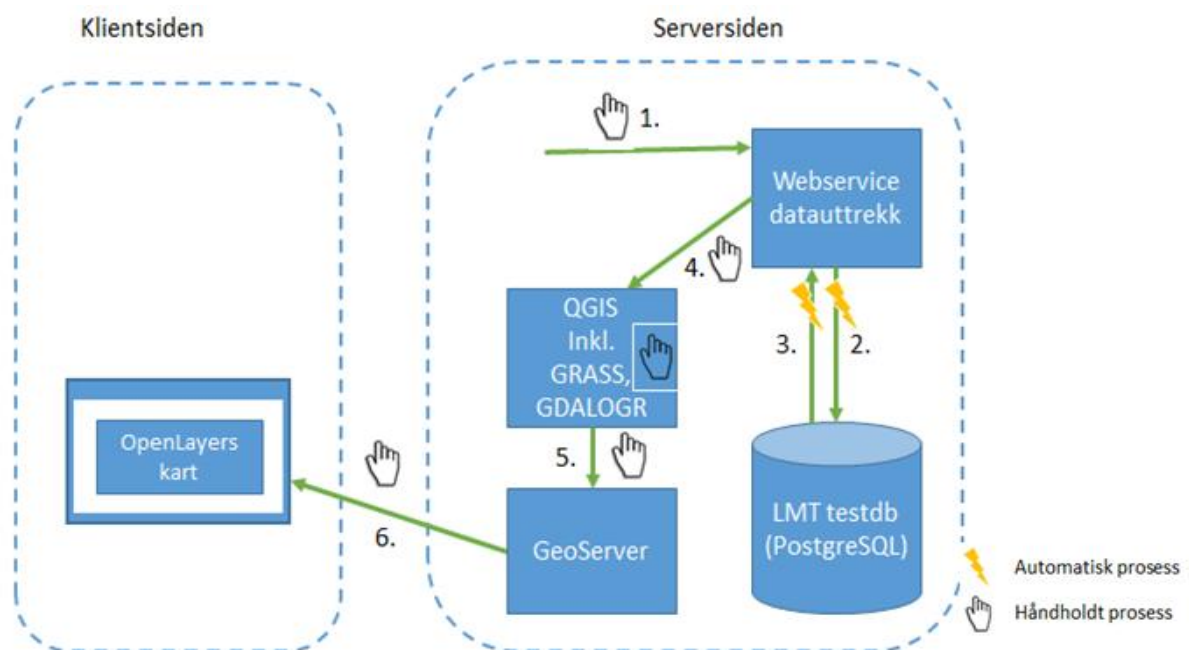
I dette avsnittet vil vi beskrive hvordan vi har satt opp den tekniske løsningen som er prototypen. Prototypen er tenkt brukt for å demonstrere hvordan vi tenker systemet skal fungere når ovennevnte tekniske utfordringene er løst i den endelige implementeringen. Oppsettet vi beskriver er brukt til å lage de geovisualiseringene som inngikk i brukertesten vi vil beskrive i neste kapittel.

## Hva er en prototype?

En prototype er en tidlig eller foreløpig versjon av en tenkt løsning, som skal demonstrere hvordan den endelige løsning vil se ut og hvordan den vil fungere. Prototyping av brukergrensesnitt fokuserer på utseende og innhold i brukergrensesnittet til et system, mens teknisk prototyping fokuserer på å teste ut virkemåten og funksjonaliteten i de tekniske komponentene (Reeve & Petch, 1999).

Vår prototype har først og fremst det formål å vise brukerne hvordan den endelige løsningen kommer til å se ut. Vi bruker den også til å demonstrere at det rent teknisk kan la seg gjøre å nå frem til det viste resultat, selv om prosessene ikke avvikles fullstendig automatisert i prototypen.

Under har vi satt inn en figur (Figur 20), som viser hvilke deler av den endelige løsningen prototypen inneholder, og hvilke som avvikles henholdsvis automatisk og manuelt. I resten av kapittelet går vi gjennom de enkelte prosessene i mer detalj.



Figur 20: Teknisk oppsett av prototype



#### 4.6.1 Datauttrekk

I den endelige løsningen skal brukeren kunne oppgi sine parametere via LMT-stasjonenes klimastatistikk-fane. Vi har imidlertid enda ikke laget noen funksjon som henter disse brukervalgene i form av variabler som kan brukes i URL-kallet til webservicen. I brukertestene har vi derfor tastet inn parametrene manuelt i URLen (se eksempel nedenfor).

Brukerens inputdata (fra mail):	<b>Foretrukket værstasjon</b>	<b>Brunlanes klimastasjon</b>
	<b>Basistemperatur</b>	<b>6,0</b>
	<b>Startdato</b> (i 2014 eller 2015)	<b>1.5.15</b>
	<b>Sluttdato</b> (i samme kalenderår som startdatoen)	<b>1.9.15</b>

Disse brukervalgte parametere inngår i webservice-kallet som vist med rødt i denne URLen:

[http://lmttest.nibio.no/mtm/varmesumws\\_gison/varmesum.php?periodStart=2015-05-01&periodEnd=2015-09-01&basisTemp=6](http://lmttest.nibio.no/mtm/varmesumws_gison/varmesum.php?periodStart=2015-05-01&periodEnd=2015-09-01&basisTemp=6)

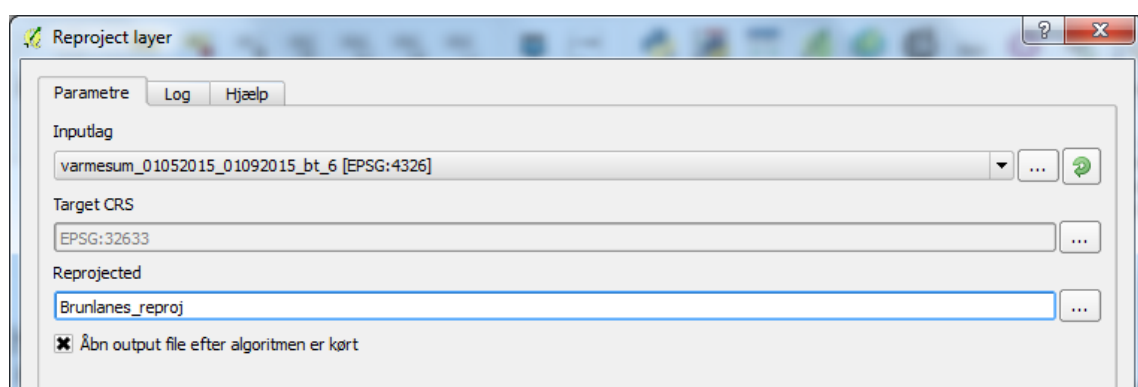
Webservice-kallet resulterer i et tekstoutput i GeoJSON-format i den aktuelle *browseren*. Dette kan lagres på en lokal mappe som en GeoJSON-fil via browserens «Lagre som...»-funksjon.

Filen åpnes deretter i QGIS sammen med høydemodellen.

#### 4.6.2 Geoprosessering

GeoJSON-filen har koordinater i lengde/breddegrader (EPSG: 4326) da det er disse verdier som ligger i databasen. Av hensyn til den senere høydekorreksjonen vil det være praktisk at outputet fra interpolasjonen er i samme projeksjon som høydemodellen, som ligger i UTM33, WGS84 (EPSG: 32633).

Første trinn i prosessen er derfor å reprojisere GeoJSON-filen til et datasett i EPSG:32633. Her brukes QGIS-funksjonen «Reproject layer». For å utføre denne prosessen må man bare angi hvilket datasett som skal brukes som input, og hvilken projeksjon outputet skal ha (Figur 21).



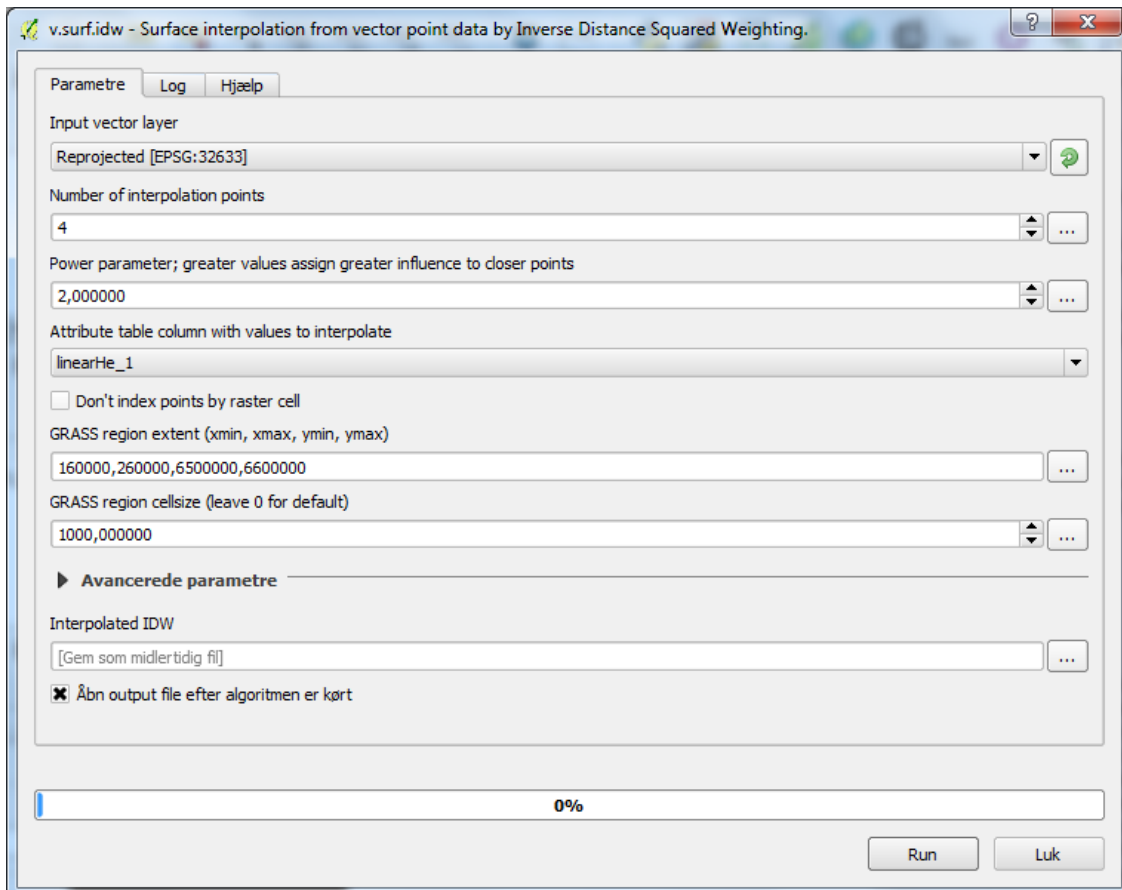
Figur 21 Reprojisering fra EPSG:4326 til EPSG:32633

Output fra prosessen blir et vektorlag som heter «Reprojected», som blir input til interpolasjonen.

Interpolasjonen gjøres ved hjelp av algoritmen «v.surf.idw», som kommer fra GRASS-biblioteket. De enkelte inputparametere er nærmere beskrevet i

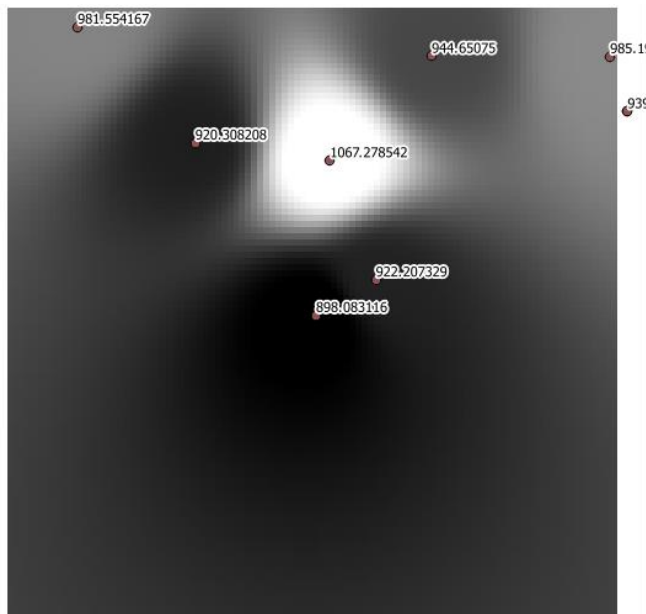
Figur 22. Det valgte attributtet som algoritmen skal interpolere på er varmesummen korrigeret til havnivå, altså feltet linearHeatSumZAdjusted. Dette feltet har i reprojiseringsprosessen blitt forkortet linearHe\_1.

Den valgte *bounding box* på 100 x 100 km er satt ut fra den valgte stasjonens koordinater, avrundet til hele kilometer (her stasjonen Brunlanes, som ligger på ca. 210000, 6550000), og celledørrelsen for det resulterende rasterlag er satt til 1000 x 1000 m.



Figur 22 Interpolasjon av det reprojiserte rasterlaget med varmesum, ved hjelp av Inverse Distance Weighting-algoritmen

Output fra prosessen er rasterlaget InterpolatedIDW, vist i Figur 23.

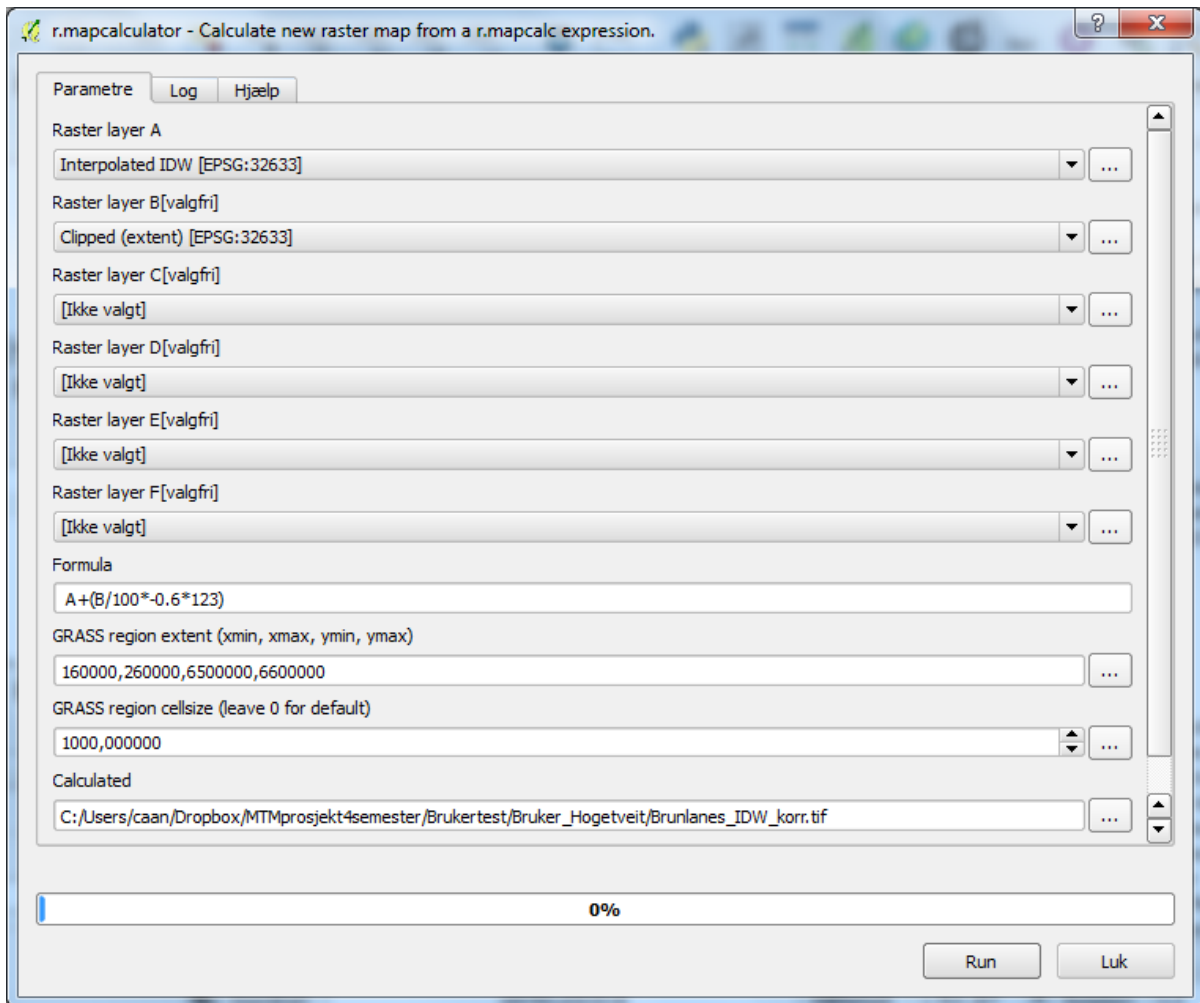


Figur 23: Resultatet av interpolasjonen

Neste trinn i prosessen er å foreta høydekorreksjon som beskrevet i kapittel 4.3. I prototypen har vi brukt et utklipp av høydemodellen som har samme utstrekning (*bounding box*) og cellestørrelse som resultatlaget fra interpolasjonen. For å lage dette utklippet bruker vi algoritmen «Clip raster by extent», som kommer fra GDALOGGR-funksjonsbiblioteket. Vi bruker samme *bounding box* som ved interpolasjonen, og cellestørrelsen på 1000 x 1000 meter settes ved å angi «outsize 100 100», som betyr at det resulterende rasterlaget deles opp i 100 x 100 celler.

Output fra prosessen er laget «Clipped(Extent)», som brukes i neste trinn, hvor den interpolerte varmesummen på havnivå korrigeres til aktuell terrenghøyde. Dette gjøres ved hjelp av rasterkalkulatoren, hvor celledverdiene i de to rasterlagene gjennom bruk av alminnelige regnefunksjoner resulterer i et nytt rasterlag. Både GRASS og GDALOGGR inneholder slike funksjoner. Her har vi brukt GRASS' «r.mapcalculator» (se Figur 24).

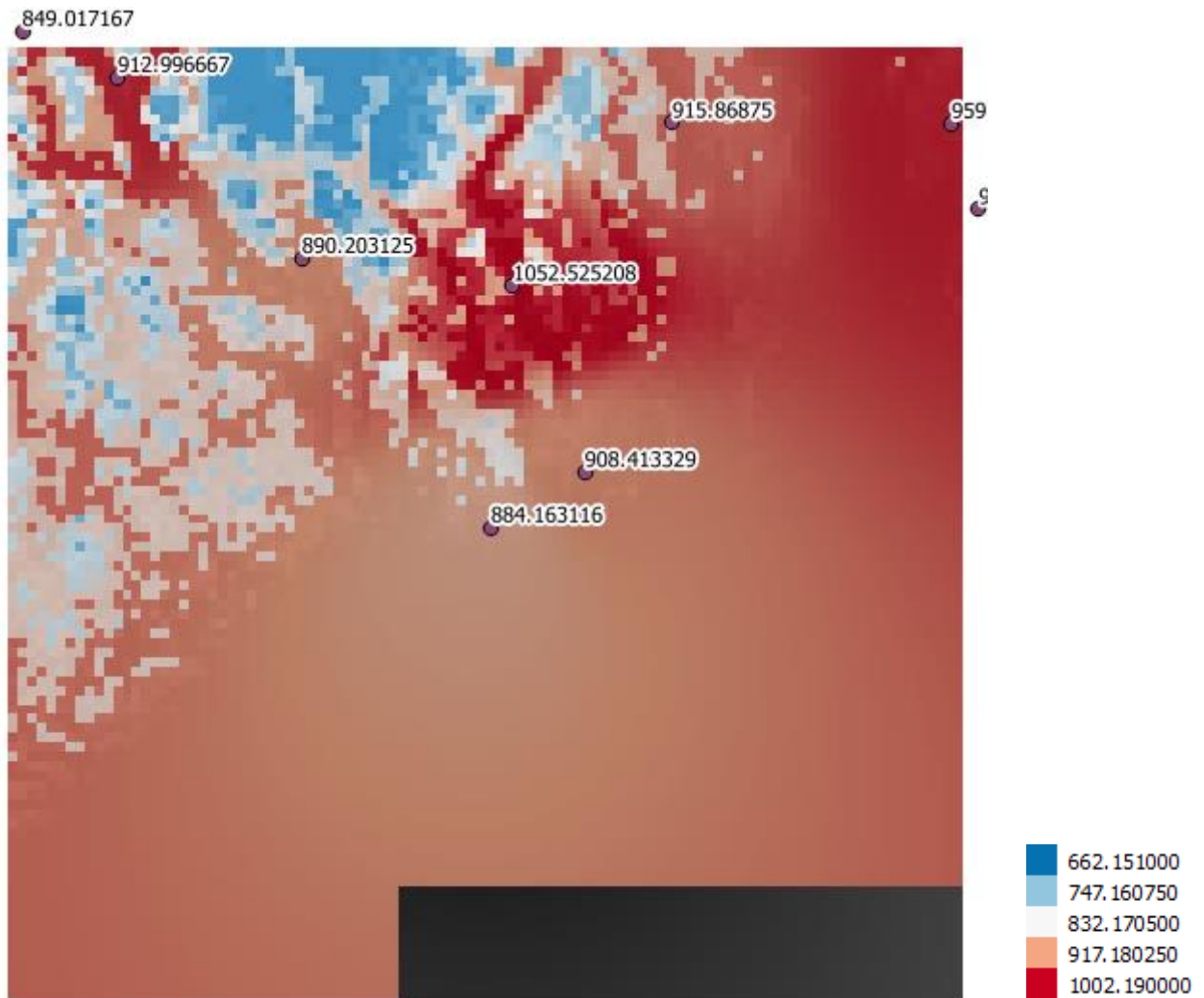
Vi har brukt samme *bounding box* og cellestørrelse som i de foregående prosessene.



Figur 24 Høydekorreksjon via rasterkalkulasjon

Formelen  $A+(B/100*-0,6*123)$  angir at høydekorreksjonen utføres over de 123 dagene som uttrekket omfatter.

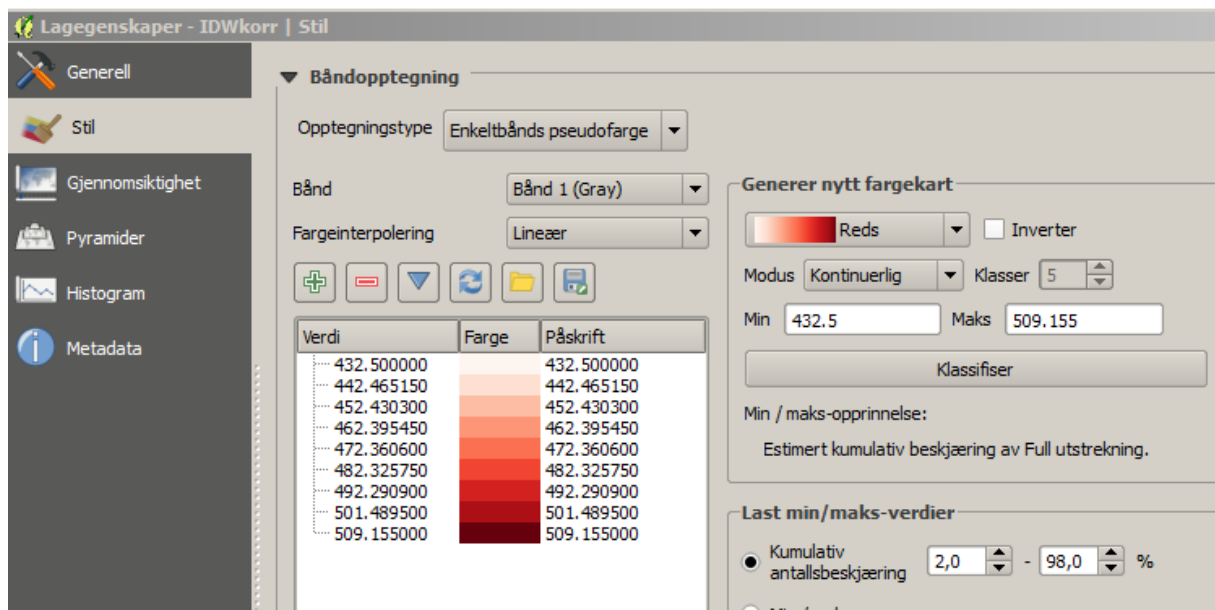
Resultatet er vist i Figur 25.



Figur 25 Resultatet av høydekorreksjonen av interpolasjonslaget.

#### 4.6.3 Visualisering via Open Layers

I prototypen hadde vi ingen automatikk for klassifisering av rasterverdier eller overføring til Open Layers-klienten. Vi baserte visningen av rasteret på at vi la den høydekorrigerte rasterfilen inn i et QGIS-kart. Ved hjelp av QGIS' stiloppsett for enkeltbåndes pseudofarge fikk vi klassifisert rasterverdiene i 9 klasser.



Figur 26: klassifisering av rasterverdier i QGIS

Deretter satte vi opp en WMS-tjeneste av rasterfilen i GeoServer hvor presentasjonen kan styres av en SLD-fil. Vi har ikke klart å implementere en automatisk prosentvis skala via SLD i GeoServer, i prototypen har vi endt opp med å definere en egen SLD-presentasjon for hver rasterfil. Vi brukte de klassifiserte verdiene fra QGIS til å sette opp en egen SLD-fil for hvert rasterbilde, for å få inn det riktige verdispennet. Resultatet «simulerer» den effekten vi er ute etter, men inneholder altså ikke en automatisert prosess for produksjon av presentasjonen.

Figur 27 og Figur 28 viser de to SLD-definisjonene som brukes for å simulere en lik, prosentvis skala for to ulike varmesumkart.

```

<FeatureTypeStyle>
  <FeatureTypeName>Feature</FeatureTypeName>
  <Rule>
    <RasterSymbolizer>
      <ColorMap extended="true">
        <ColorMapEntry color="#fff5f0" quantity="1037.0" label="1037 graddager" opacity="0.6"/>
        <ColorMapEntry color="#fee0d3" quantity="1111.0" label="1111 graddager" opacity="0.6"/>
        <ColorMapEntry color="#fcbda4" quantity="1184.0" label="1184 graddager" opacity="0.6"/>
        <ColorMapEntry color="#fc9e77" quantity="1257.0" label="1257 graddager" opacity="0.6"/>
        <ColorMapEntry color="#fb7050" quantity="1331.0" label="1331 graddager" opacity="0.6"/>
        <ColorMapEntry color="#f14431" quantity="1404.0" label="1404 graddager" opacity="0.6"/>
        <ColorMapEntry color="#d32020" quantity="1477.0" label="1477 graddager" opacity="0.6"/>
        <ColorMapEntry color="#ac1016" quantity="1545.0" label="1545 graddager" opacity="0.6"/>
        <ColorMapEntry color="#67000d" quantity="1601.0" label="1601 graddager" opacity="0.6"/>
      </ColorMap>
    </RasterSymbolizer>
  </Rule>
</FeatureTypeStyle>

```

Figur 27: SLD-oppsett for varmesum for Landvik 2014

```

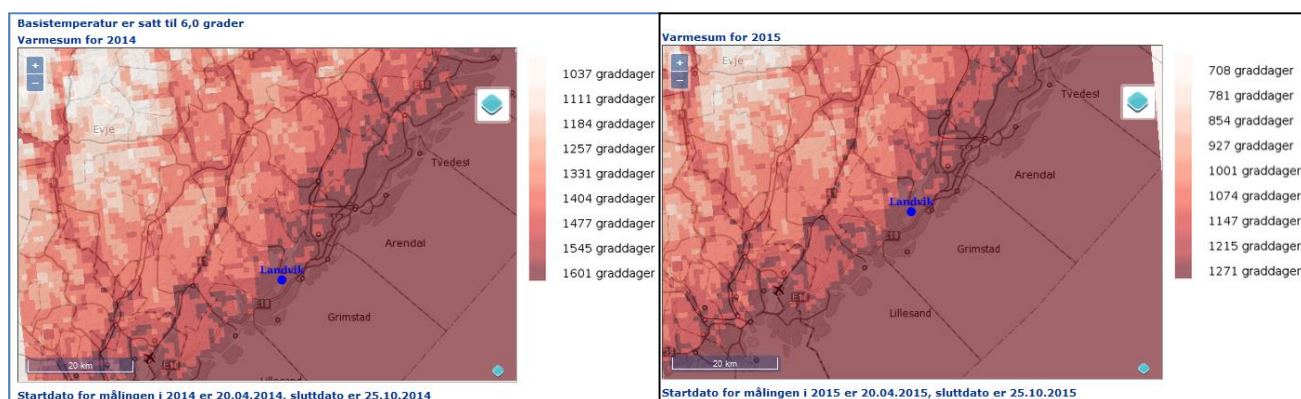
-----
<FeatureTypeStyle>
<FeatureTypeName>Feature</FeatureTypeName>
<Rule>
<RasterSymbolizer>
<Opacity>0,6</Opacity>
<ColorMap extended="true">
<ColorMapEntry color="#fff5f0" quantity="708.0" label="708 graddager" opacity="0.6"/>
<ColorMapEntry color="#fee0d3" quantity="781.0" label="781 graddager" opacity="0.6"/>
<ColorMapEntry color="#fcbda4" quantity="854.0" label="854 graddager" opacity="0.6"/>
<ColorMapEntry color="#fc9e77" quantity="927.0" label="927 graddager" opacity="0.6"/>
<ColorMapEntry color="#fb7050" quantity="1001.0" label="1001 graddager" opacity="0.6"/>
<ColorMapEntry color="#f14431" quantity="1074.0" label="1074 graddager" opacity="0.6"/>
<ColorMapEntry color="#d32020" quantity="1147.0" label="1147 graddager" opacity="0.6"/>
<ColorMapEntry color="#ac1016" quantity="1215.0" label="1215 graddager" opacity="0.6"/>
<ColorMapEntry color="#67000d" quantity="1271.0" label="1271 graddager" opacity="0.6"/>
</ColorMap>
</RasterSymbolizer>
</Rule>
</FeatureTypeStyle>
-----

```

Figur 28: SLD-oppsett for Landvik 2015

WMS-tjenestene ble deretter satt inn i eksisterende Open Layers-klienter som allerede inneholdt bakgrunnskart og stasjonspunkter.

Resultatkartene vises i Figur 29.



Figur 29: Samme fargeskala i bruk på to ulike verdispenn. Lik grad av gjennomsiktighet for alle verdier bidrar til at brukeren kan orientere seg i grunnkartet.

WMS-tjenestene som ble resultatet kunne deretter legges inn i Open Layers-kartet på vår *mockup*-webseite. Som tilleggsinfo i dette kartet ligger et bakgrunnskart<sup>23</sup>, og oversikten over stasjoner i LMT-nettet.

I likhet med bakgrunnskartet Topografisk Norgeskart Gråtone er klienten i EPSG:32633. Vi velger dette koordinatsystemet fordi også DTM-en er i EPSG:32633, og høydekorreksjonen fungerer bare hvis punktdata og DTM har samme koordinatsystem.

Innholdet består for øvrig av en WMS-tjeneste fra LMT-server/Geoserver for målestasjonene. WMS-tjenesten leser PostGIS-tabellen *mtm.klimastasjoner*, som er et uttrekk fra *public.weatherstations*, hvor lat/long-koordinatene er gjort om til et geometrifelt (*shape*), og transformert til EPSG:32633. Dette gjør at klienten henter data direkte fra LMT-serveren.

Resultatet etter interpolasjon av måleverdier er laget <Stasjonsnavn> i form av en tiff-fil. I den endelige løsningen forutsetter vi at dette laget kan leses inn i Open Layers direkte fra den genererte tiff-filen. Dette har vi imidlertid ikke klart å få til i testen. For å løse dette i prosjektfasen, har vi lagt

<sup>23</sup> <http://wms.geonorge.no/skwm1/wms.topo2.graaton?>

tiff-filen inn i GeoServer og generert en WMS-tjeneste på den, med en tilhørende SLD-fil som inneholder presentasjonsparametere.

Disse tre lagene er deretter satt opp i samme Open Layers-kart. For å demonstrere hvordan dette kan se ut for brukeren, har vi deretter laget en HTML-fil hvor kartet er lagt inn som en IFRAME <sup>24</sup>. I HTML-filen for øvrig har vi lagt inn noen metadata som beskriver kartet og dets formål, hvilke parametere det er bygget på og en enkel veiledning for navigasjon. Vi mente også at det var viktig å forklare hva kartet IKKE er, så vi har i tillegg lagt inn en *disclaimer* som beskriver de gjeldende forbehold.

Ettersom vi ikke har klart å fullføre automatisering av prosessen har vi valgt å sette opp en enkeltstående webkartklient for hver test av prototypen, som skal vise at prinsippene fungerer, selv om vi ikke har fått lagt kartet inn i LMTs stasjonsside. Hovedhensikten vår i denne fasen er å demonstrere prinsipper, ikke å få til et ferdig produkt.

For å kunne dokumentere arbeidet underveis har vi satt opp en testside på serveren <http://lmttest.nibio.no/mtm/>, hvor vi har lagt inn løsningsseksempler etter hvert som prototypen utviklet seg.

---

<sup>24</sup> <http://lmttest.nibio.no/mtm/IFRAME/IFRAME4.html>



## 5 Test av prototype

Vi ville gjerne at noen av de potensielle fremtidige brukerne skulle teste løsningen vår, selv om den ikke foreligger i en fullstendig automatisert versjon. Dette er også i henhold til den overordnede samtalemодellen vi har styrt prosjektet etter. I de tidligere semestrene har vi vært i kontakt med en del landbruksrådgivere, og brukt deres tilbakemeldinger som grunnlag for de avgjørelsene som har ledet frem til prototypen som foreligger. Vi hadde et ønske om at de rådgiverne som tidligere har vist mest interesse og engasjement for løsningen vi har utviklet, skulle få sjansen til å teste den. Derfor tok vi kontakt med fire rådgivere, og spurte om disse ville teste løsningen vår.

### Hypotese for brukertesten:

Vi har fremstilt en prototype for et kart som vi antar at rådgiverne vil bruke til å formidle informasjon om varmesum til sine medlemmer.

Hvis vi stiller brukerne (rådgiverne) spørsmål om hvordan de tolker kartet, og hvilke bruksmuligheter de ser for kartet, kan vi få svar på om kartet/produktet er nyttig og lesbart. Det vil si om brukerne via kartet kan finne svar på de spørsmålene som vi forestiller oss at løsningen skal besvare, og om brukerne kan forstå de budskap vi ville sende via kartet.

### 5.1 Slik utførte vi testen

Løsningen kunne testes av brukerne ved at disse sendte oss input-parametere på e-post, og vi utførte de delvis manuelle analysene og la det endelige resultatet på en nettside, hvis URL ble sendt tilbake til rådgiverne sammen med noen utvalgte spørsmål.

Det var 4 parametere vi ba om i e-posten. Dette er de samme som brukeren ville måtte fylle inn dersom løsningen var helautomatisert. I tillegg oppgav vi litt ekstra informasjon i parentes, for å forklare hva som ville vært valgmulighetene i nedtrekksmenyen dersom dette faktisk hadde vært en interaktiv løsning på en nettside.

- **Foretrukket værstasjon** (løsningen er tenkt slik at man velger en stasjon, og så gjøres beregningene for de omliggende områdene)
- **Basistemperatur**
- **Startdato** (i 2014 eller 2015)
- **Sluttdato** (i samme kalenderår som startdatoen)

Den første brukeren som svarte, ba om data fra Brunlanes stasjon, og tok med hele vekstsesongen 2015. Vi laget følgende kart: <http://lmttest.nibio.no/mtm/IFRAME/IFRAME4.html>. Bruker nummer to som svarte, ba om Landvik stasjon, og ville gjerne se både vekstsesongen 2014 OG 2015. Resultatet vi kom frem til var følgende: <http://lmttest.nibio.no/mtm/IFRAME/Iframe5.html>. Resultatene ligger også i bilag VI og i bilag VII.

Sammen med lenken til kartene, sendte vi også ut noen spørsmål vi ville at brukeren skulle besvare, etter at de hadde sett på og testet kartene vi hadde laget.

Vi brukte relativt mye tid i gruppen på å diskutere hvilke spørsmål vi skulle stille. Vi var opptatt av at det ikke skulle være for mange spørsmål for brukeren, så vi måtte sørge for at vi stilte spørsmål som fortalte oss de viktigste tingene vi ønsket å vite, og at disse spørsmålene var enkle og utvetydige for mottakeren. Dette er spørsmålene vi sendte til de to brukerne:

- *Hva kan du bruke et kart som dette til?*  
Formålet med dette åpne spørsmålet er å få brukerens umiddelbare reaksjoner og ideer. Det kan jo være bruksområder vi ikke har tenkt på?
- *Svarer resultatet på kartet til dine egne forventninger for varmesummen i den valgte perioden?* Her vil vi vite om brukeren anser resultatet som troverdig.
- *Er det lett å tolke varmesummen på en gitt lokalitet ut i fra kartet?* Det primære budskapet i formidlingen er varmesummen uttrykt som tall. Er kartet lesbart nok til at brukeren forstår dette budskapet?
- *Hva tenker du om nøyaktigheten i dataene som vises på kartet?*  
Et sekundært budskap er at verdiene skal leses med en viss forsiktighet, ettersom både interpolasjon og høydekorrigering er foretatt på generaliserte terrenghøyder. Spørsmålet er tenkt å avsløre hvordan brukeren oppfatter dette budskapet.
- *Kan kartet i sin nåværende form brukes til formidlingsformål (for eksempel overfor medlemmer)?*  
Dette er et lukket spørsmål som skal teste vår hypotese som at kartet kan brukes til formidling til medlemmene i landbruksrådgivingen.
- *Har du noen ønsker eller tips til oss?*  
Her kan brukeren skrive kommentarer som ikke er dekket av overnevnte spørsmål

Etter at spørsmålene var sendt, var det en av brukerne som svarte ganske umiddelbart. En annen av brukerne valgte å videresende e-posten med lenke og spørsmål til en lang rekke kollegaer, som ikke kjente forhistorien med hvordan kartet var kommet til, og hva det var meningen at det skulle kunne gjøre/ikke-gjøre. To av kollegaene svarte oss, og svarene vi fikk bar preg av nettopp manglende kjennskap til prosjektet og tanken bak løsningen. Vi kan bare i noen grad nyttiggjøre oss disse svarene. Av de to resterende brukerne som fikk tilbud om å teste løsningen, har en ikke besvart vår henvendelse, og én har svart at han siden sist har skiftet jobb, og dermed ikke lenger befinner seg i vår målgruppe.

## 5.2 Overvåket brukertest

Ideelt sett ville vi hatt en overvåket brukerundersøkelse av løsningen. Observasjon av brukerne når de bruker løsningen, se hvor de klikker og hva de lurer på, hadde vært meget nyttig. Dessverre var det ikke tid eller ressurser til å gjøre en slik omfattende undersøkelse. Utsendelse av e-post var dermed et kompromiss for oss.

### 5.3 Tilbakemeldingene

Som nevnt, er tilbakemeldingene på løsningen så langt meget begrensede, og det er begrenset hvor stor vekt man kan tillegge dem. Løsningen bør nok testes mer før den går «i produksjon» hos NIBIO.

Tilbakemeldingene vi har fått forteller oss likevel at resultatet på kartet ikke umiddelbart ser galt ut. Brukerne kunne ønsket seg en finere oppløsning enn 1x1 km, naturlig nok. De tror likevel kartet kan brukes til formidlingsformål overfor landbruksrådgivingens medlemmer, og en av dem nevnte at det viktigste er å vise variasjonen fra sted til sted, heller enn de nøyaktige tallene. En av brukerne mente kartet var lett å tolke som det var per i dag, mens en annen skulle ønske seg noe endring i fargebruk på stasjonsteksten, og at detaljene på bakgrunnskartet kom bedre frem, da det er disse man bruker for å orientere seg i områdene rundt hver enkelt stasjon.

Et ønske om interpolert nedbørskart ble gjentatt.

Vi har ikke gjort noen endringer på prototypen som følge av tilbakemeldingene vi fikk. Det overlater vi til NIBIO og de som eventuelt skal videreutvikle løsningen i fremtiden.

## 6 Konklusjon

I dette kapittelet vil vi konkludere på de delhypotesene vi stilte opp i kapittel 3-5, og på oppgavens hovedhypotese som er satt opp i innledningskapittelet. I tillegg vil vi av hensyn til leserens oversikt over prosjektets resultater også oppsummere de viktigste resultater og erkjennelser fra de mer tekniske delkapitlene.

### 6.1 Løsningsdesign

Ved å arbeide med problemstillinger vedrørende interpolasjon, høydeforskjell mellom stasjoner og fargeskala til presentasjon av varmesum har vi funnet ut at de stedvis store avstandene mellom stasjonene i LMT ikke er til hinder for å interpolere, og at man kan korrigere for ulikheten i stasjonshøyder ved å omregne alle målinger ned til havnivå, så de blir sammenlignbare. Vi fant dessuten ut at fargelegging av resultatet antagelig best kan gjøres ved bruk av en prosentvis skala som ta høyde for de svært variable verdispenn i resultatene.

Gjennom bruk av designmodellen har vi foretatt en systematisk gjennomgang av løsningsdesign, og via overveielser og valg har vi ført ideen videre frem til et konkret løsningsdesign. Vi tenker at kartløsningen skal plasseres i LMTs eksisterende stasjonssider, hvor den skal gi brukerne mulighet til å få et estimat over varmesummen på steder hvor den ikke er målt. Varmesummen vises som et rasterkart som er fargelagt ut fra varmesumverdien, og med et topografisk Norgeskart som bakgrunn så brukerne enkelt kan orientere seg i forhold til sine egne interesseområder. I tillegg skal brukerne få tilgang til metadata og beskrivelse av løsningen, så de kan danne seg et inntrykk av nøyaktigheten i den viste informasjonen.

Løsningsdesignet er fremstilt i en modell for den logiske dataflyten, og i et utkast til teknisk arkitektur. Den logiske modellen viser en dataflyt hvor data trekkes ut av LMTs database, og hvor det etter en validering av data gjøres en interpolasjon og en etterfølgende høydekorleksjon via bruk av en digital terrengmodell (DTM). Resultatet av denne prosessen presenteres i et webbasert kart i LMT.

Den tekniske skissen viser LMTs eksisterende arkitektur, hvor webklienten via en mulig ny webservice spør etter data fra databasen på LMT-serveren. Data inngår i den påfølgende geoprosesseringen (interpolasjon og høydekorrigering) ved hjelp av en GIS-komponent, og resulterer i et rasterlag som presenteres på et kart i webklienten. Samlet sett vurderer vi at delhypotesen for løsningsdesignet er verifisert.

### 6.2 Tekniske komponenter

Datauttrekk: Webservicen som NIBIO har utviklet bruker SQL-syntaks, og har vist seg å kunne brukes til å trekke ut data fra LMT-databasen i et geografisk format (GeoJSON). Datauttrekket kan hentes direkte inn i de GIS-komponentene vi har valgt til å utføre geoprosesseringen.

Geoprosessering: Vi har funnet frem til en rekke Open Source-komponenter som kan utføre de ønskede beregningene. Det har imidlertid vist seg å være for komplisert for oss å bruke kriging i en automatisk prosess, og i prototypen har vi derfor valgt å gjøre interpolasjonen via Inverse Distance Weighting-algoritmen fra GRASS. Omfang og effekt av NULL-verdier bør undersøkes nærmere og en eventuell filtrering av stasjoner implementeres.

Den påfølgende høydekorreksjonen av resultatet fra interpolasjonen er gjort ved hjelp av r.mapcalculator fra GRASS, og med bruk av en digital terrengmodell (DTM) for hele Norge. Vi har identifisert en rekke mulige feilkilder, deriblant de angitte stasjonshøydene i LMT, som bør undersøkes nærmere i den endelige løsningen.

Presentasjon på kart: Open Layers er trolig egnet til å presentere det resulterende rasterlaget, sammen med et bakgrunnskart og værstasjonene som et vektorlag, i et kart satt inn på en webside. I prototypen hentes kartet fra GeoServer, og fargeleggingen styres via en SLD-fil som også genererer den tegnforklaringen som vises på siden. Vi har funnet beskrivelser av hvordan en normalisert skala kan implementeres, men har ikke fått det til å virke i praksis, og har derfor utformet resultatene fra brukertesten manuelt i hvert enkelt tilfelle. Vi har heller ikke klart å implementere GetFeatureInfo-funksjonen så brukeren kan få opplysninger om celleverdien for en konkret celle. I den endelige versjonen må disse problemene løses, så fargelegging og tegnforklaring genereres automatisk ut fra de aktuelle verdiene i rasterlaget, og slik at brukeren enkelt kan få opplysning om varmesummen for en gitt lokalitet som en tallverdi.

Automatisering av prosessen: Vi har ikke klart å etablere en automatisk kjøring av alle trinnene i den samlede prosessen fra brukerinntil til visning av resultatet på kart. Vi har likevel via eksempler på andre løsninger og råd fra eksperter sannsynliggjort at PyWPS kan brukes til å automatisere prosessen, slik at Open Layers-komponenten som legges inn i klimastatistikksiden i LMT aktiverer et kall mot WPS-tjenesten. Denne sørger videre for å aktivere webservicen for datauttrekk med de brukervalgte parametrene. Datasettet inngår deretter i interpolasjonen og høydekorreksjonen via de valgte algoritmene i GRASS, hvor input og output også styres av PyWPS. Til slutt sørger PyWPS for at resultatet (rasterlaget) fremstilles via MapServer eller en annen WMS-server, og på den måten blir hentet inn som lag i Open Layers-kartet i webklienten.

Samlet konklusjon: Vi har funnet tekniske komponenter og algoritmer fra Open Source-miljøet som kan løse de forskjellige delprosessene i den fullstendige prosessen. På grunn av en rekke mangler i det tekniske oppsettet har vi ikke kunnet teste den fulle funksjonaliteten i de ulike komponentene, og vi kan derfor ikke uttale oss om hvorvidt de valgte komponentene er de best egnede til oppgaven. Delhypotesen om valg av programvare anser vi derfor bare som delvis verifisert.

### 6.3 Brukertest

Brukertesten resulterte i meget få tilbakemeldinger, men de vi fikk viste at brukerne så gode muligheter for å bruke kartet til å formidle varmesummen. Vi kan ikke på bakgrunn av denne testen konkludere med at rådgiverne rent faktisk vil bruke kartet til dette formålet, og prototypen er nok heller ikke helt ferdig nok til å kunne teste denne hypotesen, men vi mener at de tilbakemeldinger vi har fått gir en klar indikasjon på at brukerne ser et potensiale for løsningen.

Når det gjelder lesbarhet, har vi fått tilbakemeldinger fra brukerne som indikerer at bakgrunnskartet kunne ha vært mer synlig for å lette navigeringen, og at fargen på stasjonsteksten kunne vært en annen.

### 6.4 Hovedkonklusjon

Med grunnlag i vår kunnskap om geografisk informasjon, supplert med undersøkelser av spesifikke problemstillinger, har vi laget et løsningsdesign som tar høyde for problemstillingene som er nevnt i kapittel 3. Løsningsdesignet består av et flytdiagram som viser hvordan data hentes fra LMT og

omdannes til informasjon om varmesum vist på et kart i et område rundt en valgt værstasjon, og av en skisse som viser hvilke tekniske komponenter vi forventer at løsningen kommer til å bestå av.

Gjennom uttesting av en rekke tekniske komponenter og funksjoner i Open Source GIS-miljøet har vi funnet frem til en kombinasjon av komponenter som kan løse oppgaven med å hente ut data fra databasen i LMT, og omdanne dem til et varmesumkart på stasjonssidene i LMT. Vi har ikke klart å lage en helautomatisk løsning, men via blant annet QGIS har vi kunnet lage en prototype som på grunnlag av brukerinput produserer de ønskede varmesumkart. På bakgrunn av dette og av studier av andre løsninger og kilder, har vi sannsynliggjort at en sammenhengende og helautomatisk løsning kan lages dersom man har tilgang til den rette utviklingskompetansen. Vi har innsett at vi ikke har kompetanse til å programmere løsningen selv, men vi mener at vi vil kunne lage løsningen i samarbeid med webutviklere.

Vi har kunnet bruke prototypen til å teste det tenkte produktet overfor utvalgte brukere. Tilbakemeldingene fra testen har bekreftet at brukerne fortsatt ser muligheter i løsningen, selv om det enda er en rekke ting som bør tilpasses før endelig idriftsettelse.

Vi har fortløpende dokumentert det vi har funnet ut av underveis, og vi mener derfor at NIBIO har et godt grunnlag til å ferdigutvikle løsningen, dersom de velger å gjøre det.

Flere ganger i prosessen har vi, i dialog med NIBIO, justert på prosjektets målsetninger. Brukerne har ikke vært involvert i prioriteringen på samme måte, siden vi har vurdert at produktet ikke var modent nok, blant annet fordi prototypen stadig inneholder en del manuell betjening.

Sammenlagt vurderer vi likevel, på basis av ovenstående punkter, at hovedhypotesen er verifisert.

## 7 Perspektivering

Gjennom de to foregående semestrene vi har jobbet med denne oppgaven, har vi forsøkt å vurdere behovet og nytten i en løsning med varmesum på kart. Vi undersøkte om det var andre, tilsvarende løsninger som var utviklet andre steder, men fant bare én tjeneste som hadde samme idé som oss.<sup>25</sup>

Vi har fundert litt på hva grunnen kan være til at det ikke er utviklet slike tjenester andre steder i verden. Er det ikke bruk for en slik tjeneste? Er det for vanskelig å få til en god nok nøyaktighet i tjenesten? Er det for vanskelig å lage, eller er det rett og slett ikke gode nok inntjeningsmuligheter for et interaktivt varmesumskart i en større skala?

Vi synes idéen om å bruke GIS til å estimere varmesum på steder hvor det ikke er målt er grunnleggende god, og mener det er ønske og behov for en slik tjeneste hos landbruksrådgiverne (tilbakemeldingene fra disse bekrefter også dette). Hvorvidt løsningen er levedyktig på sikt, vil likevel avhenge av at løsningen blir enkel og intuitiv å bruke, og at visse tekniske utfordringer vi har støtt på, kan løses på en god måte. Til sist må nøyaktigheten i dataene som presenteres være tilstrekkelig god. Gruppen har gjennom arbeidet med løsningen identifisert en rekke steder og muligheter for forbedringer av tjenesten. Kompetanse og ressurser har hindret disse idéene fra å komme med i denne løsningen. Når vi nå overdrar det vi har utviklet til NIBIO, kommer det også med flere idéer til forbedringer, optimaliseringer og utviklingsmuligheter. Vi vil i det følgende skissere noen av disse mulighetene - listen er slett ikke uttømmende. Videre vil vi også reflektere litt omkring hvilke ressurser løsningen krever, hvilken nytte samfunnet kan tenkes å ha av løsningen og hvorvidt vi ser eksportmuligheter for løsningen.

### 7.1 Fremtidsscenarioer for LMTs varmesumskart

Vi ser at det er mange potensielle veier å gå for den/de som ønsker å utvikle denne løsningen videre:

Det finnes et større datatilfang som kan inkluderes i løsningen enn det vi har gjort i vår prototype. Vi har kun basert oss på LMTs 80-85 målestasjoner, men dersom man i tillegg tar med målingene som Meteorologisk institutt gjør, vil man få et mye større datatilgang, og mange flere observasjoner å interpolere mellom. Det er grunn til å tro at dette vil øke dekkningen og forbedre nøyaktigheten i løsningen. De aktuelle dataene fra Meteorologisk institutt er gratis tilgjengelig som en web tjeneste.

På tredje semester lekte vi med tanken på å inkludere nedbør i visualiseringen, men dette ble av praktiske hensyn utelatt fra prototypen vi har utviklet. Vi synes likevel det er en god idé som ble presentert oss av en av rådgiverne vi snakket med på tredje semester. Vår kontakt på Meteorologisk institutt redegjorde for at nedbør er vanskelig å forutse og interpolere (Tveito, pers. medd.), men ettersom vi har nedbørsdata fra alle LMT-stasjonene hadde det gått an å presentere nedbør fra samme valgte periode som en søyle på kartet, og eventuelt også sammenlikne denne med nedbøren fra året før, eller andre statistiske enheter (for eksempel årnormalen). Denne kan også presenteres enkelt som en *label* på det valgte stasjonspunktet. Meteorologisk institutt har dessuten en nedbørsradar på yr.no som viser nedbør i sanntid<sup>26</sup>. Vi har også lekt med en heller umoden idé om at denne på en eller annen måte kunne kombineres med varmesumskartet.

---

<sup>25</sup> <http://uspest.org/cgi-bin/usmapmaker.pl>. Konsultert 11.01.2016.

<sup>26</sup> <http://om.yr.no/forklaring/radar/>. Konsultert 11.01.2016.

Trolig kan selve visualiseringen og den kartografiske presentasjonen av dataene videreutvikles og forbedres dersom man er villig til å legge mer ressurser i det enn vi hadde tilgjengelig i arbeidet med denne løsningen.

Det er enkelte som mener at kriging er den beste interpoleringsmetoden for temperatur (Z Ustrnul, 2005). Vi har av praktiske hensyn valgt å bruke en annen interpoleringsmetode i løsningen som nå foreligger. Hvorvidt kriging gir et resultat med større nøyaktighet, og hvorvidt sensitivitet og andre inputfaktorer i GIS-komponentens utregning burde vært justert for at løsningen skal treffe bedre med sine estimater, er også noe som kan utforskes videre. Trolig kan også resultatet forbedres ved at andre faktorer enn interpolering og høyde trekkes inn i utregningen. Vi vet at det er en rekke faktorer som påvirker temperatur og varmesum, for eksempel avstand til kyst og hav, nærvær av innsjøer, helningsgrad og solinnstråling som følge at terrenget heller i ulik himmelretning (Halling et al 2015b). Vi har ikke tatt hensyn til disse faktorene i vår løsning, men her er det virkelig mange ting å ta tak i for den som interesserer seg for GIS-analyse og/eller meteorologi.

Gjennom vårt arbeid med denne løsningen har vi forstått at det muligens finnes programvare som kan levere bedre responstid og stabilitet i løsningen, enn den programvaren vi valgte å bruke. Vi har i høy grad latt oss styre av NIBIOs ønske om bruk av Open Source programvare. Hadde vi brukt en proprietær og kommersiell plattform, hadde det sannsynligvis vært noe enklere å utarbeide løsningen.

Vi utelukker heller ikke at kodingen kan løses mer effektivt og elegant enn vi selv har fått til, eller at applikasjonsarkitekturen kan optimaliseres. Disse tingene ligger utenfor våre primære fagfelt, og utenfor målsetningen for MTM-studiet. Vi overlater gladelig ballen til noen som har bedre forutsetninger for å ta disse løsningene i fremtiden.

I dag fungerer det slik at man interpolerer og høydekorrigerer med utgangspunkt i en allerede eksisterende LMT-stasjon. Med tanke på hvordan andre kartbaserte online-løsninger fungerer i dag - og hvilke krav som dermed i praksis stilles til online karttjenester, kunne det være hensiktsmessig med flere muligheter for oppslag i kartet. Kanskje skal man kunne søke på stedsnavn? Eller zoome inn i kartet til det området man er interessert i, og så klikke i kartet akkurat der hvor ens egen gård/åker/jorde ligger? Dette klikket kan så være triggeren for at varmesumsestimeringen setter i gang.

Samtidig vet vi også at det finnes mye interessante statistiske data internt i NIBIO, for eksempel jordbunnsinformasjon, økonomisk statistikk om for eksempel lønnsomhet i landbruket (NIBIO, 2015a), og vegetasjonsinformasjon som også allerede finnes på kart, eller kan knyttes til kart. I kombinasjon med et troverdig varmesumskart, åpner dette for at ny forskning kan utføres, og at gode, nyttige og interessante fremtidige kartløsninger kan utvikles.

I tillegg tror vi det ville det være nyttig med en mulighet for enkel eksport av kartbildet til dokument for eksempel som pdf eller jpg. Dette tar vi med som en anbefaling for videreutvikling til NIBIO, men har ikke selv satt det opp i prototypen.



## 7.2 Kostnader og ressurser

Vi har også gjort oss noen betraktninger omkring de økonomiske sidene av dette prosjektet, og en eventuell fremtidig videreutvikling av løsningen. For vår del har dette vært et studieprosjekt. Det innebærer at vi har brukt arbeidsgivers og/eller private ressurser på reiser og egen fritid på utviklingen. Det har gått med riktig mange timer, og vi har brukt ettermiddager, kvelder, helger og høytider på arbeidet. Selv om vi ikke måler denne innsatsen i kroner og øre, har den selvsagt hatt en kostnad. På den annen side er denne tiden en investering for oss i studiegruppen - en investering i kunnskap, et vitnemål og en grad/tittel.

NIBIO har velvillig bidratt med tid og kompetanse inn i vårt prosjekt. De har tilrettelagt og gitt oss tilgang til LMTs servere og databaser, de har kommet med informasjon og svart på våre spørsmål, deltatt i flere Skype-møter, lest rapportutkast, delt sitt kontaktnettverk med oss, og dessuten bekostet våre to reiser til Ås, Norge, i løpet av tiden prosjektet har pågått. Rådgivere, forskere og bønder vi har intervjuet og fått tilbakemeldinger fra gjennom utviklingen av løsningen har brukt av sin tid inn i prosjektet. Her ligger det en kostnad som har inngått i vårt prosjekt.

Etter avsluttet prosjekt vil NIBIO sitte med dokumentasjon, en anbefaling og flere løse idéer for videre implementering av varmesumskartet. Dersom NIBIO velger å fortsette å utvikle løsningen, vil det selvsagt koste enten intern ressursbruk, eller eksterne konsulenttimer. Vi har basert løsningen på Open Source programvare, og det vil trolig derfor ikke påløpe noen investeringskostnader i denne forbindelse. NIBIO har et godt kompetansemiljøer på Open Source-GIS. Trolig skal det relativt lite arbeid til for at de kan implementere løsningen, i sin enkleste form, i LMT.

## 7.3 Samfunnets nytteverdi

Gjennom hele prosjektperioden har vi vært klare på at vi, i tillegg til å gjøre en masteroppgave, ønsket å lage noe som var av nytte og interesse for noen andre. Ved å lage en løsning som viser varmesum på steder hvor denne ikke har blitt målt, håper vi varmesum kan bli brukt av flere enn i dag - og i større grad enn det som er tilfellet i dag. Vi vet at varmesum, i en landbruksmessig kontekst, kan brukes til å estimere plantetidspunkt, høstetidspunkt og tidspunkt for iverksetting av tiltak mot skadegjørere, for eksempel bruk av pesticider, herbicider, tildekking eller andre tiltak.

Ved å bruke kunnskapen om hver enkelt plante eller skadegjørers varmepreferanser, kan man øke avlingenes størrelse, og redusere bruken av sprøytemidler i landbruket. Dette er økonomisk gunstig for bonden som kan tjene mer penger på en større avling, og samtidig spare penger ved å bruke mindre sprøytemidler. Ved at det brukes mindre sprøytemidler sparer man også miljøet, det blir mindre rester liggende i jordsmonn som eventuelt kan medføre avrenning til nærtliggende vann og på den måten komme inn i økosystemer og næringskjeder. Ved mindre bruk av sprøytemidler blir det også mindre produktrester i avlingen, hvilket gir sunnere produkter for forbrukeren.

Dersom resultatet som vises i løsningen vår holder god nok kvalitet, vil dette bidra til at NIBIOs måledata foredles og får enda større rekkevidde og bruksområde enn de har i dag. Det finnes per i dag ingen andre aktører som tilbyr varmesum, slik LMT gjør i dag, og heller ingen som tilbyr en tjeneste som estimerer varmesum på kart, slik vi gjør i vår løsning. Vårt håp er samtidig også at landbruksrådgiverne skal kunne utføre sin jobb bedre når det gjelder rådgiving av medlemmer ved å bruke kartet i visualiseringer og rapporter til medlemmene.

Den økonomiske reelle økonomiske verdien og potensialet av disse prosessene er vanskelig, om ikke umulig, å tallfeste.

#### 7.4 Eksportpotensiale

NIBIO er eier av løsningen som vi nå overdrar til dem. De kan nå selv gjøre som de vil, ønsker og ser mest hensiktsmessig med idéen vi har utviklet, komponentene vi har funnet frem til, og koden som ligger til grunn for tjenesten. Dette betyr at løsningen kan skrotes - eller den kan videreutvikles og eventuelt også eksporteres til andre land. Vi kjenner til at NIBIOS tjeneste VIPS (som er kort nevnt i innledningen) har blitt tatt med til andre land i Europa, og etter hvert som den meteorologiske infrastrukturen også er på plass her, kan også idéen implementeres andre steder. Land som allerede har målestasjoner, har enda kortere vei. Vi har ikke funnet veldig mange tilsvarende eller konkurrerende tjenester - her kan det potensielt være muligheter.

## 8 Refleksjon og læring

### 8.1 Hvordan gjøre ting man ikke kan

Prosjektet med å utvikle et varmesumskart for LMT har vært meget lærerikt. Vi har valgt å lage en løsning som det trolig ikke finnes maken til i hele verden (= på hele internettet), men som består av flere individuelle prosesser som hver for seg, eller i andre kombinasjoner, utføres i mange sammenhenger rundt om i verden.

Det er ingen tvil om at det har vært lærerikt å stå foran en uoverskuelig oppgave, hvor ingen av oss hadde den kompetansen som skulle til, for å løse oppgaven slik vi gjerne ville.

Hva gjør man så – hvis det for eksempel skal programmeres en komponent som skal automatisere arbeidsgangen? En mulighet er å sette seg ned og lære seg et programmeringsspråk fra bunnen av. Det gir kanskje den grundigste forståelsen, men det innebærer også en risiko for at man bruker mye tid på å lære mye man egentlig aldri får bruk for i prosjektet. En annen mulighet er å prøve å finne ut om noen andre har gjort noe liknende, og så se på deres erfaringer.

Problembasert læring handler om å oppsøke og innhente den nødvendige kunnskap for å løse det konkrete problemet man står overfor. Det gir kanskje ikke den bredeste innsikten, men incentivet er til gjengjeld ganske stort for å lære nøyaktig det det er behov for. Det er dessuten også større sjanse for at den tilegnede kunnskap fester seg, når man har en knagg å henge det på, og har brukt det i et konkret eksempel eller oppgave. Dersom man har klart å bruke for eksempel et programmeringsspråk til en spesifikk oppgave, virker det ikke uoverkommelig å bruke det i en annen oppgave på et senere tidspunkt.

Vår læring i dette prosjektet er selvsagt bygget på toppen av den plattformen vi allerede hadde, fra tidligere oppgaver og kurs på studiet, samt individuelle erfaringer fra arbeidslivet. Det var for eksempel sånn at da vi første gang så syntaksen i NIBIOs PHP-scripts, kunne vi kjenne igjen SQL-strukturen, og begynte å se andre mønstre i koden. På den måten kunne vi, trinn for trinn og med kun enkelte unntak, resonnerer oss frem til hva de enkelte deler av koden resulterte i.

Innhenting av ny kunnskap har i høy grad skjedd via ulike kilder på nettet, hvor vi blant annet har søkt på diverse fora for utviklere og administratorer. Ettersom vi har brukt «kjente» komponenter i løsningen vår, var det heldigvis slik at nesten alle problemer vi støtte på, var det andre som hadde slitt med før oss, og hjelp var å finne på nevnte fora. Det finnes mange hjelpsomme og kunnskapsrike personer der ute, som er villige til å bruke tid på å besvare spørsmål fra mindre rutinerne brukere/utviklere. Dette er særlig nyttig når man sitter med konkrete utfordringer i et konkret programmeringsspråk.

Flere fora har egne systemer for å indikere hvilke av svarene som vurderes å ha størst verdi. Det er både slik at enkelte svar får *ratings*, samtidig som brukere av forumet opptjener ulike former for *credits* når de gir svar mange brukere har nytte av. I programmeringsverdenen er det en fordel at ekspertenes kunnskaper raskt kan testes i praksis, og hvis deres input har den ønskede virkning, er det ingen grunn til å granske motivene bak utsagnet nærmere. Det kan selvsagt være mer eller mindre hensiktsmessige måter å løse et problem på.

Når det gjelder løsningsarkitekturen har det vært vanskelig å finne gode kilder på internett, og vi har i stor grad vært nødt til å støtte oss på utsagn fra fagpersoner vi kjente fra før, eller som vi har møtt underveis i prosjektet, som veilederne på Aalborg Universitet eller utviklere hos NIBIO.

## 8.2 Hva forventet vi å lære – og hva lærte vi faktisk?

Vi gikk i gang med dette prosjektet med et ønske om at det faktisk skulle foreligge en funksjonell og helautomatisert prototype som resultatet når prosjektet var ferdig. Underveis i prosessen fikk vi bekreftet at ting ikke er så enkelt som man ved første øyekast kanskje skulle tro - og at ting tar lenger tid enn man først fikk beregnet. Likevel skal man ikke undervurdere hva man faktisk kan få utført, også av ting man ikke egentlig kan, når man først går i gang. Metoden som er nevnt over, med å innhente eksempler fra nettet for på den måten å tilegne seg kunnskaper om for eksempel programmering, viste seg å fungere til en viss grad. *Tutorials* i skrift og video-format, samt fora kan til en viss grad være nok til å sette i gang arbeidet. Derimot så vi at det ikke var nok til å komme i mål med prosjektet. Vi opplevde flere ganger at eksemplene på nett ikke virket - i alle fall ikke i vårt script- og at ingen av løsningene som andre brukere har skissert kunne løse problemet. I noen tilfeller gikk vi da til ressurspersoner vi visste kunne programmeringsspråket bedre enn oss selv, og fikk hjelp der. Likevel opplevde vi å ikke komme helt i mål med for eksempel PyWPS-scriptet som skulle fullautomatisere løsningen i prototypen. Et eller annet manglet i vårt script, slik at det ikke virket som vi håpet. Hva dette var, har vi ikke klart å identifisere. Dersom vår kunnskap på dette området hadde vært bedre, ville dette kanskje ikke vært et problem. Det vi derimot har lært av denne prosessen, er at nettopp det å ha personer som innehar den riktige kompetansen er et viktig suksesskriterium. Vår rolle som prosjektstyrere bør i høy grad ligge i å definere og spesifisere kravene til de enkelte komponenter, og så kan selve kodingen settes bort til noen som har den riktige spisskompetansen. Trolig ville prosjektet vårt sett ganske annerledes ut i dag, om dette hadde vært en reell mulighet.

## 8.3 Hensiktsmessighet

Hvor hensiktsmessig kan man så si at denne strategien for løsning har vært? Vi kom langt - men ikke helt i mål. Metoden vi valgte å bruke har vært tidkrevende og til tider frustrerende. Mye tid har gått med til å google og lete etter svar på internett-fora, til å gjøre feil, til å forsøke å rette dem opp, uten at dette virket. Likevel er det grunn til å tro at gitt hvordan situasjonen var, var dette måten å gå frem på. MTM-studiet legger opp til problembasert læring - og det var nettopp det dette var. Videre er jo også den store målsetning at vi skulle lære "noe" – å lage en funksjonell løsning er egentlig sekundært i det store bildet. Og har vi lært dette "noe" på den mest hensiktsmessige måten? Vi har i alle fall lært veldig mye av måten vi har jobbet på. Gruppens medlemmer kunne lite eller ingenting om disse tingene før prosjektet gikk i gang, og nå har vi faktisk klart å få på plass en løsning som virker - med litt manuell hjelp. Læringskurven har vært bratt. Vi har definitivt gjort "ting-vi-ikke-kan" og i stor grad har det gått bra. Om ikke annet, forteller dette oss at vi har lært hvordan vi finner ut av ting, og hvordan dette kan operasjonaliseres i en større sammenheng - med eller uten hjelp fra utenforstående. Vi har lært oss hvor og hvordan vi skal lete, og vi kan nok om konteksten til at vi kan gjøre oss nytte av det vi finner og leser. I tillegg har vi lært noe om hvordan man spesifiserer og utarbeider løsningsprinsipper som andre så kan programmere for oss. Dette er viktig kunnskap å ta med videre i arbeidssituasjoner innen geoinformatikk. Å vite hvordan man tilegner seg kunnskap, og evne å se kunnskapen som legoklosser, hvor man kan bygge bit for bit oppå hverandre for å nå målet man ønsker. Dette er også en betydelig del av læringsprosessen.

## 8.4 Samarbeidet i gruppen og samarbeidet med andre

Samarbeidsformen og –metodene i gruppen har blitt innarbeidet gjennom de to foregående semestrene hvor dette prosjektet har pågått. Vi kjenner hverandre, vet hva vi kan og ikke kan, og hvordan vi best kan få samarbeidet vårt til å fungere. Allerede på tidligere semestre har vi blitt enige om, og funnet ut at vi har noenlunde samme målsetning og vilje til innsats i prosjektet. Dette har medført at vi hadde en god stemning i gruppen i det vi gikk i gang med arbeidet med oppgaven på fjerde semester. På det første møtet på fjerde semesteret, som ble avholdt tidlig i august 2015, snakket vi dessuten litt om hvilke forventninger vi hadde til det kommende arbeidet, og hvor vi anså at våre egne styrker og svakheter i forhold til det som lå foran oss var.

## 8.5 Rollefordeling

Gruppen har opplevd at arbeidet med prosjektet har vært markant forskjellig dette semesteret, sammenliknet med de foregående to semestrene. Dette skyldes naturlig nok at vi nå har kommet inn i den praktiske delen av prosjektet – selve «utførelse» som det kalles i Brodersens (2014) samtalemodell. Tidligere har prosjektet i stor grad bestått av planlegging og vi har brukt mye tid på å velge hva prosjektet vårt skulle omfatte – og hva det IKKE skulle omfatte. Nå har tiden kommet til faktisk å utføre. Vi opplever at gruppens medlemmer har ulike kompetanser og forutsetninger for de forskjellige delene av prosjektet. Dette er gruppens styrke, samtidig som det har medført en ganske tydelig rollefordeling mellom gruppens medlemmer. I løpet av arbeidet har vi fordelt de ulike oppgavene, og de ulike rollene, mellom oss. Én av oss har satt seg inn i PHP og webservices, og bearbeidet tjenesten vi fikk fra NIBIO. Et gruppemedlem hadde fra tidligere god erfaring med bruk av PostgreSQL i sitt arbeid, og fikk naturlig nok oppgaven med å gjøre den delen av prosjektet som hørte hjemme her. Det siste gruppemedlemmet fikk oppgaven med å sette seg inn i Open Layers og sørge for den presentasjonsmessige sistedelen av prosjektet. Vi anser det som naturlig at arbeidet i gruppen fordeles på denne måten, det er en måte hvor man kan få bruke tid til å sette seg inn i en ting om gangen, og finne ut av hvordan nettopp denne delen av prosessen foregår. Gruppen har hatt jevnlig møter på Skype og hatt flere arbeidshelger sammen, hvor man i tillegg har vært fysisk samlokalisert når man har jobbet med disse tingene. Flytskjemaet som er beskrevet tidligere har vært rettleidende under arbeidet, og vi har brukt det til å avtale at et gruppemedlem jobber med uttrekket, mens en annen jobber med interpoleringen. Slik har vi ikke mistet oversikten over prosessen og vissheten om hvilken sammenheng den har med andre elementer i prosjektet, selv om ulike personer har jobbet på ulike deler av prosessen, noen ganger også i ikke-kronologisk rekkefølge.

## 8.6 Metode

På andre og tredje semester brukte vi samtalemodellen som prosjektstyringsverktøy og retningsgiver for arbeidet. Vi skjønte tidlig at det ikke ville være like hensiktsmessig å navigere utelukkende etter denne planen i den siste delen av prosjektet. På søken etter en ny metode å basere arbeidet på, snuste vi innom agil-metodikk. Vi bestemte oss for å la oss inspirere av disse metodene, og bruke det vi mente var mest hensiktsmessig for oss i det arbeidet vi skulle løse. Vi ville ta med oss tanken om fleksibilitet, og har tilpasset oss underveis i prosessen, selv om vi ikke brukte metoden formelt. Metodikken som ble tatt i bruk i stor grad var egenutviklet – og den utviklet seg underveis. Et flytskjema som vi først tegnet opp for at vi selv skulle forsikre oss om at vi hadde husket på alle de sentrale elementene som skulle inngå, og at vi hadde en mest mulig lik forståelseshorisonst viste seg å kunne ha et prosjektplanleggingspotensiale når den ble kombinert med en ganske enkel tabell utformet i Microsoft Word. Denne tillot oss å holde oversikt over hvilke deler vi var ferdige med, og hva som gjensto å gjøre. Både tabell og flytskjema var stadig et tema på våre ukentlige prosjektmøter

og ble revidert og endret mange ganger i løpet av semesteret. Gruppen forventet nok at det skulle være et lite element og innslag av uoversiktighet i denne prosessen. Det har det også vært, men vi har likevel opplevd at vi gjennom å sette oss ned sammen, hente frem skjemaet og henvise til boksene, har kunnet gjenvinne oversikten over prosjektplanen. I slutten av prosjektet så vi likevel at det var selve flytskjemaet vi hadde best bruk for, og at tabellen, selv om idéen kanskje ikke var så dum, ble for omstendelig og omfattende å fylle ut. Vi var stadig på etterskudd med utfyllingen, og det var stort sett aldri fullstendig oppdatert. Tabellen ble etter hvert lagt bort, og arbeidets fremdrift ble i stedet dokumentert i gruppens ukentlige møterefater, i tillegg til egne dokumentasjonsnotater som hver enkelt utarbeidet underveis i prosjektet.

## 9 Kildekritikk

I dette kapitlet reflekterer vi omkring hvilke kilder vi har brukt i arbeidet med dette prosjektet og denne rapporten, og hvordan dette har påvirket resultatet. NIBIO og landbruksrådgiverne har vært kilder for oss i dette arbeidet, slik de også var i arbeidet på de to foregående semestrene. På grunn av at prosjektet har vært i forskjellige faser på de forskjellige semestrene, og dette fjerde semester har vært en utviklingsfase, synes vi likevel vi ikke har forholdt oss til disse kildene på helt samme måte som tidligere. Dette vil vi beskrive mer i detalj i dette kapitlet. Vi kommer også til å nevne kort hvordan vi har brukt måledataene fra LMT som input i løsningen, før vi går litt nærmere inn på bruken av internett – og spesielt ulike webfora – i dette prosjektet. Det er her kildebruken skiller seg aller mest ut fra andre og tredje semester.

### 9.1 NIBIO

Som det fremkommer av trepartsmodellen i kapittel 2.2, er NIBIO som oppdragsgiver å regne i vårt prosjekt. Likevel har vi brukt to av NIBIOs ansatte som kilder til informasjon om LMT som tjeneste, og for å få kunnskap om tjenestens brukere i de tidligere semestrene. Vi har også kunne bruke dem til å luften og få tilbakemelding på idéer vi har hatt underveis. Idéen om å interpolere og høydekorrigere varmesum mellom stasjoner kom i samspill med NIBIO. Dette har vi beskrevet i tidligere rapporter. Denne gangen har NIBIO også bidratt med tilrettelegging av testserver, utvikling av webservice og tilgang til databasen. Vi har også diskutert systemarkitektur med NIBIOs webutvikler, og fått tips og gode råd derfra. Dette har helt klart vært et viktig bidrag, og har påvirket den endelige løsningen når det gjelder valg av komponenter. Et eksempel på dette er at vi kun har vurdert Open Software-programvare, og at vi valgte løsningen med webservice som henter data fra LMT-databasen basert på brukerens input.

### 9.2 Forholdet til landbruksrådgiverne

Landbruksrådgiverne har vært involvert i noe mindre grad denne gangen enn de har på de foregående semestrene. I dette fjerde semesteret har vi kun forholdt oss til rådgivere vi har vært i kontakt med før, og som har vist interesse og positivitet mot vårt prosjekt. Dette begrenser selvsagt utvalget, og også tilbakemeldingen som kom. Likevel oppfatter vi at på dette tidspunktet i prosjektprosessen hadde vi mer nytte av tilbakemelding fra nettopp disse, da de bruker både LMT, varmesum og nettbaserte kartløsningen allerede, og kan lett ha en oppfatning om løsningen. Denne gangen var tilbakemeldingen vi fikk i stor grad styrt av spørsmålene vi sendte ut sammen med løsningen. Det var bare to rådgivere som ønsket å delta i uttestingen av kartet og tre stykker som sendte tilbakemelding<sup>27</sup>. Tilbakemeldingene har altså vært få, men de har vært viktige for prosjektet og fremdriften av prosjektet i henhold til den overordnede samtalemодellen.

### 9.3 LMT som datakilde

Vi har forholdt oss til måledataene LMTs nettverk av målestasjoner leverer uten noen nærmere kontroll, ettersom NIBIO og LMTs brukere kontrollerer dataene regelmessig. Som beskrevet i denne rapporten oppdaget vi at det kunne være unøyaktigheter i angivelsen av høyde som stasjonene sto plassert på. Dette er data som har ligget i LMTs database som metadata, men som ikke har blitt brukt i beregninger eller noe tilsvarende før nå. Når det gjelder tjenesten vi har utviklet i dette prosjektet

---

<sup>27</sup> Egentlig var det fire, men den ene tilbakemeldingen kan vanskelig kalles seriøs eller konstruktiv. Vi har derfor valgt å se bort fra denne.

har høyden over havet plutselig mye å si for hvordan nøyaktigheten i det endelige resultatet blir. Dette har vi redegjort for i kapittelet om høydekorrigering.

#### 9.4 Nettbaserte kilder

En av de største forskjellene i kildebruk på dette semesteret sammenliknet med de foregående er vår bruk av kilder fra internettet. Vi har brukt internett til å lete etter Open Source GIS-komponenter gjennom engelskspråklige søk som for eksempel: "*open source interpolation kriging*", eller ved å spørre personer i våre nettverk som har erfaring med denne typen software. Fordelen med sistnevnte metode er at vi kan starte en dialog om softwaren og at disse personene har konkrete erfaringer med hvordan den virker. På internett er det litt vanskeligere å gå i dialog med kildene, selv om det nesten alltid er noen som har hatt samme spørsmål før – og fått svar som i beste fall kun trenger mindre bearbeiding for å passe i ens eget prosjekt. Det er selvfølgelig varierende grad av nytte i det man finner. Noe av grunnen til dette er nok at det er skrevet av og for utviklere på bakgrunn av kunnskaper, forutsetninger og/eller komponenter vi ikke nødvendigvis har. Vi har opplevd at det kan være en ganske høy terskel for å fullt ut kunne utnytte potensialet som bor i slike fora. På den andre siden er det heller ingen kvalitetskontroll av det som ligger der. Hvem som helst kan i prinsippet svare på spørsmålene, legge ut "tips" og "gode" råd. Ansvar legges dermed på den som velger å bruke tipsene som ligger der. Selve testen er relativ enkel: virker det, eller virker det ikke? Dersom tips og kode man finner i disse foraene faktisk virker, er det nok ikke mange som tenker noe videre over problemet, selv om det jo kan være bedre, eller mer effektive måter å gjøre det på. Det vet man ikke på daværende tidspunkt.



## 10 Litteratur

Brodersen, L., 2008. *Kommunikation med kort*. 1 red. Aalborg: Nyt Teknisk Forlag.

Brodersen, L., 2014. *Informationsledelse*. Aalborg: Lars Brodersen.

Det Norske Meteorologiske Institutt, 2015. *eklima.no*. [Internett]

Available at: [www.eklima.no](http://www.eklima.no)  
[Funnet 30 september 2015].

Fu, P. & Sun, J., 2011. *Web GIS - Principles and applications*. Redlands, Cal: ESRI Press.

GeoJSON Format Specification, The, 2015. *The GeoJSON Format Specification*. [Internett]

Available at: <http://geojson.org/geojson-spec.html>  
[Funnet 24 november 2015].

GeoServer Team, The, 2015. *The SLD cookbook*. [Internett]

Available at: [http://docs.geoserver.org/latest/en/user/downloads/raster\\_brightnessandcontrast.sld](http://docs.geoserver.org/latest/en/user/downloads/raster_brightnessandcontrast.sld)  
[Funnet 27 Desember 2015].

Halling, L., Andersen, C. & Nordskog, K., 2015a. *Klima på kart. Geokommunikasjon i Landbruksmeteorologisk Tjeneste (LMT)*, Aalborg: Aalborg Universitet.

Halling, L., Andersen, C. & Nordskog, K., 2015b. *Varmesum og nedbørsmengde på kart. Geokommunikasjon i Landbruksmeteorologisk Tjeneste (LMT)*., Aalborg: Aalborg universitet.

Hengl, T., 2009. *A Practical Guide To Geostatistical Mapping*, Luxembourg: The Office for Official Publications of the European Communities.

Kartverket, 2015. *Geonorge.no*. [Internett]

Available at: <https://kartkatalog.geonorge.no/metadata/kartverket/dtm-50/e25d0104-0858-4d06-bba8-d154514c11d2>  
[Funnet 04 Januar 2016].

Køltzow, M. A. Ø. et al., 2014. *Temperatur-varsler på yr. Hvordan produseres de og hvor gode er de?..* [Internett]

Available at: <http://om.yr.no/2013/12/10/temperatur-varsler-pa-yr-hvordan-produseres-de-og-hvor-gode-er-de/>  
[Funnet 11 Januar 2016].

McBride, B., 2015. *Query a database table or view and return the results in GeoJSON format, suitable for use in OpenLayers, Leaflet, etc..* [Internett]

Available at: <https://github.com/bmcbride/PHP-Database-GeoJSON>  
[Funnet 17 Oktober 2015].

NIBIO, 2015a. *Driftsgranskingar*. [Internett]

Available at: <http://www.nibio.no/tema/driftsgranskingar>  
[Funnet 05 Januar 2016].

NIBIO, 2015b. *Om LMT*. [Internett]

Available at: <http://lmt.nibio.no/about>  
[Funnet 06 Januar 2016].

Ninyerola, M. e. a., 2007. *Objective air temperature mapping for the Iberian Peninsula using spatial interpolation and GIS*, s.l.: International journal of Climateology,7:1231-1242.

osgeo.org, 2015. *GRASS-Wiki*. [Internett]

Available at: [https://grasswiki.osgeo.org/wiki/Large\\_raster\\_data\\_processing](https://grasswiki.osgeo.org/wiki/Large_raster_data_processing)

[Funnet 11 November 2015].

Reeve, D. E. & Petch, J. R., 1999. *GIS, Organizations and People - A socio-technical approach*. London: Taylor & Francis.

Tobler, W., 1970. A computer movie simulating urban growth in the Detroit region. *Economic Geography*, 46, pp. 234-240.

Ustrnul, Z. & Czekierda, D., 2005. *Application of GIS for the development of climatological air temperature maps: an example from Poland*, s.l.: Meteorol. Appl. 12, 43-50.

## Personlige referencer

Fuglsang, Morten. *Ekstern lektor*, Institut for Planlægning, Aalborg Universitet, København.

Nystuen, Ingvild. *Sjefsingeniør, Leder for seksjon Geomatikk*, NIBIO, Ås.

Skog, Tor-Einar. *Senior Software Developer*, NIBIO, Ås.

Tveito, O. E. *Seksjonsleder, Seksjon for Klimadata og romlig analyse*, Klimatjenesteavdelingen Meteorologisk institutt, Oslo.

## Bilag

### I. Ordforklaring

Term	Forklaring
Bounding Box (BBOX)	Avgrensning av et område som angis ved hjelp av hjørnekoordinater
CSS	Cascading Style Sheets: Syntaks for stil- og presentasjonsparametere til bruk i HTML
Daggrader	Summen av registrerte døgnmiddeltemperaturer som ligger over en gitt basistemperatur.
DTM	Digital TerrengModell: Rasterfil hvor celleverdien er lik cellens høyde over havet.
EPSG	European Petroleum Survey Group: Standard for koordinatsystem-referanser.
Feature Info	Informasjon om attributtene til et objekt i en kartfil, enten raster eller vektor. Inngår som en standard del av WMS-spesifikasjonen.
GDALOGR	GDAL: Geospatial Data Abstraction Library. Dataformater for raster- og vektordata, samt et funksjonsbibliotek for dataprosessering. Basert på åpen kildekode.
GeoJSON	Åpent standardformat for fremstilling av geodataobjekter. Basert på JSON, med geometriobjekter i tillegg.
GeoTiff	Datastandard som tillater georeferering av tiff-filer
GRASS	Geographic Resources Analysis Support System: Åpent kildekode-GIS.
iFRAME	Inline Frame. Kan plassere et HTML-dokument inn i en annen HTML-frame.
JavaScript	Scriptspråk som er mye brukt i webutvikling
JSON	JavaScript Object Notation. Standard for datautveksling. Alternativ til XML.
Label	I GIS-sammenheng en metode for påskrift av attributtverdier i kart
LMT	LandbruksMeteorologisk Tjeneste. NIBIOs system for å skaffe meteorologiske data til landbruket.
Metadata	Data-om-data: Overordnet informasjon om et kart, et datasett og lignende.

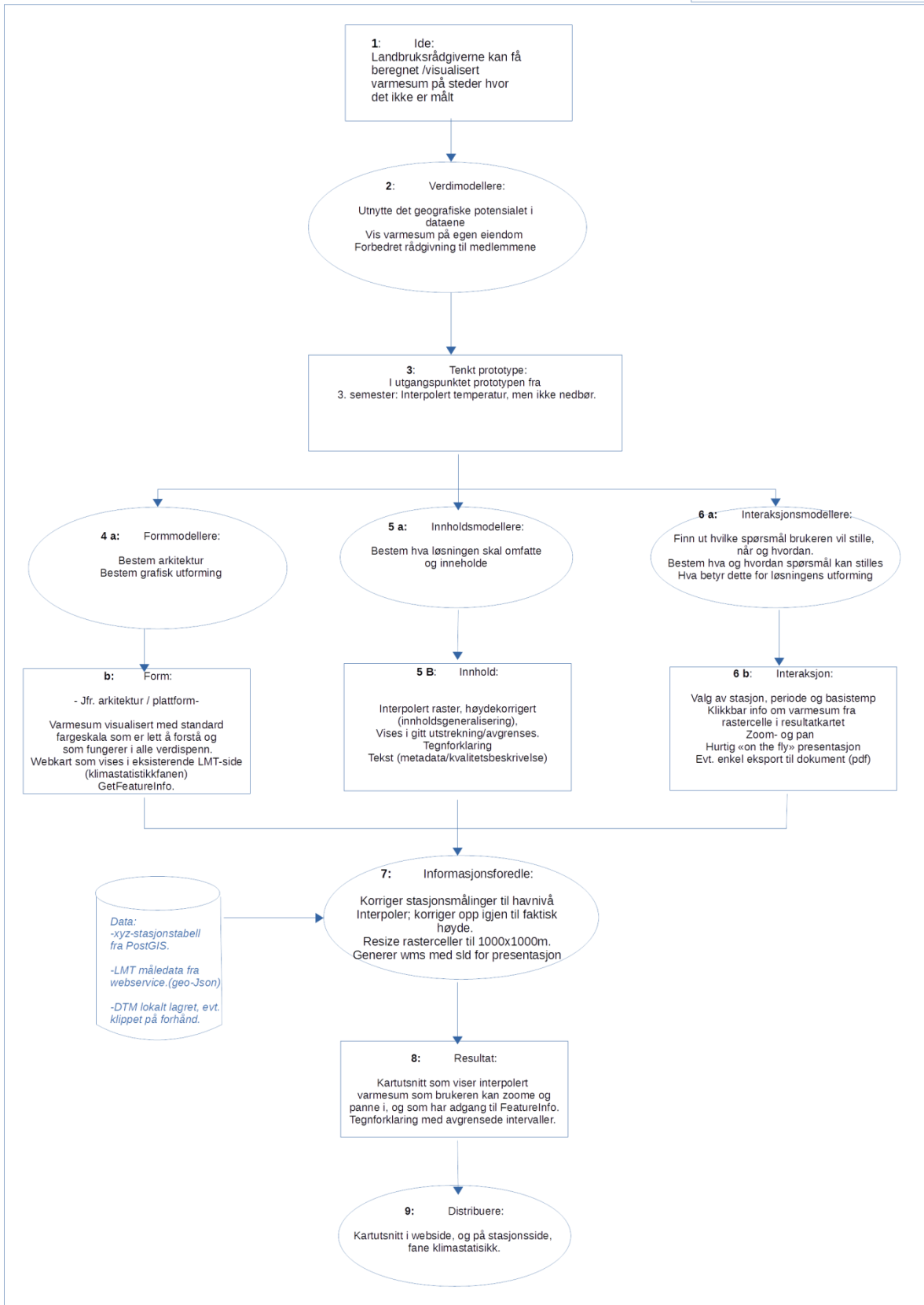
Mock-up	Modell av et design eller system, laget for å demonstrere utforming og virkemåte.
NIBIO	Norsk Institutt for Bioøkonomi
Open Layers	Åpent JavaScript-bibliotek for å håndtere kartdata i et webgrensesnitt
Open Source	Åpen kildekode: Fritt tilgjengelig for alle
PHP	Åpent kildekode-programmeringsspråk
PostGIS	GIS-utvidelse til PostgreSQL
PostgreSQL	Åpent og fritt databasesystem
Process wrapper	Program som kan kalle og styre prosesser i sammenheng med bakgrunnsprosesser.
PyWPS	Prosesswrapper for Python og Web Process Services
SLD	Styled Layer Descriptor. OGC-basert syntaks for oppsett av visningsparametere i blant annet GeoServer.
Spatial View	Et dynamisk utvalg fra databasetabeller, satt sammen til et datasett, og som inneholder geometri.
SQL	Structured Query Language. Spørre- og programmeringsspråk mot databaser.
Varmesum	Se Daggrader
WPS	Web Processing Service. OGC-standard for å utføre webbaserte prosesser.

## II. Designmodellen

Designmodell: Masterprosjekt i MTM ved Aalborg universitet 2014-2016.

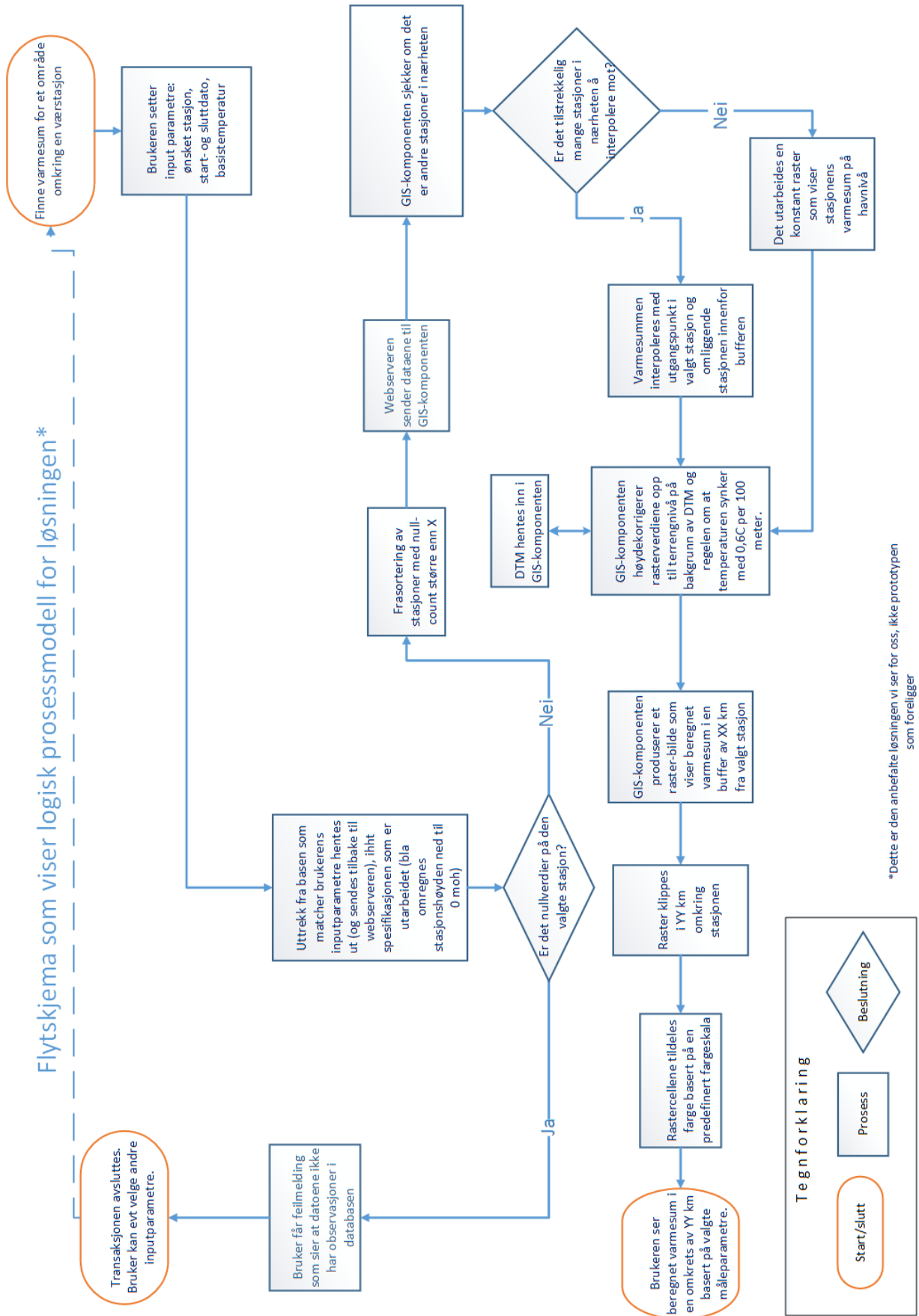
### Varmesum på kart

Gruppe 3:  
Carsten Andersen, Kjersti Nordskog,  
Lene Halling



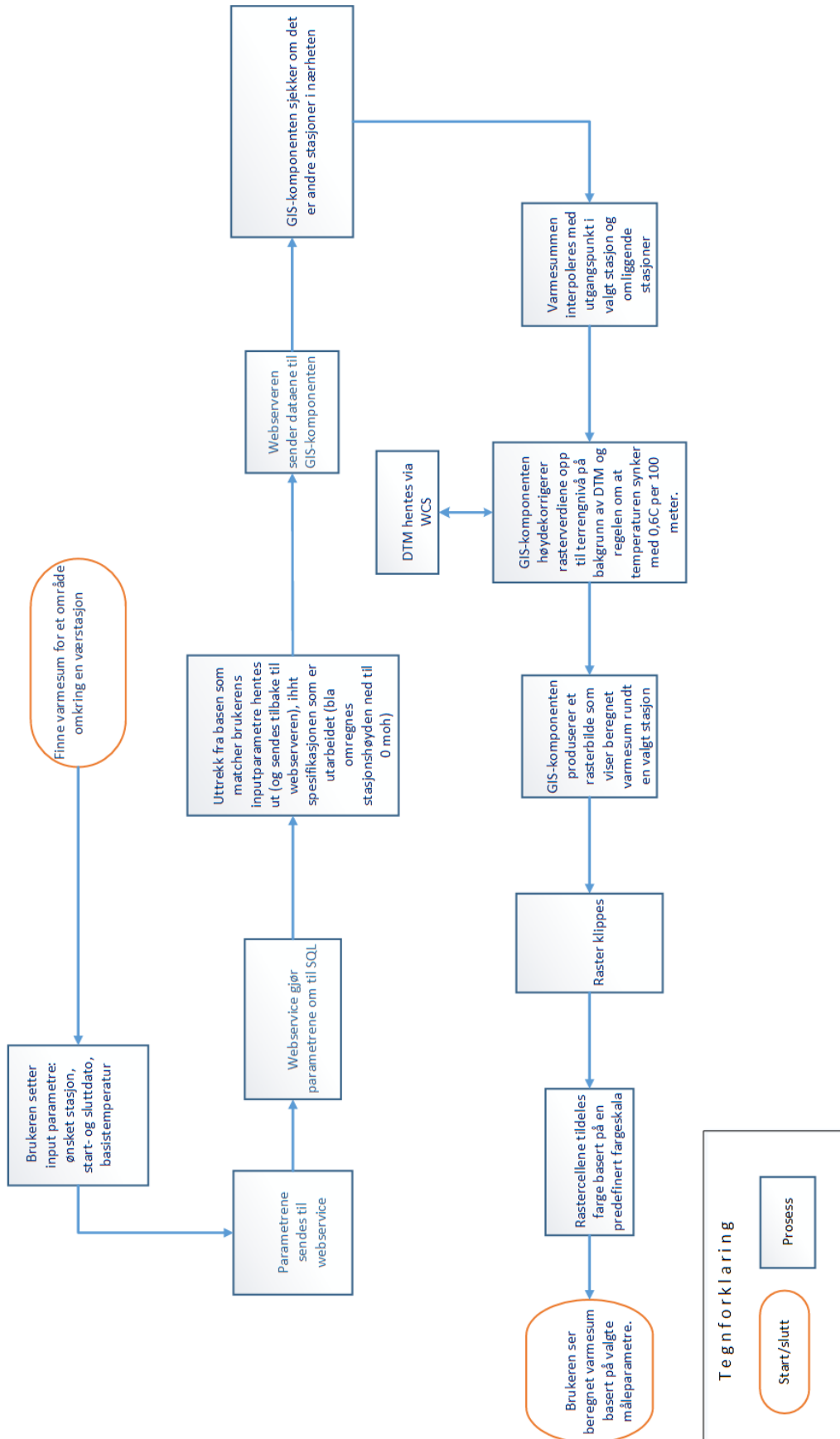


#### IV. Flytdiagram som viser hvordan et endelig resultat kan se ut



V. Flytdiagram som viser prosessen for den foreliggende prototypen

Flytskjema som viser logisk prosessmodell for prototypen





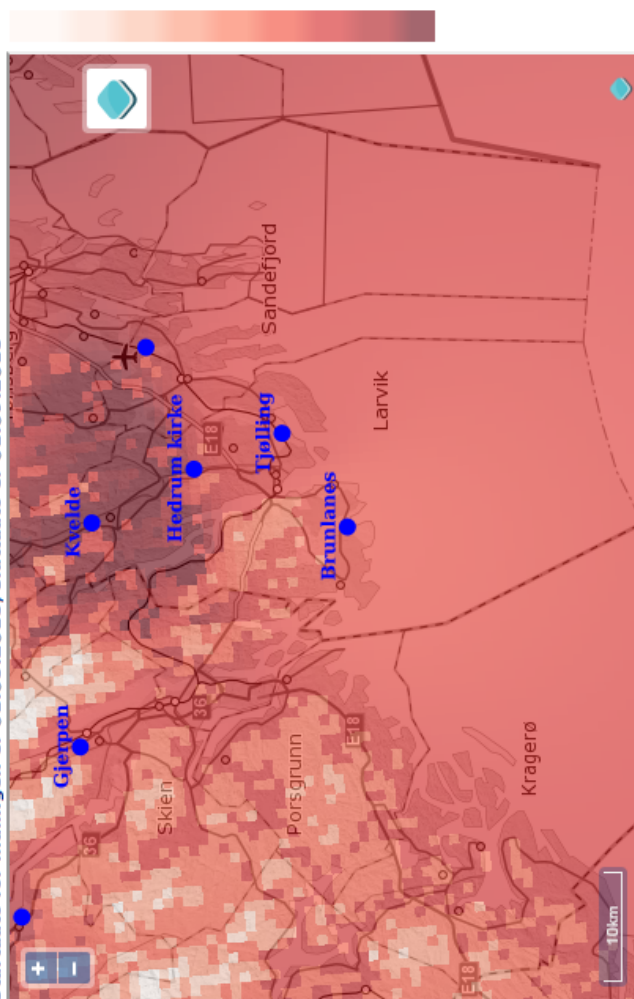
## VI. Prototype brukertest for Brunlanes

### Test av varmesumkart for Brunlanes

Dette kartet er en demonstrasjon av muligheten for å vise interpolerte varmesummer i et kart. Kartet er primært beregnet på landbruksrådgivere, til å kunne finne varmesum for områder mellom målestasjonene, der det ikke finnes målte data.

**Basistemperatur er satt til 6,0 grader**

**Startdato for målingen er 01.05.2015, slutt dato er 01.09.2015**



662 gradddager

706 gradddager

750 gradddager

794 gradddager

838 gradddager

883 gradddager

927 gradddager

968 gradddager

1002 gradddager

### Nøyaktighetsnivå i kartet

Varmekartet er korrigert for høydeverdi, slik at temperaturen reduseres med 0.6 grader pr. 100 m stigning.

Kartet er ikke korrigert for nærhet til vann eller hav, eller i forhold til helingsretning på terrenget.

Vi planlegger en funksjon som viser eksakt varmesum for et gitt punkt ved klikk i kartet, men denne er dessverre ikke på plass enda

**Rull mushjulet framover for å zoome inn, bakover for å zoome ut, og hold venstre mustast inne og flytt musa for å flytte kartbildet.**

**Bakgrunnskartet blir mer detaljert etterhvert som man zoomer innover**

Kartet viser beregnet varmesum for et område rundt Brunlanes klimastasjon. Varmesum / gradddager er summen av den døgnmiddelmiddeltemperatur som er over en valgt basistemperatur i løpet av gitt periode.

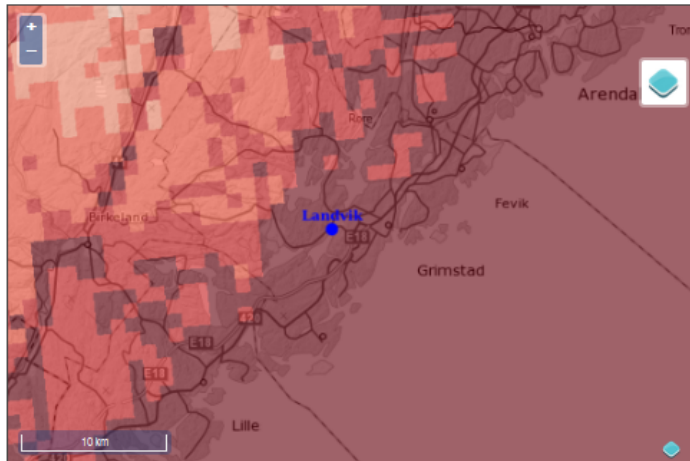
## VII. Prototype brukertest for Landvik

### Test av varmesumkart for Landvik for 2014 og 2015

Dette kartet er en demonstrasjon av muligheten for å vise interpolerte varmesummer i et kart. Kartet er primært beregnet på landbruksrådgivere, til å kunne finne varmesum for områder mellom målestasjonene, der det ikke finnes målte data.

#### Basistemperatur er satt til 6,0 grader

#### Varmesum for 2014



#### Nøyaktighetsnivå i kartet

Varmekartet er korrigert for høydeverdi, slik at temperaturen reduseres med 0,6 grader pr. 100 m stigning.

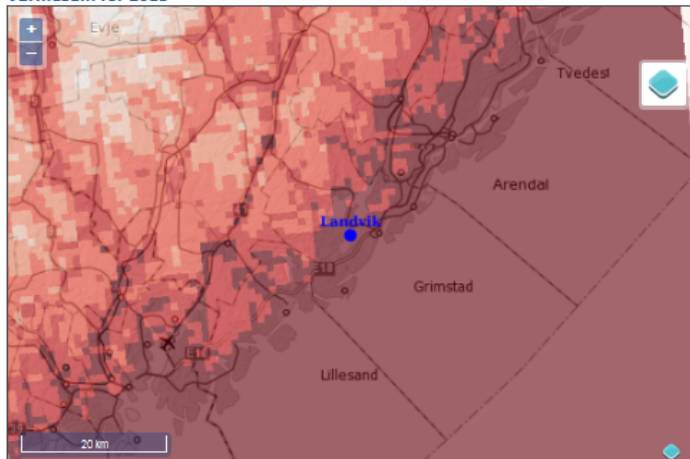
Kartet er ikke korrigert for nærhet til vann eller hav, eller i forhold til helningsretning på terrenget.

Vi planlegger en funksjon som viser eksakt varmesum for et gitt punkt ved klikk i kartet, men denne er dessverre ikke på plass enda

**Rull mushjulet framover for å zoome inn, bakover for å zoome ut, og hold venstre mustast inne og flytt musa for å flytte kartbildet. Bakgrunnskartet blir mer detaljert etterhvert som man zoomer innover**

Startdato for målingen i 2014 er 20.04.2014, sluttdato er 25.10.2014

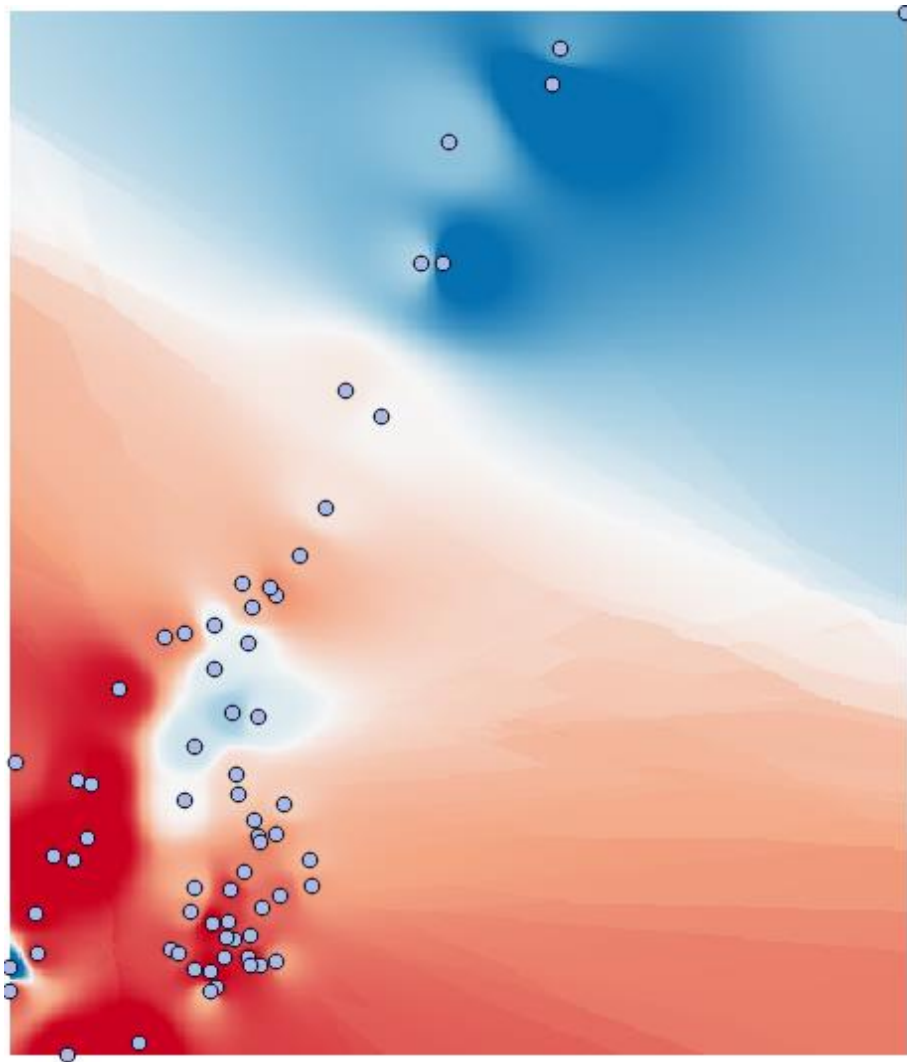
#### Varmesum for 2015



Startdato for målingen i 2015 er 20.04.2015, sluttdato er 25.10.2015

Kartet viser beregnet varmesum for et område rundt Landvik klimastasjon. Varmesum / graddager er summen av den døgnmiddeltemperatur som er over en valgt basistemperatur i løpet av gitt periode.





*"Vi finner det hermed bevist at det er kaldere i Nord-Norge enn i Sør-Norge."*