



Pia E. Tønjum

**Teknisk dokumentasjon av det
årlege realregnskapets
FAME-databaser og rutiner**

Notater

INNHold

1. INTRODUKSJON	4
1.1 Formål og brukerkategorier	4
1.2 Programmeringsstandard	4
1.3 Oversiktsdiagram	6
2. FAME-DATABASER FOR ÅRLIG REALREGNSKAP	7
2.1 NR-REA	7
2.1.1 Overføring av tall fra ORACLE til FAME	7
2.1.2 Innhold i databaser	10
2.2 KNR-REA	12
2.2.1 Omkoding NR - KNR	12
2.2.2 Overføring av tall fra ORACLE til FAME	12
2.2.3 Innhold i databaser	15
2.2.4 KNR-aggregater	15
2.2.5 Metadata	16
2.3 LØNN OG SYSSELSETTING, NR- OG KNR-NIVÅ	17
2.3.1 Overføring av tall fra flat fil til FAME	17
2.3.2 Innhold i databaser	18
2.3.3 Navnestruktur	19
2.4 SYSSELSETTING FORDELT PÅ KJØNN OG UTDANNING	20
2.4.1 Innhold i databaser	20
2.4.2 Navnestruktur	20
2.5 KAPITALSLIT OG KAPITALBEHOLDNING	21
2.5.1 Overføring av tall fra flat fil til FAME	21
2.5.2 Innhold i databaser	22
3. AGGREGATER	22
3.1 Generelt	22
3.2 Navnestruktur	24
3.3 Type aggregater	24
3.3.1 «Rene» næringsaggregater	24
3.3.2 Mellomaggregater	24
3.3.3 Aggregater for næring og produkt/art	27
3.3.4 Aggregater for art	27
3.3.5 Konsumaggregater	28
3.3.6 Andre avledete aggregater	28
3.4 Hvordan definere nye aggregater	29

4. METADATA	30
4.1 Generelt	30
4.2 Attribute desc_source	31
4.3 Organisering av metadata	32
4.3.1 Tekst til variabler og verdsett	32
4.3.2 Tekst til detaljerte næringer og produkter	32
4.3.3 Tekst til aggregater	33
4.3.4 Vedlikehold/endring av metadata	33
5. TABELLER	34
5.1 Hovedprinsipp	34
5.2 Tabelloppsett	35
5.3 Analysetabeller	36
5.3.1 Hovedrutiner	36
5.3.2 Hjelperutiner	37
5.3.3 Tabellpakker	37
5.3.4 Eksempel på bruk	38
5.3.5 Vindusform	39
5.4 Publiseringstabeller	40
5.4.1 Argumenter til tabellrutinen	40
5.4.2 Tabellpakker	41
5.4.3 Eksempel på bruk	41
5.4.4 Vindusform	42
5.4.5 Konvertering til SGML	42
6. REFERANSER	44
7. VEDLEGG	46
1. Ferdige tabelloppsett med NR-aggregering	46
2. Ferdige tabelloppsett med KNR-aggregering	47
3. Brukerveiledning for uttak av analysetabeller	48
4. Brukerveiledning for uttak av publiseringstabeller	49
5. Hvordan definere nye aggregater - et eksempel	50
DE SIST UTGITTE PUBLIKASJONENE I SERIEN NOTATER	53

1. Introduksjon

Første utkast til denne rapporten ble skrevet av *Eirin Pettersen* i 1996. Kapitlet om organisering av nasjonalregnskapets FAME-databaser er tatt ut, da temaet omtales i et nyere notat (Amdal 1998). I kapitlene om NR-REA og KNR-REA er både tekst og oversiktstabeller noe revidert, da rutinene måtte deles opp som følge av flytting av ORACLE-databasen til en annen server. Kapitlet om lønn og sysselsetting er derimot helt omarbeidet, og det er i tillegg skrevet et nytt kapittel om sysselsetting fordelt på kjønn og utdanning (KNR-nivå). Kapitlet om kapitalslit og realkapitalbeholdning er også nytt. Dette kapitlet er skrevet av *Steinar Todsen*. I resten av rapporten er det også foretatt enkelte endringer.

Det er laget en egen dokumentasjon for FAME-databaser og rutiner for det kvartalsvise nasjonalregnskapet, se Pia E. Tønjum (1999). Det er også laget en egen dokumentasjon for rutiner for FAME-databaser med kjedede års- og kvartalstall og omregningsformler for valgfritt referanseår, se Pia E. Tønjum (1999). Se for øvrig Nils Amdals oversikt over alle FAME-databaser som nasjonalregnskapet har pr. i dag (1998). For en generell innføring i UNIX og FAME vises til Abelsæth (1999), Søberg (1995) og Vogt (1994 og 1997).

1.1 Formål og brukerkategorier

Denne rapporten er en teknisk dokumentasjon av rutiner og tidsseriedatabaser opprettet i FAME for det årlige realregnskapet.

Først beskrives produksjonsdatabaser for NR-REA og KNR-REA. Deretter beskrives innhold og rutiner for produksjonsdatabaser for lønn og sysselsetting på NR- og KNR-nivå. Det gis også en oversikt over rutiner samt innhold i produksjonsdatabaser for kapitalslit og realkapitalbeholdning på NR- og KNR-nivå. Det er forøvrig lagt hovedvekt på hvilke rutiner som er utviklet til å oppdatere databasene, aggregere, legge inn metadata samt ta ut tabeller.

Notatet forutsetter at man har grunnleggende FAME-kunnskaper, og er først og fremst beregnet på de som skal kjøre og vedlikeholde rutiner. Andre som bruker nasjonalregnskapstall, og som ønsker bedre oversikt over hva databasene inneholder, kan imidlertid også ha nytte av å lese deler av notatet. Det forutsettes videre at man kjenner strukturen i realregnskapet. Forhåpentligvis er innholdet oversiktlig og 'praktisk nyttig'. Kapitlet om aggregater, herunder hvordan man definerer nye aggregater, forutsetter at man har noe programmeringserfaring.

1.2 Programmeringsstandard

Hvis ikke noe annet er oppgitt, ligger alle filene som er nevnt i dette notatet på kataloger under \$NASJREGN/tidsserier/. \$NASJREGN er «kortversjonen» av navnet på fellesområdet til seksjon 210, det fulle navnet er /ssb/lynx/a2/nasjregn/.

All overføring fra Oracle-databaser gjøres vha. SQL¹-script, mens aggregering, oppdatering av metadata og generering av tabeller gjøres i FAME. Styringsrutiner er laget som UNIX-script. FAME-prosedyrer og funksjoner starter med dollartegnet «\$» i navnet. Alle aggregerings- og metarutiner er laget som prosedyrer, mens mange prosedyrekall for aggregering, oppdatering av metadata og gen-

¹ Structured Query Language

erering av tabeller er samlet i inputfiler. Vindusrutiner for uttak av analysetabeller eller publiserings-tabeller krever at man har logget seg på via Xvision. Alle andre rutiner kan også kjøres via annet grensesnitt.

Under avsnittet om aggregater blir det mye snakk om formler og tidsserier. Under følger derfor en definisjon på begge FAME-objekter:

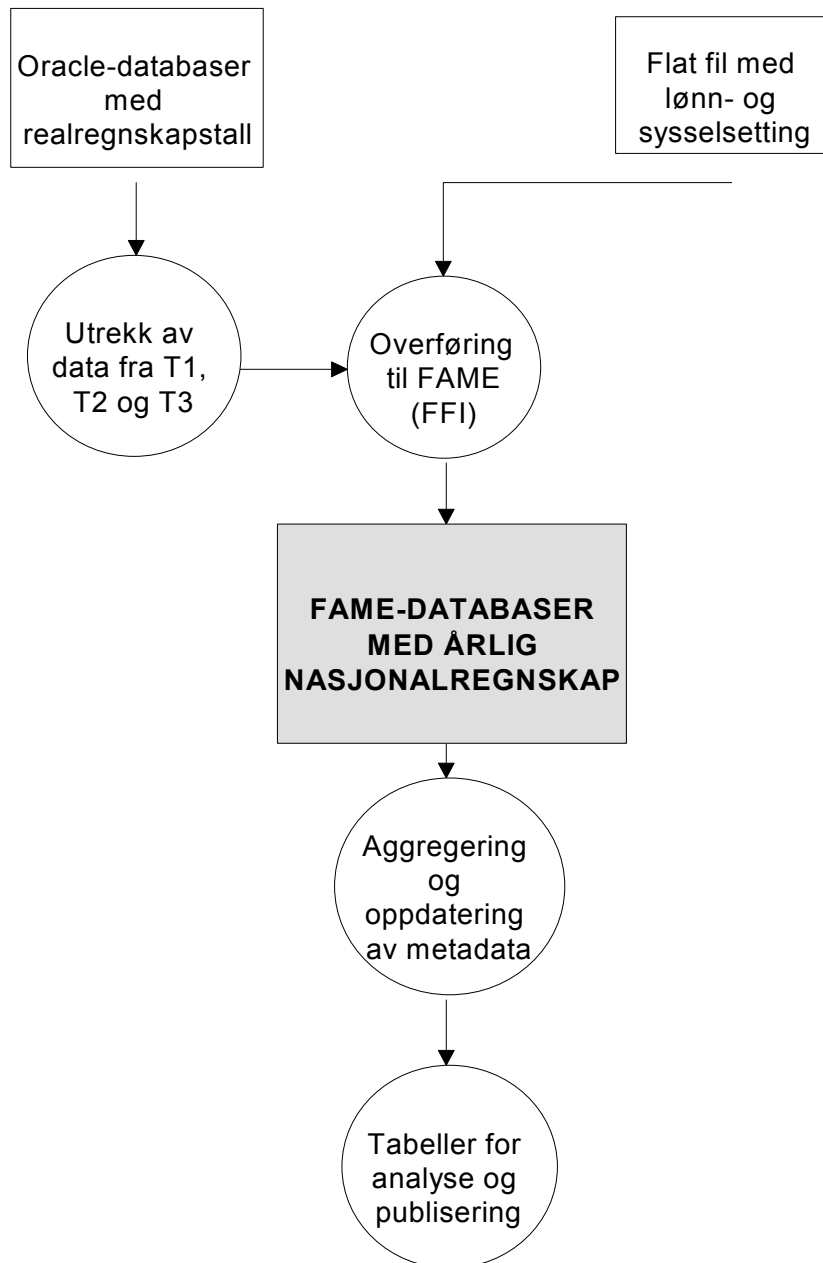
1. **Tidsserie:** Serier som er indeksert med dato og som inneholder en verdi. Kan sette innholdet lik hva man vil. Eksempel: $series\ bnpb.nr2301_vr = bnpb.nr23010.vr + bnpb.nr23014.vr$

Dersom seriene $bnpb.nr23010.vr$ og $bnpb.nr23014.vr$ endrer verdi vil ikke automatisk innholdet i $bnpb.nr2301_vr$ bli endret. Innholdet ligger fast i serien inntil neste gang man oppdaterer $bnpb.nr2301_vr$

2. **Formel:** Inneholder en matematisk sammenheng mellom serier, skalarer og andre formler. Eksempel: $formula\ koh.nr6_vr = koh.nr61_vr + koo.nr64_vr + koo.nr65_vr + koi.nr66_vr$

Skriver man ' $disp\ koh.nr6_vr$ ' blir formelen utført, og verdien som ligger i de underliggende serier og/eller formler blir brukt for å beregne en verdi for $koh.nr6_vr$. Dersom serier som inngår i formelen endrer verdi vil automatisk innholdet i formelen bli oppdatert.

1.3 Oversiktsdiagram



Figur 1-1 Oversiktsdiagram for NR-REA og NR-SYSS

2. FAME-databaser for årlig realregnskap

Dette kapitlet omhandler produksjonsdatabaser for NR-REA, KNR-REA, NR-SYSS, KNR-SYSS, sysselsetting fordelt på utdanning etter KNR-nivå, og kapitalslit og realkapitalbeholdning på NR- og KNR-nivå. Det beskrives hvordan man overfører data fra Oracle og flate tekstfiler til FAME, oppdaterings- og aggregeringsrutiner samt hva databasene inneholder og oppbyggingen av serienavn i FAME.

2.1 NR-REA

2.1.1 Overføring av tall fra ORACLE til FAME

Det er kun tall på detaljert nivå som overføres fra ORACLE til FAME. Aggregatene i nr02.db må lages ved hjelp av aggregeringsrutiner i FAME.

- Det er veldig viktig at man alltid oppdaterer aggregater samtidig som man overfører nye data fra ORACLE. Dette sikres ved at man kjører hele overføringen vha. UNIX-scriptene **nrtofame_1** og **nrtofame_2** (det opprinnelige scriptet er delt opp i to forskjellige script fordi man ikke får tilgang til FAME fra cervus-serveren). Dette er styringsrutiner som starter rutiner for overføring av data fra ORACLE til UNIX, oppdaterer FAME-databaser vha. FFI, aggregerer og oppdaterer metadata.

For å trekke tall ut av ORACLE-databasen må man logge seg på maskinen cervus og gå til katalogen \$NASJREGN/tidsserier/oracle/. Dette kan evt. gjøres med 'rlogin cervus' (eller 'rlogin -l bruker-ident cervus') fra en annen server.

For å trekke ut tall fra T1 eller T2 i løpende priser, faste priser eller T3 kjøres UNIX-scriptene **nrtofame_1** og **nrtofame_2**. Nrtofame_1 kjøres fra cervus, mens nrtofame_2 kjøres fra en server der FAME er tilgjengelig (micmac eller frisch). Overføringen gjelder løpende priser, faste priser og T3.

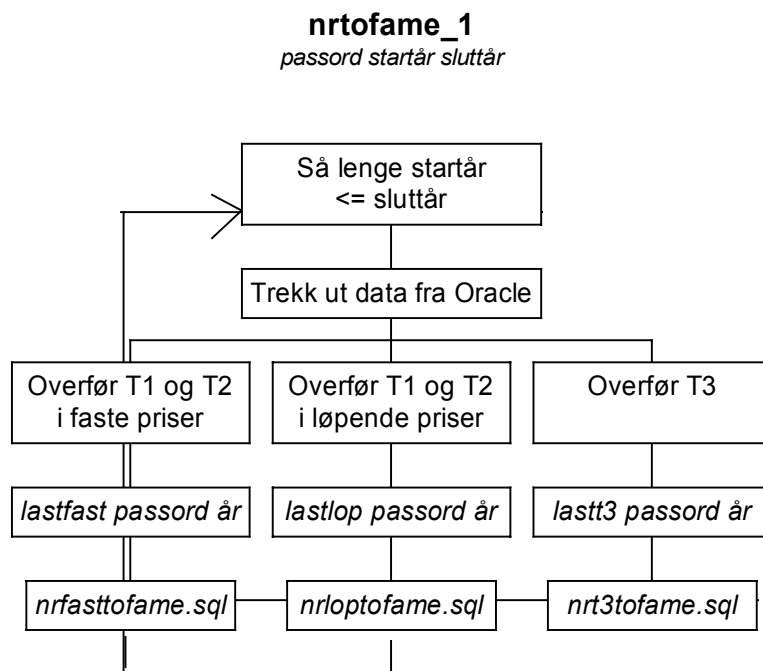
Nrtofame_1 starter opp underrutinene lastlop1, lastfast1 og lastt3_1 for hvert år som skal overføres. Deretter kjøres nrtofame_2, som starter opp underrutinene lastlop2, lastfast2 og lastt3_2 for hvert år som skal overføres. Nrtofame_1 starter opp SQLPLUS og kjører SQL-scriptene nrloptofame.sql, nrfasttofame.sql og nrt3tofame.sql som trekker ut tall for et år om gangen. Resultatet er ASCII-filer med navnestruktur lop/fast/t3+ år +.dat, eks. lop88.dat, fast90.dat eller t389.dat. UNIX-scriptet nrtofame_2 kjører overføringsprogrammet FFI, som lager/oppdaterer tidsserier i FAME-databasen nr01.db med innholdet i dat-filen. For nærmere beskrivelser av aggregerings- og metarutiner vises til kapittel 4 og 5. Inputfilene fiktiv_set.inp (fiktiv_set_nr02.inp på aggregert nivå) lager enkelte fiktive serier for 29-konti.

Input til nrtofame_1 er passord, startår og sluttår for overføring (kan være lik startår) og argument for om metadata skal oppdateres i noen av databasene. Argument for nrtofame_2 er startår og sluttår for overføring (kan være lik startår) og argument for om metadata skal oppdateres i noen av databasene. Metadata kan være lik «nr01.db» for oppdatering av metadata i nr01.db (tar tid!), «nr02.db» for nr02.db, «begge» for begge databaser og «ingen» dersom ikke metadata skal oppdateres. Vær for øvrig oppmerksom på at **årstall for manglende tidsserier i faste priser er hardkodet, og må ev. rettes i lastfast1 og lastfast2**. Eksempel på forskjellig input til nrtofame_1 og nrtofame_2:

Tabell 2-1 Parametre til nrtofame 1 og nrtofame 2

Mulige parametre	Resultat
nrtofame_1 <i>passord</i> 88 92	Henter ut data for 88-92 fra Oracle til tekstfiler
nrtofame_2 88 92 ingen	Overfører 88-92, oppdaterer ikke beskrivelser
nrtofame_2 92 93 nr01.db	Overfører 92-93, oppdaterer beskrivelser i nr01.db
nrtofame_2 88 93 nr02.db	Overfører 88-93, oppdaterer beskrivelser i nr02.db
nrtofame_2 88 89 begge	Overfører 88-89, oppdaterer beskrivelser i begge databaser

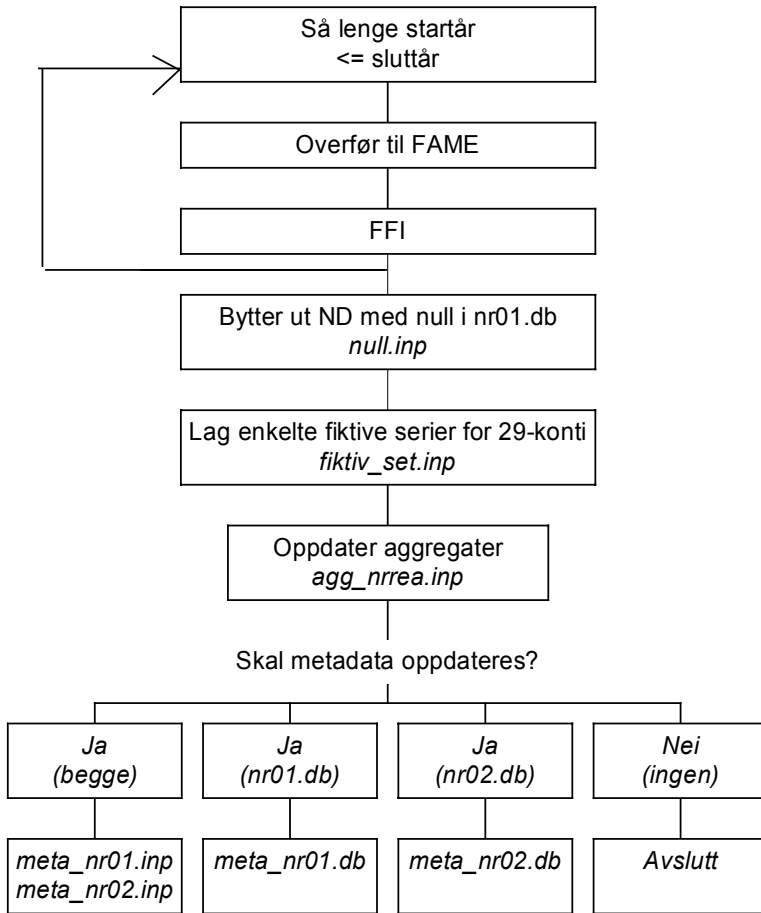
Figurene nedenfor viser hvilken rekkefølge rutiner kalles opp i².



Figur 2-1a. Styringsrutine nrtofame_1

² Første del av scriptet nrtofame har endret navn til nrtofame_1, og henter ut data til fil fra Oracle. Overføring til FAME vha. FFI og FAME-rutinene kalles opp av nrtofame_2. Scriptet er delt fordi FAME ikke er tilgjengelig fra cervus (hvor man må være pålogget for å få tilgang til Oracle). Ellers er alt uendret.

nrtofame_2
startår sluttår metadata



Figur 2-1b. Styringsrutine nrtofame_2

Tabell 2-2 Oversikt over rutiner for overføring av NR-REA

Navn på rutine	Kaller opp rutine	Beskrivelse
nrtofame_1	lastfast1, lastlop1, lastt3_1	Hovedrutine (del 1) som kaller opp SQL-rutiner for overføring av data fra Oracle til tekstfil (startes fra cervus)
nrtofame_2	lastfast2, lastlop2, lastt3_2, null.inp, fiktiv_set.inp, fiktiv_set_nr02.inp, agg_nrrea.inp, meta_nr01.inp, meta_nr02.inp	Hovedrutine (del 2) som kaller opp FFI-rutiner for overføring, FAME-rutiner for aggregering og oppdatering av metadata (startes fra micmac eller frisch)
lastlop1	nrloptofame.sql	Kaller opp nrloptofame.sql og spesifiserer Oracle-database, år og passord.
lastlop2	FFI	Kaller opp FFI. Tar år som parameter
lastfast1	nrfasttofame.sql	Kaller opp nrfasttofame.sql og spesifiserer Oracle-database, år og passord.
lastfast2	FFI	Kaller opp FFI. Tar år som parameter
lastt3_1	nrt3tofame.sql	Kaller opp nrt3tofame.sql og spesifiserer Oracle-database, år og passord.
lastt3_2	FFI	Kaller opp FFI. Tar år som parameter
nrloptofame.sql		Utfører SQL-kommandoer mot Oracle-databaser for å trekke ut tall i løpende priser fra T1 og T»
nrfasttofame.sql		Utfører SQL-kommandoer mot Oracle-databaser for å trekke ut tall i faste priser fra T1 og T2
nrt3tofame.sql		Utfører SQL-kommandoer mot Oracle-databaser for å trekke ut tall fra T3
FFI		Flat File Interface (dataoverføring til FAME)

Område for SQL- og overføringsrutiner: *\$NASJREGN/tidsserier/oracle/*

Område for aggregerings- og metarutiner: *\$NASJREGN/tidsserier/meta/*

Område for datafiler: *\$NASJREGN/tidsserier/oracle/famedata/*

2.1.2 Innhold i databaser

Det er foreløpig opprettet to FAME-databaser for det årlige realregnskapet. De ligger på *\$NASJREGN/tidsserier/NR/DB/*:

nr01.db (som inneholder tall på detaljert nivå)

nr02.db (som inneholder aggregater)

FAME-databasene inneholder tidsserier for utvalgte realregnskapsstørrelser. Navn på tidsseriene er laget etter prinsipper i FAME Level 1. De består av tre hoveddeler:

1. Variabeltype (Konsum, eksport osv.)
2. Klassifikasjon (Leverandør eller mottaker og eventuelt produkt)
3. Prissett (Verdi, volum, prisindeks osv.)

Navnene kan bestå av tre eller fire ledd, avhengig av om klassifikasjonsleddet spesifiserer produkt eller ikke. Punktum brukes som skille mellom leddene.

Eksempel på navn er:

PIN.NR23111.VR og *EKS.NR51110.NR211110.VR*

I øyeblikket inneholder basene tall i løpende og faste priser for følgende variabler:

Tabell 2-3 Variabler i NR-REA

Variabel	Beskrivelse
PROP	Bruttoproduksjon i produsentverdi, etter næring
PROB	Bruttoproduksjon i basisverdi, etter næring
PIN	Produktinnsats, etter næring
BNPP	Bruttoprodukt i produsentverdi, etter næring
BNPB	Bruttoprodukt i basisverdi, etter næring
KOH	Konsum i husholdningene, etter formål
KOO	Konsum i offentlig sektor, etter formål
KOI	Konsum i ideelle organisasjoner, etter formål
BIF	Bruttoinvestering i fast kapital, etter næring og art
BIL	Bruttoinvestering i lager, etter kategori og produkt
EKS	Eksport, etter eksportkategori og produkt
IMP	Import, etter importkategori og produkt
LOKO	Lønnskostnader
ARBTP	Arbeidsgivers trygde- og pensjonspremier
NTA	Næringsskatter
NSU	Næringssubsidier
PTA	Produktskatter
PTAI	Produktskatter, import
TOLL	Toll
MOMS	Merverdiavgift
INTA	Investeringsavgift
PSU	Produktsubsidier
DRIB	Brutto driftsresultat
DEP	Kapitalslit
DRI	Driftsresultat

2.2 KNR-REA

2.2.1 Omkodning NR - KNR

Det er laget tekstfiler til ORACLE-tabeller som viser omkodning fra NR til GRL (modellhovedbok) og GRL til KNR. Aggregering fra NR til KNR gjøres via GRL.

Omkodings-filene ligger på område *\$NR/sqlload/nrrea/loppris/knr/*:

Tabell 2-4 Filer for omkodning NR-KNR

Navn på Oracle-tabell	Innhold
cpa_grl.cat	Kobler NR-produkt med GRL-produkt
produkt_knr.cat	Kobler GRL-produkt med KNR-produkt
nace_grl.cat	Kobler NR-næring med GRL-næring
sektor_knr.cat	Kobler GRL-næring med KNR-næring

NB: Husk at absolutt alle endringer i kontoplan for NR, GRL eller KNR må oppdateres i omkodingskatalogen. Logg på cervus. Etter oppretting legges de inn i ORACLE-bruker **koder_sna** i SNAN-databasen med scriptet loadknr.bat. Scriptet ligger på område *\$NR/sqlload/nrrea/loppris/sqlload/*.

Gi kommandoen: **loadknr.bat koder_sna/passord@SNAN**

I tillegg må nødvendig granting for å få tilgang til grl/knr-tabeller sikres. Scriptet *x:\210\snant\sqlscript\grantgrl.sql* må kjøres på bruker **koder_sna**, slik:

plus80 koder_sna/sqlsna@SNAN @grantgrl oraclebruker
hvor oraclebruker er brukeren du ønsker å gi lesetilgang, f.eks. *sna77*

2.2.2 Overføring av tall fra ORACLE til FAME

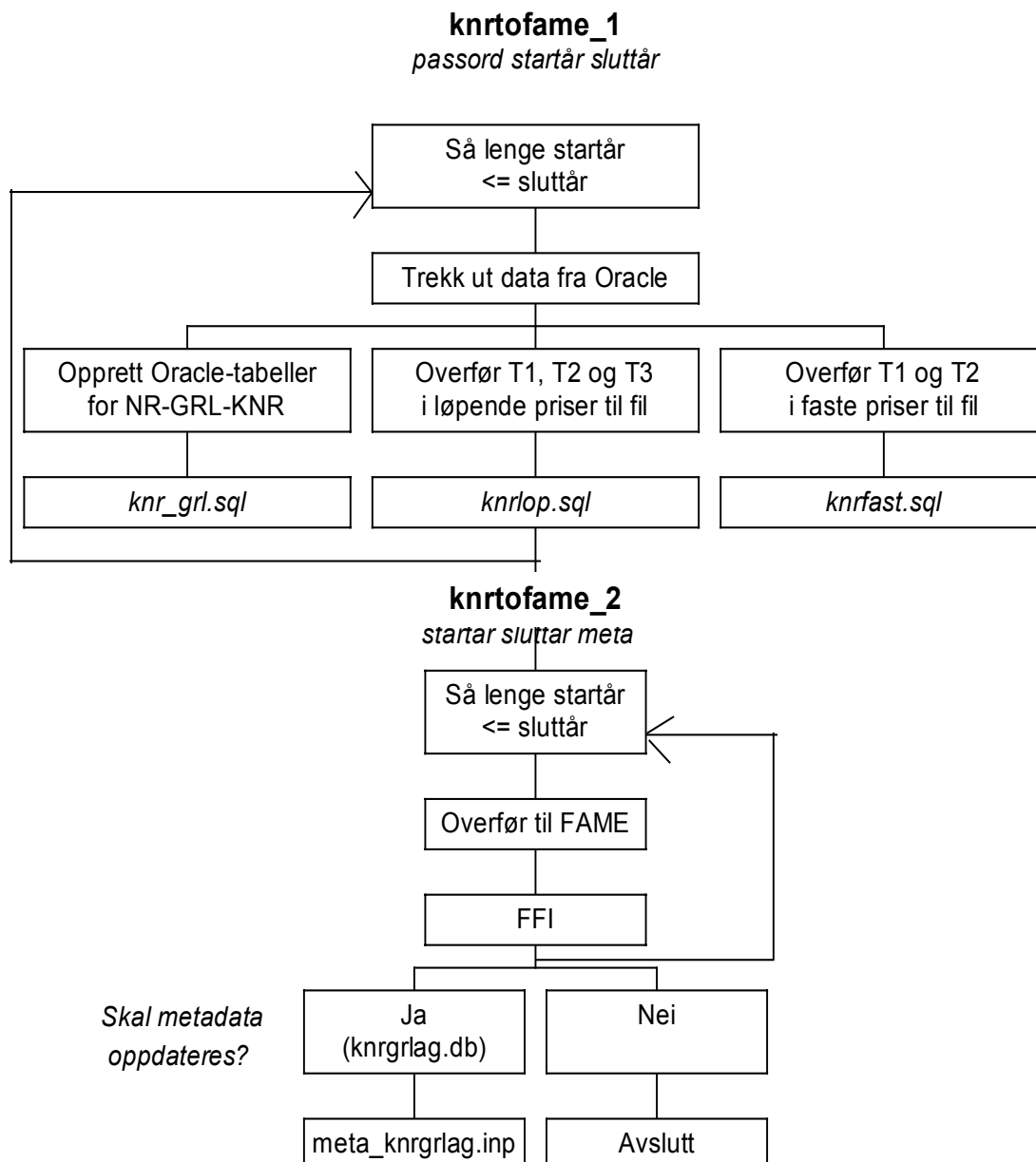
Til beregning av kvartalsvis nasjonalregnskap trenger vi grunnlagsdata på KNR-nivå, samt enkelte detaljerte serier for produkt og næring.

Aggregering fra NR til KNR gjøres via omkodingskatalogen i ORACLE, og det er opprettet tabeller i ORACLE som kobler GRL og KNR med T1, T2 og T3. SQL-scriptet *knr_grl.sql* oppretter tabellene *t1_knr_grl*, *t2_knr_grl*, *t3_knr_grl* og *t2_31_knr_grl*. Scriptet forutsetter at omkodingskatalogen finnes i ORACLE-databasen.

For å trekke tall ut av ORACLE-databasen NR-REA må man logge seg på maskinen cervus og gå til katalogen *\$NASJREGN/tidsserier/oracle/*.

- For å trekke ut tall for T1 og T2 i løpende priser, faste priser eller T3, kjøres UNIX-scriptene **knrtofame_1** og **knrtofame_2** (det opprinnelige scriptet er delt opp i to forskjellige script fordi man ikke får tilgang til FAME fra cervus-serveren). Parametre til *knrtofame_1* er *passord*, *startår* og *sluttår*. Parametre til *knrtofame_2* er *startår*, *sluttår* og *ja/nei* for om metadata skal oppdateres eller ikke.

Knrtofame_1 er en styringsrutine som starter rutiner for overføring av data fra ORACLE til flate filer, mens knrtofame_2 oppdaterer knrgrlag.db med FFI. Knrtofame_1 kjøres fra cervus, men nrtofame_2 må kjøres fra en server der FAME er tilgjengelig, enten micmac eller frisch. Overføringen gjelder løpende priser, faste priser og T3. For hvert år som skal overføres starter knrtofame_1 SQL-scriptet knr_grl.sql og kaller deretter opp underrutinene lastfastknr1 (**OBS! Årstall for manglende tidsserier i faste priser er hardkodet, og må ev. rettes**) og lastlopknr1. Disse programmene starter igjen opp SQLPLUS og kjører SQL-scriptene knrfast.sql knrlop.sql som trekker ut tall for et år om gangen. Resultatet er ASCII-filer med navnestruktur **lop_knr/fast_knr/+ år +.dat**, eks. lop_knr88.dat, fast_knr90.dat. Deretter startes UNIX-scriptet knrtofame_2, som kjører FFI og lager/oppdaterer tidsserier i FAME-databasen knrgrlag.db med innholdet i dat-filen. Til slutt kalles eventuelt metarutinene opp. Figur nedenfor viser rekkefølgen rutiner kalles opp i³.



³ Første del av scriptet knrtofame har endret navn til knrtofame_1, og henter ut data til fil fra Oracle. Overføring til FAME vha. FFI og FAME-rutinene kalles opp av knrtofame_2. Scriptet er delt fordi FAME ikke er tilgjengelig fra cervus (hvor man må være pålogget for å få tilgang til Oracle). Ellers er alt uendret.

Figur 2-2 Styringsrutinene knrtofame_1 og knrtofame_2
 Tabell 2-5 Oversikt over overføringsrutiner for KNR-REA

Navn på rutine	Kaller opp rutine	Beskrivelse
knrtofame_1	knr_grl.sql, lastfastknr1, lastlopknr1	Hovedrutine (del 1) som kaller opp SQL-rutiner for overføring av NR-REA på KNR-nivå til flate tekstfiler. Må startes fra cervus
knrtofame_2	lastfastknr2, lastlopknr2, null_knr.inp, bif_tillegg.inp, bif.inp, fiktiv_knr.inp, trkons.inp meta_knrgrlag.inp	Hovedrutine (del 2) for overføring av data vha. FFI, samt diverse tilleggs- og evt. metarutiner (FAME-rutiner). Må startes fra micmac eller frisch
knr_grl.sql		Oppretter tabellene t1_knr_grl, t2_knr_grl, t3_knr_grl og t2_31_knr_grl i Oracle. Tabellene kobler GRL og KNR med T1, T2 og T3.
lastlopknr1	knrlop.sql	Kaller opp knrlop.sql og spesifiserer Oracle-database, år og passord.
lastlopknr2	FFI	Kaller opp FFI. Tar år som parameter
lastfastknr1	knrfast.sql	Kaller opp knrfast.sql og spesifiserer Oracle-database, år og passord.
lastfastknr2	FFI	Kaller opp FFI. Tar år som parameter
knrlop.sql		Utfører SQL-kommandoer mot Oracle-databaser for å trekke ut tall i løpende priser fra T1, T2 og T3
knrfast.sql		Utfører SQL-kommandoer mot Oracle-databaser for å trekke ut tall i faste priser fra T1 og T2.
FFI		Flat File Interface (dataoverføring til FAME)

Område for overføringsrutiner: *\$NASJREGN/tidsserier/oracle/*

Område for aggregerings- og tilleggsrutiner: *\$NASJREGN/tidsserier/aggknr/*

Område for metarutine: *\$NASJREGN/tidsserier/meta/*

Database: *\$NASJREGN/tidsserier/NR/DB/knrgrlag.db*

2.2.3 Innhold i databaser

Det er opprettet to databaser med årsdata på KNR-nivå. De ligger på \$NASJREGN/tidsserier/NR/DB/.

knrglag.db	Inneholder tall på detaljert nivå
knrglag_agg.db	Inneholder aggregater

Databasen knrglag.db inneholder tidsserier med grunnlagstall for enkelte KNR-størrelser. I tillegg finnes produksjon, produktinnsats, nyinvesteringer og konsum på detaljert NR for utvalgte næringer og produkter.

Navnstandard på serier er den samme som for NR-REA. I øyeblikket inneholder knrglag.db tall for årene 1978-1998. I tillegg til de variabler som finnes i nr01.db finnes også følgende variabler:

Tabell 2-6 Variabler i KNR-REA

Variabel	Beskrivelse
EKSB	Eksport, basisverdi
SBRK	Salg av brukt realkapital
NYIB	Nyinvesteringer i basisverdi
PINe	Produktinnsats av elektrisitetsprodukter
PINf	Produktinnsats av fyringsoljer
MOMS	Merverdiavgift etter produkt
TOLL	Toll etter produkt
AVA	Varehandelsavanse, fordelt på innenlandsk anv. og leveranse til eksport
KOH+kode+gebyr	Sum gebyrer for enkelte konsumgrupper

2.2.4 KNR-aggregater

Tabellaggregater etter KNR-næring og produkt lagres som formler i knrglag_agg.db. Aggregering og navnsetting er den samme som for kvartalsvis nasjonalregnskap, men fordi de måtte tilpasses noe er det allikevel laget et sett med egne aggregeringsrutiner til årsdata på KNR-nivå. Dette betyr at dersom endringer gjøres for aggregering av kvartalsvise serier vil ikke automatisk aggregering av årsdata bli oppdatert tilsvarende. Endringer må gjøres begge steder.

For å unngå problemet med ikke-eksisterende kombinasjoner av variabel, klassifikasjon og verdsett er det laget enkelte fiktive null-serier. Dette for å kunne bruke samme aggregering som for KNR, hvor vi alltid har fullstendige lister uansett om innholdet er null. Inputfilen null_knr.inp oppretter null-serier, mens aggreger.inp kaller opp alle aggregeringsrutiner med riktige variabler og verdsett.

Tabellen på neste side viser alle aggregeringsrutiner for knrglag.db. Originale rutiner for KNR ligger på området \$NASJREGN/knr/struktur/.

Tabell 2-7 Aggregeringsrutiner på KNR-nivå

Rutine	Type aggregat
\$aggcp_nr	Konsum i husholdninger aggregert
\$aggps_nr	Næringsaggregater uten 29-konti
\$aggps29_nr	Næringsaggregater med 29-konti (for BNP, produktinnsats og driftsresultat)
\$aggjs_nr	Investeringer aggregert etter næring
\$aggjr_nr	Investeringer aggregert over art
\$aggeks_nr	Eksport aggregert
\$aggimp_nr	Import aggregert
\$aggmakro_nr	Enkelte makroøkonomiske hovedstørrelser
\$aggva_nr	Enkelte produkt-aggregater
\$aggkonsum_nr	Aggregert konsum i stat og kommune, og hovedtall for konsum

Område: *\$NASJREGN/tidsserier/aggknr/*
 Inputfil: *aggreger.inp, lag_null.inp*

2.2.5 Metadata

For å generere beskrivelser på detaljerte serier og aggregater brukes samme metarutine som til NR-REA, men beskrivelser fra kvartalsvis nasjonalregnskap. Inputfilen *meta_knrglag.inp* åpner databaser og kaller opp metarutinen *\$meta_ledd3*.

Område for metarutine: *\$NASJREGN/tidsserier/meta/*
 Område for beskrivelsesfiler: *\$NASJREGN/knr/struktur/*
 Inputfil: *\$NASJREGN/tidsserier/aggknr/aggreger.inp*

2.3 LØNN OG SYSSELSETTING, NR- OG KNR-NIVÅ

2.3.1 Overføring av tall fra flat fil til FAME

Selve overføringen av lønns- og sysselsettingstall er ikke like automatisert som for NR-REA, da mesteparten av beregningene fremdeles utføres vha. Calcit regneark. Alle detaljerte serier for sysselsetting blir gitt som flate tekstfiler av Stein Hansen (seksjon 210). Aggregatene lages som serier og formler i FAME, og lagres i separate databaser (se kapittel 2.3.2). Detaljerte serier med årsanslag for lønn på 5-siffer NR-nivå blir også gitt som flate tekstfiler av Stein Hansen til Utsyns- og Marsregnskapet, og legges i en temporær database, men seriene benyttes bare til beregning av kvartalstall. Disse seriene aggregeres til KNR-nivå og lagres i knrglag.db. Til de ordinære årsregnskapene hentes imidlertid årstall for lønn ut fra T3 og legges i nr01.db og nr02.db (se kapittel 2.1). For lønn og sysselsetting er det for øvrig laget forskjellige oppdateringsrutiner.

Datafiler gitt av Stein samles i to filer med lønn og sysselsetting hver for seg. Detaljerte serier lastes inn i FAME vha. overføringsrutinen FFI. For å kjøre FFI brukes en beskrivelsesfil, `syss.des`, som beskriver innholdet i datafilen. Date-range i `syss.des` må endres etter hvilke år som overføres. Det er også viktig at antall desimaler som er brukt i datafilene stemmer overens med antall desimaler som er definert i beskrivelsesfilen. Den samme beskrivelsesfilen benyttes til overføring av både lønn og sysselsetting.

- Det er viktig at aggregater lagret som tidsserier oppdateres hver gang nye detaljer er overført. For å sikre dette må overføringen kjøres vha. UNIX-scriptene `sysstofame` og `lokotofame`. Scriptene har samlet kommandoer for kall på FFI, aggregerings- og meta-rutiner. Før filene tas inn, må man endre navn på datafiler og databaser i UNIX-scriptene etter hva overføringen gjelder, og evt. 'luke' ut dersom man ikke ønsker å oppdatere formler eller metadata.

Eks. på FFI-kall fra UNIX-scriptet `sysstofame` (ved kall fra FAME benyttes "!" foran kommandoen):

FFI-kall	desc_fil	datafil	FAME-database hvor serier skal lagres
<code>\$FAME/ffi</code>	<code>syss.des</code>	<code>syss88_92.dat</code>	<code>\$NASJREGN/tidsserier/NR/DB/nrsyss.db</code>

Tabell 2-8 Aggregerings- og metarutiner som kalles av UNIX-scriptet `sysstofame`

Kalles opp	Beskrivelse
<code>nrsyss_sum.inp</code>	Summerer lønnstakere og selvstendige
<code>agg_nrsyss.inp</code>	Kaller aggregeringsrutine for tabellaggregater, NR-nivå
<code>agg_tillegg_nrsyss.inp</code>	Kjøres bare hvis nye aggregeringsformler skal legges inn, NR-nivå
<code>nrsyss_knr.inp</code>	Kaller aggregeringsrutine, summerer fra NR- til detaljert KNR-nivå
<code>meta_nrsyss.inp</code>	Legger inn beskrivelser, NR-nivå
<code>meta_nrsyss_agg.inp</code>	Legger inn beskrivelser, NR-aggregater
<code>meta_knrsyss.inp</code>	Legger inn beskrivelser, KNR-nivå
<code>meta_knrsyss_agg.inp</code>	Legger inn beskrivelser, KNR-aggregater

Område for overføringsrutiner: `$NASJREGN/tidsserier/oracle/famedata/`

Område for aggregerings- og metarutiner: `$NASJREGN/tidsserier/meta/`

Eks. på FFI-kall fra UNIX-scriptet `lokotofame` (ved kall fra FAME benyttes "!" foran kommandoen):

FFI-kall	desc_fil	datafil	FAME-database hvor serier skal lagres
\$FAME/ffi	syss.des	loko95.dat	\$NASJREGN/tidsserier/NR/DB/nrsyss_tmp.db

Tabell 2-9 Aggregerings- og metarutiner som kalles opp av UNIX-scriptet *lokotofame*

Kalles opp	Beskrivelse
agg_nrloko.inp	Kaller evt. aggregeringsrutine for tabellaggregater, NR-nivå
nrloko_knr.inp	Kaller aggregeringsrutine, summerer fra detaljert NR- til detaljert KNR-nivå (kjøres til alle regnskapsversjoner)
meta_knrglag_loko.inp	Legger inn beskrivelser, KNR-nivå
meta_knrglag_agg_loko.inp	Legger inn beskrivelser, KNR-aggregater

Område for overføringsrutiner: *\$NASJREGN/tidsserier/oracle/famedata/*

Område for aggregerings- og metarutiner: *\$NASJREGN/tidsserier/meta/*

Totalstørrelser for sysselsatte personer (PES), sysselsatte normalårsverk (AAVN), utførte timeverk (TVU) og ukeverk (UV) kommer ikke på flat fil, men lages i FAME vha. rutinen *nrsyss_sum.inp*. Eksempel *pes = pesl + pess*. (Husk å endre start- og sluttdato i filen). For mer detaljert beskrivelse av aggregerings- og metarutiner vises til kapittel 3 og 4.

Lønnstallene overføres til *nrsyss_tmp.db*, og aggregeringsrutinene summerer først opp lønns serier og beregner deretter arbeidsgiveravgift på KNR-nivå. Seriene etter KNR-næring legges i *knrglag.db*, som benyttes til kvartalsberegningene, med de nye tallene. Med **ett unntak** skal data for lønn og lønnskostnader **ikke** overføres til *nr01.db* og *nr02.db* til Mars- og Utsynsregnskapene, fordi det ikke gis data for alle variable. Næring 22950 må imidlertid oppdateres for å få riktige serier for markeds- og ikke-markedsrettede næringer. Til de øvrige regnskapsversjonene inngår overføringen som en del av de ordinære årsregnskapsrutinene.

2.3.2 Innhold i databaser

Det er opprettet fire FAME-databaser for sysselsettingstall, og en database for foreløpige lønns- og sysselsettingstall. Alle ligger lagret på UNIX-området *\$NASJREGN/tidsserier/NR/DB/*:

<i>nrsyss.db</i>	Sysselsetting, detaljert (5-siffer) NR-nivå og industri KNR-nivå
<i>nrsyss_agg.db</i>	Sysselsetting, NR-aggregater
<i>knrsyss.db</i>	Sysselsetting, KNR-næringer
<i>knrsyss_agg.db</i>	Sysselsetting, KNR-aggregater (formler)
<i>knrglag.db</i>	Lønn og lønnskostnader, KNR-næringer
<i>nrsyss_tmp.db</i>	Foreløpige tall for lønn, detaljert (5-siffer) NR-nivå og industri på KNR-nivå

Databasen *nrsyss.db* inneholder detaljerte og aggregerte årlige tidsserier for sysselsetting, for øyeblikket for årene 1978 til 1998. De fleste serier er etter NR-næring, mens industrien er etter både NR- og KNR-næring, da opplysninger på KNR-nivå foreligger på et tidligere tidspunkt. Aggregater opprettes som tidsserier og formler, se kapittel 4.

Databasen *nrsyss_tmp.db* inneholder foreløpige årstall for lønn som gis i forbindelse med Utsyns- og Marsregnskapene. Aggregering av foreløpige tall fra NR- til KNR-næring gjøres for å bruke de sist tilgjengelige sysselsettingstall i indikatormakroer i KNR, og kvartalsfordele disse. Sysselsetting etter KNR-næring lagres i en separat database *knrsyss.db*. KNR-aggregatene (formler) lagres i

knrsyss_agg.db. Aggregering fra NR til KNR gjøres i rutinen \$nrsyss_knr vha. case-seriene liskps (KNR-næring) og aggpss (NR-næringer) i \$NASJREGN/knr/struktur/kontoplan.db.

Område: *\$NASJREGN/tidsserier/meta/*

Rutiner: *nrsyss_sum.inp, agg_nrsyss.inp, \$nragg, nrsyss_knr.inp, \$nrsyss_knr*

2.3.3 Navnestruktur

Navn på tidsserier i nrsyss.db, nrsyss_tmp.db og knrsyss.db er laget etter samme prinsipper som KNR og NR-REA, og består av 3 hoveddeler:

1. Variabeltype (Sysselsatte personer, utførte timeverk etc.)
2. Næringsklassifikasjon
3. Prissett (vr)

Eksempel på serienavn: peslk.nr23010.vr, pess.nr23045.vr, loko.nr2310.vr, aavnsk.nr22015.vr, aavn1.nr26851.vr. Databasene inneholder absolutte tall/løpende priser for følgende variabler:

Tabell 2-10 Variable i NR-SYSS

Variabel	Beskrivelse
LOKO	Lønnskostnader
LON	Lønnssum
PES *	Sysselsatte personer (aggregat)
PESL	Sysselsatte personer, lønnstakere
PESS	Sysselsatte personer, selvstendige
PESLK	Sysselsatte personer, lønnstakere, kvinner
PESSK	Sysselsatte personer, selvstendige, kvinner
AAVN *	Sysselsatte normalårsverk (aggregat)
AAVNL	Sysselsatte normalårsverk, lønnstakere
AAVNS	Sysselsatte normalårsverk, selvstendige
AAVNLK	Sysselsatte normalårsverk, lønnstakere, kv.
AAVNSK	Sysselsatte normalårsverk, selvstendige, kv.
TVU *	Utførte timeverk (aggregat)
TVUL	Utførte timeverk, lønnstakere
TVUS	Utførte timeverk, selvstendige
TVULK	Utførte timeverk, lønnstakere, kvinner
TVUSK	Utførte timeverk, selvstendige, kvinner
TVNL	Normaltimeverk, lønnstakere
TVOL	Overtidstimeverk, lønnstakere
UV *	Ukeverk (aggregat)
UVL	Ukeverk, lønnstakere
UVS	Ukeverk, selvstendige

2.4 SYSSELSETTING ETTER KJØNN OG UTDANNING

2.4.1 Innhold i databaser

Det er opprettet en FAME-database for lønns- og sysselsettingstall fordelt på kjønn og utdanning. Databasen ligger lagret på UNIX-området \$NASJREGN/frem_arb/_ut_530/, og heter frem_arb.db. Den inneholder årlige tidsserier etter KNR-nivå, for øyeblikket for årene 1978 til 1995. Rutinene som benyttes til å utføre beregninger og fordele årstallene på utdanningsvariable er programmert i SAS. Data kjøres ut vha. SAS til flate filer, og overføres deretter til FAME-databasen vha. overføringsprogrammet FFI (Flat File Interface). I tillegg er det laget en rutine som gjør det mulig å hente ut data fra Myfame (*oppdater_refdb.inp*).

En teknisk dokumentasjon av SAS-rutinene mv. er under utarbeiding (Ingunn Sagelvmø).

Databaseområde: *\$NASJREGN/frem_arb/_ut_530/*
Databaser: *frem_arb.db, frem_arb_myfame.db*
Område for SAS-rutiner: *\$NASJREGNfrem_arb/prog/*

2.4.2 Navnestruktur

Navn på tidsserier i frem_arb.db er laget etter samme prinsipper som til øvrige data etter KNR-nivå. I tillegg inneholder variabelnavnet et ledd for utdanning. Navnet består av 4 hoveddeler:

1. Variabeltype (Sysselsatte personer, utførte timeverk etc.)
2. Næringsklassifikasjon
3. Utdanningsklassifikasjon
4. Prissett (vr)

Eksempel på serienavn: peslk.nr2301.gr.vr, pessm.nr2305.uopp.vr, lotlk.nr2310.uni.vr osv.
Databasene inneholder absolutte tall/løpende priser for følgende variabler:

Tabell 2-11 Sysselsettingsvariable i frem_arb.db

Variabel	Beskrivelse
LOTLM	Lønn per timeverk, lønnstakere, menn
LOTLK	Lønn per timeverk, lønnstakere, kvinner
PESLM	Sysselsatte personer, lønnstakere, menn
PESLK	Sysselsatte personer, lønnstakere, kvinner
PESSM	Sysselsatte personer, selvstendige, menn
PESSK	Sysselsatte personer, selvstendige, kvinner
AAVNLM	Sysselsatte normalårsverk, lønnstakere, menn
AAVNLK	Sysselsatte normalårsverk, lønnstakere, kvinner
AAVNSM	Sysselsatte normalårsverk, selvstendige, menn
AAVNSK	Sysselsatte normalårsverk, selvstendige, kvinner
TVULM	Utførte timeverk, lønnstakere, menn
TVULK	Utførte timeverk, lønnstakere, kvinner
TVUSM	Utførte timeverk, selvstendige, menn
TVUSK	Utførte timeverk, selvstendige, kvinner

Tabell 2-12 Utdanningsvariable i frem arb.db

Variabel	Beskrivelse
GR	Grunnskoleutdanning
VF	Videregående fagutdanning
VA	Videregående almenfag, adm/økonomisk utd.
HOY	Høgskoleutdanning
UNI	Universitetsutdanning
UOPP	Uoppsett utdanning

2.5 KAPITALSLIT OG KAPITALBEHOLDNING

2.5.1 Overføring av tall fra flat fil til FAME

Tallene for kapitalslit og kapitalbeholdning beregnes ved hjelp av SAS. Resultatet kommer på flate filer, som legges inn i FAME med FFI.

Filene ligger på UNIX-området \$NASJREGN/nyberkap/wk12. Det er seks stykker i alt, tre for detaljert NR-nivå og tre med KNR-nivå:

Tabell 2-13 Flate filer med kapitaltall

Flat fil	Innhold
nr_s_YY0X_lf.dat	Kapitalslit og kapitalbeholdning etter NR-næring, i løpende og faste priser
nr_a_YY0X_lf.dat	Kapitalslit og kapitalbeholdning etter BERKAP-art
nr_s_a_YY0X_lf.dat	Kapitalslit og kapitalbeholdning etter NR-næring og BERKAP-art
knr_s_YY0X_lf.dat	Kapitalslit og kapitalbeholdning etter KNR-næring, i løpende og faste priser
knr_a_YY0X_lf.dat	Kapitalslit og kapitalbeholdning etter KNR-art
knr_s_a_YY0X_lf.dat	Kapitalslit og kapitalbeholdning etter KNR-næring og KNR-art

Her viser YY til årgangen, for eksempel 96.

X viser til regnskapsversjon (0 for Utsynsregnskapet, 1 for Marsregnskapet og 2 for Endelig årsregnskap).

Overføringsprogrammene ligger på \$NASJREGN/nyberkap/fame.

Tabell 2-14 Overføringsprogrammer for kapitaltall

Overføringsprogram	Innhold
lastnrkap	Legger inn kapitalslit og kapitalbeholdning på detaljert NR-nivå i basen nrkap.db
lastknrkap	Legger inn kapitalslit og kapitalbeholdning på detaljert KNR-nivå i basen knrkap.db

2.5.2 Innhold i databaser

Det er opprettet fire FAME-databaser for kapitaltallene. Alle ligger lagret på UNIX-området \$NASJREGN/tidsserier/NR/DB/:

Tabell 2-15 Databaser med kapitaltall

Base	Innhold
nrkap.db	Kapitaltall, detaljert (5-siffer) NR-nivå og "BERKAP-art"
nrkap_agg.db	Kapitaltall på aggregert NR-nivå
knrkap.db	Kapitaltall på detaljert KNR-nivå
knrkap_agg.db	Kapitaltall på aggregert KNR-nivå

Tabell 2-16 Variable i kapitaldatabasene

Variabel	Beskrivelse
DEP	Kapitalslit
KAPN	Kapitalbeholdning, netto
KAPB	Kapitalbeholdning, brutto (ikke i bruk, kopi av KAPN)

3. Aggregater

Dette kapitlet omhandler aggregater til NR-REA og NR-SYSS. Det gis en oversikt over hvilke aggregater som finnes, navn på aggregeringsrutiner og organisering av disse. Til slutt følger retningslinjer for hvordan man definerer nye aggregater.

3.1 Generelt

Et aggregat vil si en sum av flere andre serier. Omfanget gjør at de nesten regnes som en ekstra kontoplan i tillegg til vår egentlige. Se vedlegg ?-? for oversikt over alle aggregater i nr02.db og nrsyss.db. Alle aggregeringsrutiner og tekstfiler ligger på \$NASJREGN/tidsserier/meta/.

Aggregater til NR-REA er laget som formler og tidsserier i nr02.db, tilsammen rundt 10,000 aggregater. De fleste aggregater som summerer opp mange underliggende serier er lagret som tidsserier (rundt 8,000), og **må alltid oppdateres** samtidig med at detaljer i nr01.db endrer verdi. Andre hovedstørrelser til NR-REA er lagret som formler og **oppdateres automatisk** når underliggende serier endrer verdi.

Detaljerte 5-siffer sysselsettingstall lagres i nrsyss.db (ca. 8000 serier). Aggregater til NR-SYSS kan **ikke** lagres i samme database som de detaljerte seriene (slik rutinene opprinnelig var programmert), da dette medfører dobbelttelling. Aggregater lagres derfor separat i nrsyss_agg.db (ca. 6500 tidsserier og formler).

I starten var alle aggregater lagret som formler, noe som sparer en for mye arbeid med oppdatering av databaser. Erfaringen er imidlertid at det er vanskelig å garantere at rundt 10,000 formler til enhver tid inneholder det de skal. På den annen side kan vi til dels løse dette problemet ved å bruke noen FAME-funksjoner (selectnames, exists(@name)) som overser manglende serier (f.eks. i et crosslist-uttrykk).

Vi kan også benytte FAME-funksjoner som ignorerer ND-verdier som skulle vært lik null. Et annet problem vil da være at når underliggende serier ikke er oppdatert for alle variable i hele perioden blir aggregatet feil, noe man må være oppmerksom på bl.a. for lønns- og sysselsettingsvariable, da disse seriene oppdateres i flere omganger. Problemet vil imidlertid ikke løses ved å lage serier, med mindre aggregatene bare oppdateres innenfor et tidsrom der alle seriene like lange. For publisering og oppslag på seksjonen vil det derfor ofte være sikrere å jobbe med serier fordi innholdet «fryses» etter hver oppdatering. Dersom man finner det mer hensiktsmessig å jobbe med serier enn formler (eller omvendt), bør rutiner endres slik at alle aggregater defineres som serier (eller formler).

Felles for alle rutiner er en innlagt sjekk på om aggregatet finnes fra før av. Finnes det allerede brukes en FAME-kommando 'set' for å oppdatere serien, hvis ikke lages en ny serie vha. FAME-kommandoen 'new'. Dette for å unngå at metadata må oppdateres hver gang man overfører nye data fra ORACLE, men kun når nye aggregater/beskrivelser dukker opp.

Attributten desc_source oppdateres samtidig med innholdet i aggregatet. Dersom nye næringer og produkter «dukker opp» fra en ORACLE-overføring til neste vil disse automatisk bli inkludert der hvor aggregater lages som serier (når lsum(wildlist(...))-funksjonen benyttes). Aggregering for de forskjellige prissett gjøres adskilt slik at en variabel ikke nødvendigvis må finnes i alle verdisett.

Alle aggregeringsrutiner, foruten formel-aggregater, bruker samme tekstfilen til å generere et aggregat som til å oppdatere metadata. Eksempel tekstfilen agg_4sifret.txt inneholder alle næringsaggregater på 4-sifret nivå, med tilhørende norske og engelske beskrivelser.

Tabell 3-1 Eksempel på tekstfil med klassifikasjonskoder

2301	Jordbruk	Agriculture
2302	Skogbruk	Forestry
2305	Fiske og fangst	Fishing

Når man skal lage selve aggregatet loop'es det gjennom første kolonne i tekstfilen i kombinasjon med ønskede variabler og verdisett. Man slipper mao. å skrive inn listen over alle aggregater, det holder at de er definert et sted. For å unngå ajourhold av uendelig mange tekstfiler er alle lister fullstendige uansett variabeltype, dvs. agg_4sifret.txt inneholder alle kontotyper fra 22 til 86. Alle rutiner sjekker for ikke-eksisterende kombinasjoner, og det lages f.eks. ikke aggregater for eksport i 23010 eller konsum av 51110. Alle aggregeringsrutiner kalles opp vha. input-filer. Tabellen nedenfor viser hvilke inputfiler som brukes for de forskjellige databaser:

Tabell 3-2 Inputfiler som kaller opp alle aggregeringsrutiner

Inputfil	Database
agg_nrrea.inp	Oppretter alle aggregater lagret som serier i nr02.db
agg_ekstra.inp	Oppretter aggregater lagret som formler i nr02.db
agg_tillegg.inp	Oppretter aggregater lagret som formler i nr02.db
nrsyss_sum.inp	Summerer lønnstakere og selvstendige i nrsyss.db
agg_nrsyss.inp	Oppretter alle aggregater lagret som formler i nrsyss.db

3.2 Navnestruktur

Navnsetting av aggregater følger samme prinsipper som detaljerte serier i nr01.db. Det er forsøkt å bruke en navnestruktur hvor det fremkommer av navnet at en serie er et aggregat, enten ved bruk av symboler eller ved alfanumeriske koder. Det er brukt ulik navnsetting for de forskjellige aggregat-typer.

3.3 Type aggregater

3.3.1 «Rene» næringsaggregater

Næringsaggregater som følger struktur i kontoplanen summerer opp alle underliggende serier vha. wildcards i FAME. Tegnet «^» markerer et tegn i FAME, mens «?» angir et ukjent antall tegn. Det summeres alltid helt fra detaljerte næringer, altså ikke «nøstet» aggregering. Tegnet underscore «_» i serienavnet angir at aggregatet er en sum av alt under. Rutinen \$aggnaer tar som input lister med variabeltyper, navn på aggregater (ut fra første kolonne i en tekstfil), verdsett og antall wildcards som skal brukes. Eksempel:

Tabell 3-3 Rene næringsaggregater

Navn	Innhold	FAME
series prob.nr23_.vr	Sum produksjon 23xxx, løpende priser	sum prob.nr23^^.vr
series bif.nr846_.fp	Sum investeringer 846xx, faste priser	sum bif.nr846^^.fp
series pin.nr2310_.vr	Sum produktinnsats 2310x, løpende priser	sum av pin.nr2310^.vr
series lon.nr2402_.vr	Sum utbetalt lønn 2402x, løpende priser	sum lon.nr2402^.vr

Navn på rutine: \$aggnaer

Utbedring: Kunne vært gjort enklere mht. argumenter... Antall 'hatter' er ikke så enkelt å forstå, men innført for å unngå dobbel summering der hvor variabler finnes både for sum næringer og detaljert næring og produkt/art.

3.3.2 Mellomaggregater

Aggregater som er definerte som sum av 4-sifrede næringsaggregater av type 2301_, 2445_ er (av en eller annen grunn) kalt mellomaggregater. Denne type aggregering forutsetter at 4-sifrede aggregater er laget først.

Næringsaggregater på mellomnivå lages ved hjelp av navnelister. Navnelistene opprettes i lag_mellomi.inp og lag_mellomp.inp, og lagres i mellom.db. Rutinen lag_mellomi.inp lager lister ut fra investeringsnæringer 82-86 med prefikset I, mens lag_mellomp.inp lager lister ut fra produksjonsnæringer 22-26 med prefikset P. Hver enkelt navneliste inneholder navn på alle mulige 4-sifrede næringer innen et angitt intervall. Navnelisten med alle mulige blir så brukt når aggregat for de forskjellige variablene skal lages.

Input til aggregeringsrutinen \$aggmellom er lister med variabeltyper, navn på aggregater ut fra `agg_mellom.txt`, verdisett og prefiks på navnelisten (P for produksjonsnæringer og I for investeringsnæringer). Rutinen tester på om hvert enkelt 4-sifret aggregat finnes for akkurat den variabelen, og kun de som eksisterer vil inngå i summen. Dersom ingen finnes blir ikke aggregatet opprettet for den variabelen.

Mellomaggregater som er sum av 22-26 eller 82-86 er navnsatt med suffikset N for å vise at det er et næringsaggregat. De som er sum av 24-25 eller 84-85 har suffikset OFN for offentlig forvaltning. Aggregater som er sum 22-23 har suffikset M for markedsrettet og eget bruk. Nedenfor følger en oversikt med eksempel på navnsetting.

Tabell 3-4 Navnsetting av mellomaggregater

Suffix	Definisjon	Eksempel på serie
N	22-26 evt. 82-86	pin.nr01_95n.fp
OFN	24-25 evt. 84-85	bnpb.n01_95ofn.vr
M	22-23 evt. 82-83	prob.nr01_95m.vr

Under følger eksempel på forskjellige mellomaggregater. Tabellen viser navn på aggregatet og hvilken navneliste fra `mellom.db` med alle mulige 4-sifrede aggregater som er brukt. Siste kolonne viser hva som finnes, og som dermed inngår i summen.

Tabell 3-5 Eksempel på avledete aggregater

Navn på aggregat	Navneliste med alle mulige koder i intervallet	Hva som faktisk inngår i aggregatet
series bif.nr10_14n.vr	namelist I10_14N = nr8210_, nr8211_, nr8212_, nr8213_, nr8214_, nr8310_,nr8311_, nr8312_, nr8313_, nr8314_, nr8410_, nr8411_, nr8412_, nr8413_, nr8414_, nr8510_, nr8511_, nr8512_,nr8513_, nr8514_, nr8610_, nr8611_, nr8612_, nr8613_, nr8614_	bif.nr8310_.vr, bif.nr8311_.vr, bif.nr8313_.vr, bif.nr8314_.vr
series prob.nr90_95n.vr	namelist P90_95N = nr2290_, nr2291_, nr2292_, nr2293_, nr2294_, nr2295_,nr2390_, nr2391_, nr2392_, nr2393_, nr2394_, nr2395_, nr2490_, nr2491_, nr2492_, nr2493_, nr2494_, nr2495_,nr2590_, nr2591_, nr2592_, nr2593_, nr2594_, nr2595_, nr2690_, nr2691_, nr2692_, nr2693_, nr2694_, nr2695	prob.nr2295_.vr, prob.nr2390_.vr, prob.nr2391_.vr,prob.nr2392_.vr, prob.nr2393_.vr, prob.nr2492_.vr, prob.nr2590_.vr, prob.nr2592_.vr, prob.nr2691_.vr,prob.nr2692_.vr
series pess.nr01_5n.vr	namelist P01_05N = nr2201_, nr2202_, nr2203_, nr2204_, nr2205_, nr2301_,nr2302_, nr2303_, nr2304_, nr2305_, nr2401_, nr2402_, nr2403_, nr2404_, nr2405_, nr2501_, nr2502_, nr2503_,nr2504_, nr2505_, nr2601_, nr2602_, nr2603_, nr2604_, nr2605	pess.nr2301_.vr, pess.nr2302_.vr, pess.nr2305_.vr

Antall næringer i kontoplanen er mange, og for en som ikke kjenner NR-REA er det vanskelig å vite hvor det finnes produksjon, men ikke produktinnsats etc.. Det er derfor valgt å lage mellomaggregater på denne måten istedet for å taste inn hvilke detaljerte næringer som faktisk inngår.

Det er forholdsvis enkelt å lage nye mellomaggregater. Det kreves at man oppretter en ny navneliste i mellom.db med samme navnestruktur som de som er der fra før (P?n, P?ofn, I?n, I?me etc). Det nye aggregatet føyes så til i tekstfilen agg_mellom.txt, med norsk og evt. engelsk tekst. Neste gang man kjører aggregeringsrutinen \$aggmellom opprettes det nye aggregatet. Beskrivelser legges inn ved å kjøre egne metarutiner (se pkt. ?).

Navn på rutiner: lag_mellomi.inp, lag_mellomp.inp, \$aggmellom

3.3.3 Aggregater for næring og produkt/art

Aggregering over næring og produkt/art følger samme prinsipper som for rene næringsaggregater med bruk av wildcards i FAME. Underscore «_» brukes i serienavnet for å markere aggregering over næring og produkt/art.

Detaljerte serier for næring og produkt/art finnes foreløpig kun for bruttoinvesteringer, eksport og import, og dermed også aggregater kun for disse variablene. For å unngå for mange ikke-eksisterende kombinasjoner, og dermed spare tid, begrenses næringer til kontotype 5 og 8.

Rutinen \$aggn_pa aggregerer produkter for detaljerte næringer 5xxxx og 8xxxx. Input er lister med variabler, de aktuelle detaljerte næringer, liste med produktaggregater (ut fra første kolonne i tekstfil) og verdsett.

Rutinen \$aggna_pa aggregerer over både næring og produkt/art. Input er det samme som til \$aggn_pa, men bruker aggregater for kontotype 5 og 8 istedet for detaljerte næringer.

Tabell 3-6 Aggregater for næring og produkt/art

Navn	Innhold	FAME
series eks.nr51110.nr011_.vr	Eksport, detaljert næring og aggregert produkt	sum eks.nr51110.nr011?.vr
series bif.nr83010.nr008_.fp	Bruttoinv., detaljert næring og aggregert art	sum bif.nr83010.nr008?.fp
series imp.nr521_.nr02_.vr	Import, 3-sifret næring og aggregert produkt	sum imp.nr521?.nr02?.vr
series bif.nr830_.nr008.fp	Bruttoinv., 3-sifret næring og aggregert art	sum bif.nr830?.nr008?.fp

Navn på rutiner: \$aggn_pa, \$aggna_pa

3.3.4 Aggregater for art

Rutinen \$agginv aggregerer bruttoinvesteringer etter art. Rutinen tar argument for verdsett, og aggregater lages som formler. Variabel, navn på aggregatet og innhold er «hardkodet» i prosedyren. Det finnes ingen sjekk på om en underliggende serie faktisk finnes. Enkelte summer er definert som sum av andre aggregater, mao. nøstet aggregering. Suffikset «A» for art er brukt i navn på aggregater for investeringer etter art. Eksport og import aggregert etter produkt finnes kun i knrglag_agg.db. Se avsnittet «KNR-aggregater».

Tabell 3-7 Eksempel på produktaggregater

Navn	Innhold	Tekst
formula bif.nroljea.vr/fp	nr008370, nr008378, nr008380, nr008388, nr008390, nr008398, nr009380, nr009385	Oljevirkosomhet
formula bif.nrboliga.vr/fp	nr0081 , nr0091	Boliger

Navn på rutine: \$agginv

3.3.5 Konsumaggregater

Foruten «rene» konsumaggregater lages også enkelte formler for konsum i husholdninger og offentlig forvaltning. Rutinen \$aggkons tar som input verdisett, og variabler og innhold er hardkodet i prosedyren. Enkelte størrelser er definerte som sum av andre aggregater. Noe av aggregeringen skjer på tvers av variablene KOH (husholdninger), KOO (offentlig forvaltning) og KOI (ideelle organisasjoner), eks. for personlig konsum. Navnsetting av denne type aggregat er noe problematisk fordi det gjelder flere variabler, og ikke alle navn er like gode. Eksempel på konsumaggregater:

Tabell 3-8 Konsumaggregater

Navn	Innhold	Tekst
formula koh.nr61tjen.vr	nr61116, nr61122, nr61211, nr61212, nr61213, nr61231, nr61232, nr61233, nr61315, nr61333, nr61363, nr61364, nr61421, nr61422, nr61423, nr61431, nr61523, nr61524, nr61525, nr6153_, nr61619, nr6162_, nr6171_, nr6181_, nr6182_, nr61911, nr6193_, nr6194_, nr6195_, nr6196_	Konsum av tjenester
formula koh.nrpers.vr/fp	koh.nr61_ + koi.nr66_ + koo.nr6404_ +koo.nr6405_ + koo.nr6406_ +koo.nr6504_ + koo.nr6505_ +koo.nr6506_ + koo.nr6508_ +koo.nr6515_	Personlig konsum
formula koh.nrkoll.vr/fp	koh.nr6_.vr/fp - koh.nrpers.vr/fp	Kollektivt konsum

Navn på rutine: \$aggkons

3.3.6 Andre avledete aggregater

Næringsaggregater på 2-sifret nivå og enkelte hovedstørrelser er lagret som formler i nr02.db og nrsyss.db. Disse opprettes i rutinen \$aggekstra. Navn, innhold og verdisett er hardkodet, mens det loopes igjennom en ferdigdefinert liste med variabler. Rutinen tar dermed ingen argumenter. Dersom samme aggregering skal brukes for flere variabler må listene inne i prosedyren endres. Eksempel på spesialaggregater:

Tabell 3-9 Spesialaggregater

Navn	Innhold	Tekst
formula bnpb.nr01_95fn.vr	bnpb.nr01_95n.vr- bnpb.nr2311_.vr- bnpb.nr23608.vr- bnpb.nr23611.vr	Fastlands-næringer
formula prob.nr2_.fp	prob.nr22_.fp+ prob.nr23_.fp+ prob.nr24_.fp+ prob.nr25_.fp+ prob.nr26_.fp	Produksjon i basisverdi
formula bnpb.nr2_.vr	bnpb.nr22_.vr+ bnpb.nr23_.vr+ bnpb.nr24_.vr+ bnpb.nr25_.vr+ bnpb.nr26_.vr+ bnpb.nr29_.vr	Bruttonasjonalprodukt
formula pin.nr2365U.vr	pin.nr2365_.vr- pin.nr23654.vr- pin.nr23659.vr	Finansiell tjenesteyting (korrigert for frie banktjenester)

Navn på rutiner: \$aggekstra

Utbedring: Kunne splitte opp i flere prosedyrer. Vanskelig å gi variabler som input når disse ikke er de samme gjennom hele rutinen. Startet med spesialaggregering kun pga. korrigerings for frie bank- og forsikringstjenester. Disse skapte problemer fordi variabler skulle behandles forskjellig.

3.4 Hvordan definere nye aggregater

Alle nye aggregater som skal brukes til publisering- og/eller analyseformål bør lagres som formler i nr02.db. Dersom enkelte brukere ønsker å lage sine egne summer lagres disse i separate databaser.

Når det skal defineres nye aggregater til NR-REA bør man først plassere de nye aggregater i forhold til hva som finnes fra før av. Ønsker man å bruke en eksisterende aggregering for en ny variabel kan dette som oftest gjøres enkelt. Variabelen inkluderes i listen som sendes som input til aggregeringsrutinen.

Mellomaggregater, altså sum av 4-sifrede næringer, kan enkelt utvides til å inneholde flere nivåer og flere variabler mens andre spesialaggregater er verre å utvide. Er det en ny aggregering som ikke finnes fra før av oppretter man disse som formler i inputfiler eller prosedyrer. For å lage helt nye aggregater gjelder da følgende retningslinjer:

- *Gi navn til aggregatet*

Aggregatet må få et navn (kun 2. ledd i navnet, resten bestemmes ut fra variabel og prissett).

Navnsetting i tråd med hva som tidligere er brukt. Alfnumeriske koder eller akronymer der hvor ikke symboler kan benyttes. Maks antall karakterer (foruten prefikset «nr») er 9.

- *Bestem variabel og prissett*

Man må bestemme hvilke variabler og prissett aggregatet skal lages i.

- *Opprett aggregatet*

Anbefaler at nye aggregater lagres som formler i nr02.db, enten vha. prosedyrer eller inputfiler. En formel kan alltid skrives helt ut i klartekst, men for å ha muligheten til enkelt å utvide variabel- og verdilisten bruker man looping i kombinasjon FAME-funksjonene LSUM og CROSSLIST for å aggregere. Dette gjøres i rutinen \$lag_formel. Rutinen lager en enkel formel, og tar følgende input:

Tabell 3-10 Argumenter til \$lag_formel

Argument	Type	Eksempel
Variabelliste	namelist	{bnpb, prob, pin}
navn	string	«nr051n»
innhold	namelist	{nr22051, nr23051}
verdisett	namelist	{vr, fp}

Nedenfor følger eksempel på begge metoder:

- Eksempel på «klartekst»:

formula koh.nr61iv.vr = koh.nr610_.vr + koh.nr61221.vr +koh.nr6124_.vr +
koh.nr61361.vr + koh.nr61362.vr + koh.nr61412.vr + koh.nr61522.vr + koh.nr61618.vr +
koh.nr61632.vr + koh.nr61633.vr + koh.nr61913.vr

- Eksempel på bruk av \$lag_formel:

\$lag_formel {koh}, «nr61iv», {nr610_,nr61221,nr6124_,nr61361.....,nr61913}, {vr,fp}

- *Legg inn beskrivelser*

Man må gi aggregatet norsk og evt. engelsk beskrivelse (kun til 2. ledd i navnet). Nye aggregater må finnes i en tekstfil med tilhørende metadata. Problemet vil være å finne ut i hvilken metafil. Der hvor man inkluderer flere nivåer til feks. mellomaggregater må tekstfilen `agg_mellom.txt` inneholde de nye intervaller fordi denne tekstfilen benyttes for å opprette selve aggregatene. Det samme gjelder for nye 'rene aggregater'.

Beskrivelser til andre spesialaggregater kan derimot legges i hvilken som helst tekstfil, da aggregatene defineres uavhengig av metafilen. Teksten kan feks. legges i `agg_naering.txt`, evt. opprette en ny `agg_tillegg.txt`. Det viktige er at alle nye tekstfiler må inkluderes i styringsrutinene `meta_nr02.inp` og `meta_nrsyss.inp`.

4. Metadata

Vil i dette kapitlet gå igjennom metarutiner til NR-REA og NR-SYSS. Alle metarutiner og tekstfiler ligger på området `/tidsserier/meta/`.

4.1 Generelt

Metadata vil i FAME-sammenheng bety beskrivende informasjon knyttet til tidsserier. FAME knytter slike beskrivelser til serier i form av attributter. Alle detaljerte serier har 3 typer attributter knyttet til seg, hvor `desc` og `doc` er FAME-standard mens `desc_en` er egendefinert. Alle aggregater har i tillegg en egendefinert attributt `desc_source` (se neste avsnitt):

attribute <code>desc</code> :	Inneholder den norske beskrivelsen til en serie
attribute <code>doc</code> :	Kan inneholde lengre dokumentasjon på en serie, eks. kilde, evt. brudd i serien,
attribute <code>desc_en</code> :	Inneholder den engelske beskrivelsen til en serie
attribute <code>desc_source</code> :	Inneholder informasjon om hvilke underliggende serier som inngår i et aggregat (se neste avsnitt)

For alle detaljerte serier inneholder "`desc`" inntil 45 posisjoner av den lange beskrivelsen på klassifikasjonsleddet fra NR-REAs kontoplan, mens «`desc_en`» foreløpig er blank. For aggregater inneholder «`desc`» den norske beskrivelsen på serien, og flere engelske tekster finnes etterhvert i «`desc_en`».

Norsk og engelsk beskrivelse på en bestemt serie vil være sammensatt av flere ledd, slik som selve serienavnet. Hvert ledd i et sammensatt serienavn har en tekst knyttet til seg, og en unik beskrivelse for hver tidsserie lages ved å lime sammen den teksten som hører til det enkelte ledd. Underscore «`_`» er brukt som separator.

Det er viktig å huske at 2. ledd i en seriebeskrivelse er den teksten som brukes som forspaltetekst dersom serien skrives ut i tabellrutinen. I eksempelet under er 2. ledd uthevet. Ønsker man å definere forskjellig forspaltetekst for en bestemt serie er ikke dette mulig. Samme tekst vil følge en serie uansett hvilken tabell det gjelder.

Eksempel på beskrivelser til serien `bnpb.nr23010.vr`

<u>Kode</u>	<u>Norsk tekst</u>	<u>Engelsk tekst</u>
-------------	--------------------	----------------------

BNPB	«Bruttoprodukt»	«Value added»
NR23010	«Jordbruk»	«Agriculture»
VR	«Verdi»	«Current prices»

- attribute desc(bnpb.nr23010.vr) = «Bruttoprodukt_ **Jordbruk** _Verdi»
- attribute desc_en(bnpb.nr23010.vr) = «Value added_ **Agriculture** _Current prices»

Rutiner for å generere og oppdatere metadata er forholdsvis enkle. Ideen er å loope seg igjennom alle mulige verdier innen de forskjellige deler i et serienavn (variabeltype, næring/produkt/aggregat og verdisett), og beskrivelser oppdateres der hvor en bestemt kombinasjon representerer en eksisterende serie eller aggregat.

Rutinen \$meta_ledd3 legger inn norske og engelske beskrivelser til serier med 3 ledd, mens \$meta_ledd4 brukes for de med 4 ledd (etter art og næring). Input til begge rutiner er lister med variabler, næringer, evt. produkter og verdisett. For de aggregater hvor beskrivelse og tabelltekst skal være forskjellig avhengig av variabeltype brukes rutinen \$meta_spesial. Input her er en liste med variabel og aggregat samt verdisett.

Navn på rutiner: *\$meta_ledd3*, *\$meta_ledd4*, *\$meta_spesial*

Navn på inputfiler: *meta_nr01.inp*, *meta_nr02.inp*, *meta_nrsyss.inp*

4.2 Attribute desc_source

Alle aggregater i nr02.db og nrsyss.db har en attributt ‘*desc_source*’ som viser hvilke serier som inngår i summen. Attributten oppdateres samtidig med aggregatets innhold.

For enkelte formler vil desc_source inneholde FAME-funksjonene LSUM og CROSSLIST. LSUM summerer alle serier som er listet opp, og CROSSLIST kombinerer variabel, næring og verdisett. I FAME skriver man ‘*disp desc_source*(navn på serien)’ for å få utskrift av underliggende serier. E

Eksempel på innhold i formelen for konsum av varig varekonsum, koh.nr61v.vr og serien bnpb.nr50_64n:

- *disp desc_source(koh.nr61v.vr)* =lsum(crosslist({koh},{nr61311, nr61312, nr61313, nr61314, nr61331, nr61332, nr61351, nr61411, nr6151_, nr61611, nr61612, nr61613, nr61614, nr61912, nr61921}, {vr}))
- *disp desc_source(bnpb.nr50_64n.vr)* =bnpb.nr2350_.vr, bnpb.nr2351_.vr, bnpb.nr2352_.vr, bnpb.nr2355_.vr, bnpb.nr2360_.vr, bnpb.nr2361_.vr, bnpb.nr2362_.vr, bnpb.nr2363_.vr, bnpb.nr2364_.vr, bnpb.nr2460_.vr, bnpb.nr2463_.vr

4.3 Organisering av metadata

Alle beskrivelsesfiler er lagret som flate tekstfiler på \$NASJREGN/tidsserier/meta/. De fleste aggregat-filer er skrevet inn manuelt eller klippet og limt fra andre dokumenter, mens detaljerte beskrivelser hentes fra ORACLE.

4.3.1 Tekst til variabler og verdisett

Norsk og engelske beskrivelser til første og siste ledd i serienavn på alle detaljer og aggregater til NR-REA og NR-SYSS ligger fordelt på følgende tekstfiler:

Tabell 4-1 Tekstfiler til variabler og verdisett

Tekstfil	Innhold
variabel.txt	Alle realregnskapsvariabler for T1 og T2
t3.txt	Variabler for T3
syss.txt	Variabler for sysselsetting
verdi.txt	Verdisett

4.3.2 Tekst til detaljerte næringer og produkter

Tekstfilene mottaker.txt og produkt.txt inneholder beskrivelser for de mest detaljerte næringer og produkter. Tekstfilene overføres fra ORACLE på tilsvarende måte som tallene, og man må være pålogget på serveren cervus.

UNIX-scriptet nr_meta kaller opp nr_meta.sql, og mottaker.txt og produkt.txt trekkes ut fra ORACLE og legges på \$NASJREGN/tidsserier/meta/.

Tekstfilene inneholder kode og 45 posisjoner av den lange teksten fra ORACLE til bruk i tabeller. Evt. engelske tekster må legges inn som en ekstra kolonne i ORACLE, og også her overføres kun 45 posisjoner. Oppdatering av beskrivelser på detaljerte serier i nr01.db gjøres med inputfilen meta_nr01.inp.

Tabell 4-2 mottaker.txt

Kode (posisjon 1-9)	Oracle langtekst (posisjon 10-55)	Fremtidig engelsk tekst (posisjon 56-100)
22010	Jordbruk	
22015	Jakt, viltstell og tjenester i tilknytning jakt og viltstell	
22454	Bygginstallasjon og innredningsarbeid	

Område for UNIX-script: \$NASJREGN/tidsserier/oracle/

Navn på rutiner: nr_meta, nr_meta.sql

Navn på metafiler: mottaker.txt, produkt.txt

4.3.3 Tekst til aggregater

Tekstfiler for aggregater er splittet opp i mange små etter type aggregat. Dette er gjort delvis fordi de brukes i aggregeringsrutiner, men også fordi det skal være enklere å oppdatere innholdet.

Tabell 4-3 Tekstfiler til aggregater

Tekstfil	Innhold
agg_2sifret.txt	Næringsaggregater, 2-sifret
agg_3sifret.txt	Næringsaggregater, 3-sifret
agg_4sifret.txt	Næringsaggregater, 4-sifret
agg_mellom.txt	Næringsaggregater, avledete
agg_naering.txt	Næringsaggregater, andre
agg_ekstra.txt	Spesialaggregater (inneholder både variabel og klassifikasjon)
prod_2sifret.txt	Produktaggregater, 2-sifret
prod_3sifret.txt	Produktaggregater, 3-sifret
prod_4sifret.txt	Produktaggregater, 4-sifret
prod_andre.txt	Produktaggregater, andre

4.3.4 Vedlikehold/endring av metadata

Ønsker man å endre beskrivelser gjøres dette ved å editere de flate tekstfilene. Neste gang meta-rutiner kjøres vil evt. endringer komme med. Hvilken som helst teksteditor kan benyttes, det eneste man må huske er at det ikke må finnes blanke linjer på slutten av en tekstfil (da vil man få en feilmelding i FAME). Dersom det skal gjøres endringer i teksten på detaljerte serier må oppdateringen skje i ORACLE-tabellen.

NB: Dersom detaljerte næringer skal inngå i engelske publikasjoner må disse tekstene legges inn spesielt på de aktuelle næringer!

5. Tabeller

Alle tabellprosedyrer og inputfiler ligger på området \$NASJREGN/tidsserier/tabell/.

Det er laget to forskjellige tabellrutiner, analysetabeller for internt bruk og publiseringstabeller for presentasjon eksternt. Rutiner for analysetabeller ligger på filene analyse.pro og analyseform.pro, med dokumentasjon på analyseform.doc.

Rutiner for publiseringstabeller ligger på filene knrtabeller.pro, nrtabeller.pro, tabeller.pro og tabellform.pro. Siden tabellprosedyrer og inputfiler som omtales her ble laget, er det utviklet nye tabellrutiner av Erik Søberg (tidligere ansatt ved seksjon 203) og Nils Amdal (seksjon 210). Da det gamle systemet fortsatt er i bruk, gir vi likevel en beskrivelse av rutineene her. For en mer detaljert beskrivelse vises det til «Dokumentasjon av tabellprogram for publisering av NR- og KNR-tabeller» (Alnæs Cecilie 1995). Det er forøvrig også utarbeidet en egen dokumentasjon av det nye systemet (Amdal 1998, Bardardottir 1999). Dette vil imidlertid **ikke** bli omtalt nærmere her.

Vil i dette kapitlet gå kort igjennom selve ideen bak (de gamle) tabellrutinene og beskrive parametre til de forskjellige rutiner. Vil til slutt gi retningslinjer for hvordan man setter opp nye tabeller, samt gi en oversikt over de tabeller som allerede er definerte.

5.1 Hovedprinsipp

Grunnideen for tabellrutiner er at man ut fra en crosslist av variabel, klassifikasjonsledd og verdsett lager en liste over de serier som skal inngå i en tabell. Dersom en bestemt kombinasjon representerer en eksisterende serie/formel i en av de åpne databaser skrives serien ut i tabellen med forspaltetekst hentet ut fra desc- eller desc_en-attributten til hver serie.

En liste med næringskoder, detaljerte eller aggregerte, må defineres for et bestemt tabelloppsett, og samme liste kan brukes for forskjellige variabler. Man definerer linje for linje i tabellen, og listen lagres som en flat tekstfil på UNIX.

For at tekstfilen skal kunne brukes for flere variabler, prissett og justeringsmåter, inneholder den kun 2. ledd fra de serienavn man ønsker i tabellen. Blanke linjer i er oppnådd ved å lage fiktive serier av typen «bnpb.blank.vr». Innrykk er laget ved å markere nivå 0, 1 eller 2.

For at en serie (eller formel) skal kunne skrives ut i tabellrutiner gjelder følgende krav

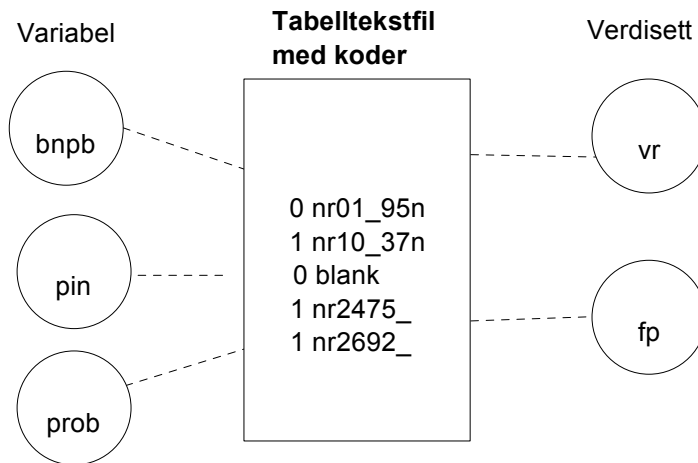
1. Serien må eksistere i en av de åpne databaser, med riktig navnsetting i forhold til hva som er definert i tabelltekstfilen
2. Serien må ha en beskrivelse i attributten desc, og for engelske tabeller også en beskrivelse i desc_en. Finnes serien uten beskrivelse, vil kolonnene inneholde tall, men forspalten bli blank.

5.2 Tabelloppsett

Alle tabelltekstfiler lagres som flate tekstfiler på \$NASJREGN/tidsserier/tabell/. Det er viktig at filnavn er navnsatt tab_*.txt for at de senere skal kunne hentes opp i vindusrutiner.

Det finnes to typer av tabelltekstfiler. Den ene kan brukes for mange forskjellige variabler, mens den andre brukes kun for et bestemt oppsett. Nedenfor følger eksempel på begge varianter.

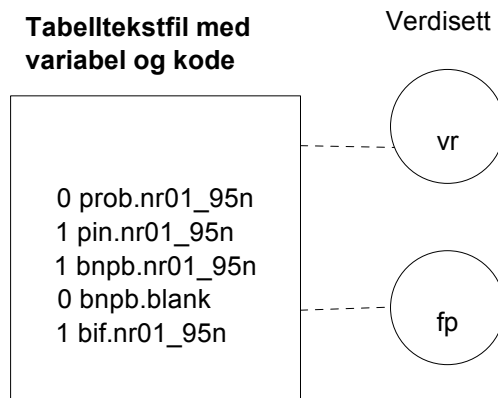
1) Samme oppsett hvor variabeltype kan variere (eks. næringstabeller)



Noen mulige kombinasjoner i en tabell:

bnpb.nr01_95n.vr		pin.nr01_95n.fp		prob.nr01_95n.vr
bnpb.nr10_37n.vr	eller	pin.nr10_37n.fp	eller	prob.nr10_37n.vr
bnpb.blank.vr		pin.blank.fp		prob.blank.vr
----		---		---

2) Spesialtabell med flere variabler i en og samme tabell



Mulige kombinasjoner i en tabell:

prob.nr01_95n.vr		prob.nr01_95n.fp
pin.nr01_95n.vr	eller	pin.nr01_95n.fp
bnpb.nr01_95n.vr		bnpb.nr01_95n.fp
bnpb.blank.vr		bnpb.blank.fp
bif.nr01_95n.vr		bif.nr01_95n.fp

Det er laget forskjellige tabelloppsett etter gjeldende publiseringsstandard for NR-REA. Se vedlegg 1 og 2 for en oversikt over ferdige tabelloppsett.

5.3 Analysetabeller

5.3.1 Hovedrutiner

Hovedrutiner for uttak av analysetabeller til NR-REA er utviklet av Erik Søberg (seksjon 203) i samarbeid med seksjon 210. Tabellrutinen kan brukes for alle databaser hvor navnestruktur på tidsserier er lik NR-REA. Rutinene ligger på område: *\$NASJREGN/tidsserier/tabell/analyse/*.

Analysetabeller for NR-REA inneholder kolonner for løpende priser, faste priser (t-1), volumendringer, verdiendringer og prisendringer. Følgende formler er brukt (vr er løpende priser, fp er faste priser t-1):

- **Volumendringer** = $(fp[t]/vr[t-1] - 1) * 100$
- **Verdiendringer** = $((vr[t]-vr[t-1]) / vr[t-1]) * 100$
- **Prisendringer** = $(vr[t]/fp[t] - 1) * 100$.

Det er maks 3 år på hver tabell. Alle tabeller blir lagret som postscript-filer på brukerens hjemmekatalog på UNIX, i landskapsformat. Det forutsettes at serier som finnes i faste priser også finnes i løpende, men ikke motsatt.

Det er laget to forskjellige tabellrutiner, avhengig av om det er serier kun for næring eller produkt og næring. Rutinen \$nr_analyse_tab brukes for serier med 3 ledd, og \$nr_analyse_tab2 til serier med 4 ledd. Følgende argumenter gis til analyserutiner:

Tabell 5-1 Argumenter til \$nr_analyse_tab

Argument	Type	Eksempel
variabel	string	«bnpb» evt. «»
tekstfil til 2. ledd i serienavn	string	«tab naer»
siste år (av 3 år)	date(annual)	1993
tittel	string	«Bruttoprodukt etter næring»
ledd av beskrivelse	numeric	2

Tabell 5-2 Argumenter til \$nr_analyse_tab2

Argument	Type	Eksempel
variabel	string	«bif» evt. «»
tekstfil til 2. ledd i serienavn	string	«tab_5»
tekstfil til 3. ledd i serienavn	string	«tab_art3»
siste år (av 3 år)	date(annual)	1993
tittel	string	«Bruttoinvestering i fast realkapital»
ledd av serienavn det skal grupperes etter	numeric	2

Navn på rutiner: *\$nr_analyse_tab*, *\$nr_analyse_tab2*

5.3.2 Hjelperutiner

Tabell 5-3 Hjelperutiner til analysetabeller

Rutine	Kaller opp rutine	Beskrivelse
\$spalte		Plukker ut ledd av beskrivelse som skal brukes som forspaltetekst
\$sil_serier		Tar som input en case-serie med alle mulige kombinasjoner av variabel, tekstfiler og verdisett. Returnerer case-serie med kun eksisterende serier. Setter boolsk variabel «tomtsoek» lik True dersom ingen serier finnes.
\$print_analyse_tab	UNIX-utskrift	Tar som argument filnavn. Sender filen til valgt printer
\$nr_arb_tab_setting		Tar som argument siste år, tabelltittel og filnavn. Bestemmer tabelloppsettet.
\$nr_select_col12		Tar argument for kolonnebredde og hvilket ledd av beskrivelse som skal være tekst i tabellen. Setter forspaltetekst lik klassifikasjonsledd + ledd av beskrivelse.
\$nr_select_data		Tar som argument kolonnebredde og siste år. Beregner innhold i kolonner med løpende priser, faste priser, verdiendringer, volumendringer og prisendringer.
\$nr_head_foot		Setter titler og fotnoter.
\$nr_print3dim		Tar som input navneliste med eksisterende serier, og ledd av beskrivelse som forspalte. Printer ut linje for linje
\$nr_print4dim		Tar som input navneliste med eksisterende serier, kolonnebredde og ledd av beskrivelse som forspalte. Printer ut linje for linje.
\$ps_landskap_a4		Tar som input filnavn. Lager en postscriptfil i landskap.

5.3.3 Tabellpakker

Man kan definere en pakke med tabeller ved å benytte input-filer med faste parametre. Filene *analyse.inp* og *analyse_t3.inp* lager detaljerte tabeller for de fleste variabler i NR-REA. Disse input-filene ligger på område *\$NASJREGN/tidsserier/tabell/analyse/*. Databaser åpnes i input-filene.

Navn på ferdigdefinerte tabellpakker: *analyse.inp*, *analyse_t3.inp*

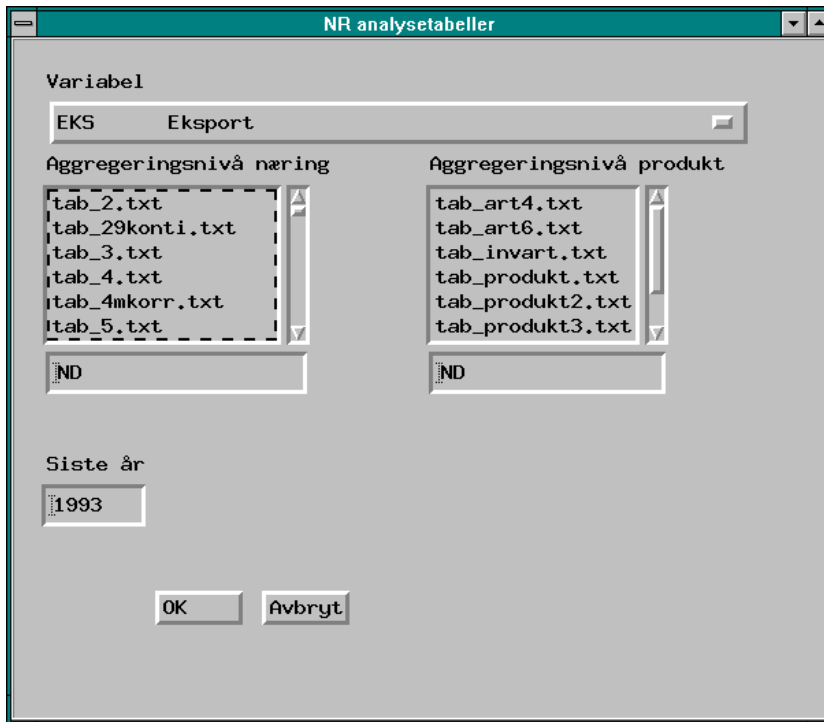
5.3.4 Eksempel på bruk

Nedenfor følger eksempler på kall til tabellrutiner, og eksempel på en analysetabell finnes som vedlegg 6.

1. Produksjon i basispris aggregert etter næring
\$nr_analyse_tab "PROP", "tab_naer",aar, "Produksjon, produsentverdi", 2
2. Sysselsatte personer totalt etter industri på KNR-nivå
\$nr_analyse_tab «PES», «tab_indknr», aar, "SYSSELSATTE PERSONER i INDUSTRI", 2
3. Bruttoinvesteringer etter detaljert næring og 4-sifret art, gruppert etter næring
\$nr_analyse_tab2 «BIF», «tab_5», «tab_art4», aar, «Bruttoinvesteringer i fast realkapital», 3

5.3.5 Vindusform

Det er laget en enkel vindusform for uttak av en analysetabell. Vindusgrensesnittet startes ved å skrive '\$analyseform'. Databaser åpnes i selve prosedyren. Man velger ut variabel, aggregeringsnivå og siste år for uttak av en enkelt tabell. Dersom tabellen skal være for næring og produkt/art kan det ta noe tid. For å unngå tomme søk bør man sjekke hvilke aggregater for produkt og næring som faktisk finnes for bruttoinvesteringer, eksport og import. Det grupperes automatisk etter næring. Tabellen lagres på brukerens hjemmekatalog. Figur nedenfor viser skjermbilde for å lage en analysetabell.



Figur 5-1 Skjermbilde for å lage en analysetabell

- *Variabel*

Listen over variabler er hentet fra tekstfilene variabel.txt, syss.txt og t3.txt på området \$NASJREGN/tidsserier/meta/. Tabelltittel vil automatisk settes lik beskrivelsen til den valgte variabel.

- *Aggregeringsnivå klassifikasjonsledd*

Listen over ferdige tabelloppsett for næringer hentes fra alle tekstfiler av typen tab_*.txt på området \$NASJREGN/tidsserier/tabell/. Kan være ND dersom tabellen gjelder kun produkt/art.

- *Aggregeringsnivå produkt/art*

Listen over ferdige tabelloppsett for produkter/arter hentes fra alle tekstfiler av typen tab_art*.txt og tab_produkt*.txt på området \$NASJREGN/tidsserier/tabell/. Kan være ND dersom tabellen gjelder sum næring.

- *Siste år*

Sluttdato for en tabell med 3 år.

5.4 Publiseringstabeller

Applikasjon for uttak av publiseringstabeller til KNR og NR utviklet av Cecilie Alnæs (seksjon 710) i samarbeid med seksjon 210.

Tabellrutinen kan brukes for alle databaser som har samme navnestruktur som NR-REA eller KNR. Hvilke databaser som skal brukes bestemmes i inputfilen. For å endre hvilke databaser som brukes i vindusrutinen må man endre i filen tabellform.pro og deretter kompilere.

Utseendet på tabeller er veldig enkle; år og kvartal i kolonner og rader med serier. De fleste rutiner for KNR og NR er felles. De forskjellige tabelltyper er:

- Strukturtabeller
- Vekstratetabeller
- Prisendringstabeller
- Pristabeller
- Volumtabeller

Man kan ta ut Postscript- eller ASCII-tabeller, og det eksisterer også et valg for om man ønsker kontotype/kode som en del av forspalteteksten. Tabellen vil lagres der hvor man startet FAME fra, med filnavn ut fra variabel, tabelltekstfil og tabelltype. For en grundig dokumentasjon av hoved- og hjelperutiner vises til C. Alnæs «Dokumentasjon av tabellprogram for publisering av NR- og KNR-tabeller».

Navn på hovedrutiner: *\$nrtabellascii*, *\$nrtabellps*, *\$nrtabellform*

5.4.1 Argumenter til tabellrutinen

Utseende og format på hver enkel tabell bestemmes vha. input til hovedrutinen \$Tabell. Følgende argumenter må gis:

Tabell 5-4 Argumenter til \$tabell

Argument	Type	Eksempel
overskrift	string	«Nasjonalregnskap 1988 - 1993»
tabelltittel	string	«Bruttonasjonalprodukt»
variabel	string	«BNPB» evt. blank «»
tekstfil med koder	string	«tab naermkorr»
prissett	string	«FP»
ledd av seriebeskrivelse	numeric	2
startdato, kvartal	string	«» (kun KNR)
sluttdato, kvartal	string	«» (kun KNR)
startdato, år	string	«1988»
sluttdato, år	string	«1993»
tabelltype	string	«level», «ppe», «prisindeks», «prosentindeks» + flere (se dok.)
referanseår	date	1988, evt. blank «»
footnote	string	Fire footnote-argumenter, f.eks. «Foreløpig 1993», evt. blanke «»
forspalte med eller uten næringskoder	string	«arb» er med næringskoder, «publ» er uten (koder helst for arbeidstabeller)
ASCII eller Postscript-tabell	string	«rpt» for ASCII, «ps» for Postscript

*strLang = global variabel for hvilket språk i forspalteteksten. Settes i styringsprogram	string	«Norsk» eller «Engelsk»
---	--------	-------------------------

5.4.2 Tabellpakker

Det finnes flere ferdigdefinerte tabellpakker på NR- og KNR-nivå, samt tabellaggregater på NR- og KNR-nivå. En oversikt over forskjellige tabell-inputfiler er gitt nedenfor.

Tabell 5-5. Ferdigdefinerte tabellpakker, NR- og KNR-nivå

Tabellpakke	Beskrivelse
nrpub.inp	Div. tabeller NR-nivå
nrpub knr.inp	Div. tabeller KNR-nivå
nrpub eng.inp	Aggregert NR-nivå, engelsk forspaltetekst
nrpub oa.inp	Aggregert NR-nivå til Økonomiske Analyser
nrpub knr oa.inp	Aggregert KNR-nivå til Økonomiske Analyser
nrpub knr oa eng.inp	Aggregert KNR-nivå til Economic Survey
nrpub knr us.inp	Aggregert KNR-nivå til Statistisk Ukehefte
nrpub nos97 no.inp	Tabeller til NOS-publikasjonen, norsk forspaltetekst
nrpub nos97 en.inp	Tabeller til NOS-publikasjonen, engelsk forspaltetekst
nrpub_sys.inp	Lønn og sysselsetting, disaggregert NR-, disaggregert KNR- og aggregert KNR-nivå

5.4.3 Eksempel på bruk

Nedenfor følger eksempler på kall til tabellrutinen, og eksempel på en publiseringstabell finnes som vedlegg 7.

1. Spesialtabell med hovedtall for konsum i faste priser for årene 1989-1993. Tabellen skal være i ASCII-format, uten næringskoder.

\$Tabell «NASJONALREGNSKAP 1988 - 1993», "Hovedtall for konsum", "", "tab_koialt", «FP», 2, «», «», »1989», »1993», "level", "", «gml. definisjon for privat konsum ...», «publ», «rpt»

2. Detaljerte serier med bruttoinvesteringer i løpende priser, prosentvis endring fra året før, for årene 1988-1993. Tabellen skal være i postscriptformat, med næringskoder.

\$Tabell «NASJONALREGNSKAP», «Bruttoinvesteringer i fast realkapital», "BIF", "tab_5", "VR", 2, «», «», »1988», »1993», "ppe", "", «Tall for 1988-1992 ble sist publisert i ...», «arb», «ps»

5.4.4 Vindusform

Det finnes et vindusgrensesnitt for uttak av en enkel publiseringstabell. Grensesnittet startes fra FAME med kommandoen '\$nrtablellform'. Tabellen vil lagres der hvor man startet FAME fra, med filnavn ut fra variabel, tabelltekstfil og tabelltype.

Hvilke databaser som skal åpnes bestemmes ut fra om man velger «KNR-nivå» eller «NR-nivå». Velger man «KNR-nivå» åpnes knrgrlag.db og knrgrlag_agg.db, og man kan velge tabelltekstfiler med KNR-aggregater. Velger man «NR-nivå» åpnes nr01.db, nr02.db og nrsyss.db, og man velger tabelltekstfiler med NR-aggregater. Figur nedenfor viser skjermbildet for \$tablellform.

Figur 5-2 Skjerm bilde for å lage en publiseringstabell

For nærmere veiledning i bruk av vindussystemet vises til C. Alnæs «Dokumentasjon av tabellprogram for publisering av NR- og KNR-tabeller».

5.4.5 Konvertering til SGML

SGML⁴ er en internasjonal standard for koding av dokumenter. Byrået var med i et pilotprosjekt i samarbeid med Falch Hurtigtrykk for å prøve ny tabellproduksjon vha. SGML, og dermed sikre en enhetlig profil. Hovedmålet for den nye tabellprosessen var «Å få til en effektiv vei fra ulike tabellverktøy til ulike publiseringverktøy, som samtidig ivaretar de krav til formatering og layout byrået har».

Opprinnelig skulle tabeller i ASCII-format konverteres til SGML vha. rutinen PCTab. Dette fungerte imidlertid ikke helt etter hensikten, og rutinen ble derfor skrinlagt. I dag benyttes et annet opplegg. Tabellene lagres først i Excel-format, og kodes deretter til SGML-format. Tabellene til både Dagens Statistikk og Statistisk Årbok lages på denne måten.

⁴ Structured Generalized Markup Language

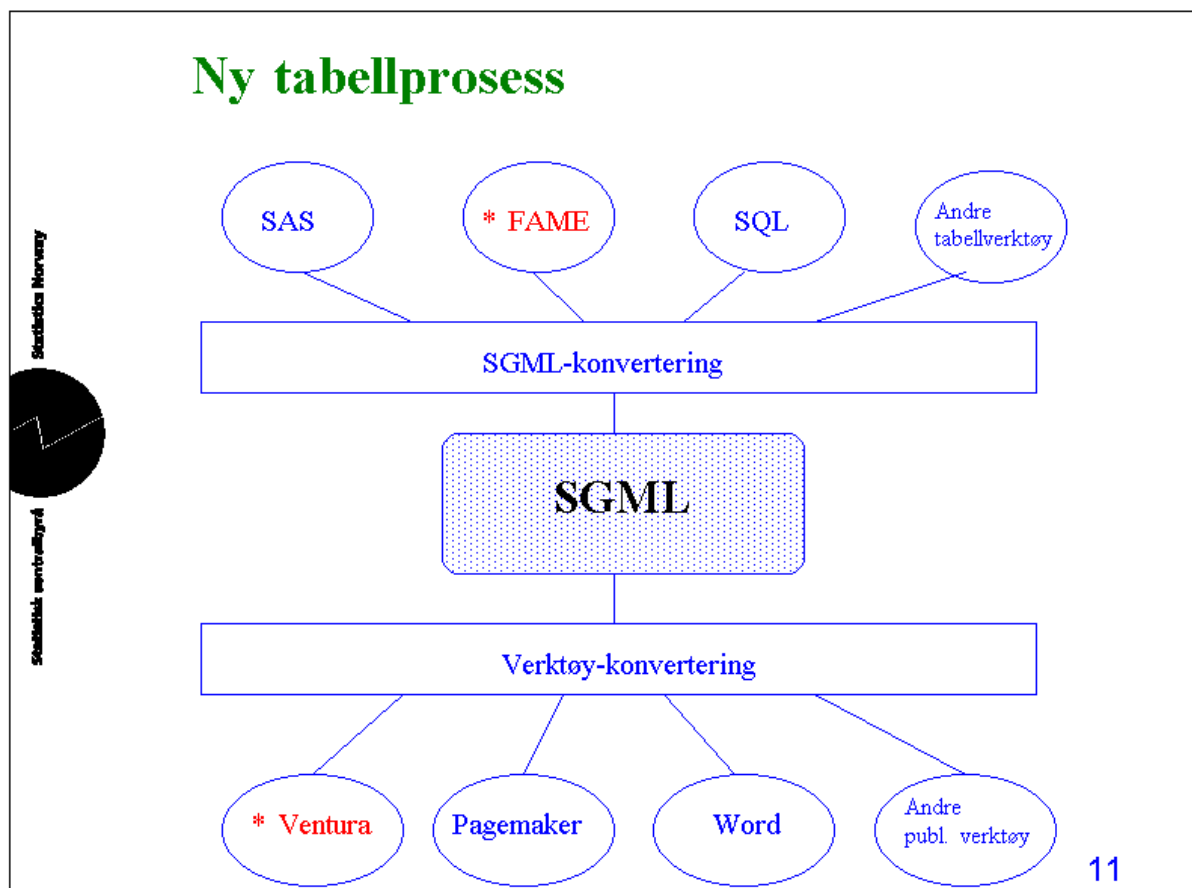
NR-tabeller til Statistisk Årbok i ASCII-format tilrettelegges for import til Excel. Tabellene kjøres først ut fra FAME i ASCII-format, og tabellinnmaten limes deretter inn i ferdige Excel-ark, som inneholder norske og engelske tabellforspalter og tabelloverskriftene. Selve publiseringen går veldig raskt og effektivt.

Vi vil ikke gå nærmere inn på hvordan disse rutine fungerer, kun gi en forklaring på hva SGML er og hva man forsøker å oppnå med det.

Hva er SGML?

- Internasjonal standard for koding av dokumenter
- Beskriver struktur, ikke utseende
- Gjør dokumenter uavhengige av bestemte applikasjoner, systemer, tegnsett osv.
- Brukes primært til tekst, men kan også håndtere referanser til andre datatyper som grafikk, lyd, etc. Gjenbruk når dokumenter skal utgis på flere medier (eks. papir, CD-ROM, WWW, blindeskrift) Dokumenter kan grupperes i klasser (dokumenttyper, eks. tabelltyper) ut fra fellestrekk i strukturen

Figur nedenfor viser sammenhengen mellom ulike tabellverktøy, SGML og sluttredigeringsverktøy.



Figur 5-3 Structured Generalized Markup Language

6. Referanser

- Alnæs, Cecilie (1995): Dokumentasjon av tabellprogram for publisering av NR-tabeller. Upublisert notat.
- Alnæs, Cecilie: Publisering av tabeller på internett. Upublisert notat.
- Amdal, Nils (1998): Standard tabellapplikasjon til uttak av FAME-data. Upublisert notat.
- Amdal, Nils (1998): FAME og nasjonalregnskapet. Dataflyt og databaser. Upublisert notat.
- Bardardottir, Asta (1999): Om bruk av Standard Tabellapplikasjon i Fame i seksjon 210. Veiledning for å skrive ut og forandre tabeller i Fame. Retningslinjer som gjelder for ulike publikasjoner. Upublisert notat
- Hoholm, Knut Vidar og Torvbråten, Hans Krstian (29. oktober 1994). Edb-utvikling. Til: Publiseringssgrupper, Informasjon og publisering og Edb-utvalget. Upublisert notat.
- Olsen, Hilde og Skjæveland, Arent (1985). Teknisk dokumentasjon av beregningsopplegget for det kvartalsvise nasjonalregnskapet. Interne Notater 85/15. Statistisk sentralbyrå.
- Ouren, Jørgen (1995): Rutiner for avstemming av kvartalstall mot årstall. Upublisert notat.
- Ouren, Jørgen (1996): Litt om avstemming av kvartalstall mot årstall . Upublisert notat.
- Pettersen, Eirin (1996): Bruerveiledning for beregning av kvartalsvis nasjonalregnskap. Upublisert notat.
- Pettersen, Eirin (1996): Teknisk dokumentasjon av beregningsopplegg for KNR. Upublisert notat.
- Pettersen, Eirin (1996): Teknisk dokumentasjon av det årlige realregnskapets FAME-databaser. Upublisert notat.
- Sexton, Joseph (1998). Fremskrivning av tidsserier i KNR. Notater 98/88, Statistisk sentralbyrå.
- Skjæveland, Arent (1985): Avstemming av kvartalsvise nasjonalregnskapsdata mot årlige nasjonalregnskap. Interne Notater 85/22, Statistisk sentralbyrå.
- Seksjon 730, Statistisk sentralbyrå (1999): Omlegging til daglig frigivning på web. Rutinebeskrivelse for fagseksjonene 3.6.99
- Statistisk sentralbyrås håndbøker 63: Håndbok i IT-metode, Versjon 2.0
- Søberg, Erik (1995): UNIX Kurs- og brukerhefte. Interne dokumenter 95/10, Statistisk sentralbyrå
- Todsen, Steinar (1999): Kvartalsvis nasjonalregnskap - dokumentasjon av beregningsopplegget Rapport 99/25, Statistisk sentralbyrå.
- Tønjum, Pia E. (1999): Teknisk dokumentasjon av beregningsopplegget for det kvartalsvise nasjonalregnskapet (KNR). Notater 99/51, Statistisk sentralbyrå
- Tønjum, Pia E. (1999): Dataoverføring fra fagseksjonene til det kvartalsvise nasjonalregnskapet. Upublisert notat.

Tønjum, Pia E. (1999): EDB-rutiner for kjeding av årlig og kvartalsvis nasjonalregnskap. Notater 99/19, Statistisk sentralbyrå

Vogt, Yngve (1994): Innføring i FAME. Interne notater 94/18, Statistisk sentralbyrå

7. Vedlegg

1. Ferdige tabelloppsett med NR-aggregering

Alle tabelltekstfiler etter NR-aggregering ligger på \$NASJREGN/tidsserier/tabell/

Tabell 5-5 Tabelltekstfiler med NR-aggregering

Tekstfil	Aggregeringsnivå
tab_5.txt	Detaljert næring
tab_5mkorr.txt	Detaljert næring, inkl. korreksjonspostene 23654, 23659, 23669
tab_5knrind.txt	Detaljert næring, industri på KNR-nivå (lønn- og sysselsetting)
tab_4.txt	4-sifret næring
tab_4mkorr.txt	4-sifret næring, inkl. korreksjonsposter
tab_3.txt	3-sifret næring
tab_2.txt	2- "
tab_produkt2.txt	2-sifrede produkter
tab_produkt3.txt	3-sifrede produkter
tab_produkt4.txt	4-sifrede produkter
tab_produkt6.txt	6-sifrede produkter
tab_art4.txt	4-sifrede arter
tab_art6.txt	6-sifrede arter
tab_6naer.txt	6 næringer
tab_naer.txt	Næringstabell (jfr. juli-publisering)
tab_naermkorr.txt	Næringstabell, inkl. korreksjonsposter
tab_ind.txt	Industri (jfr. juli-publisering)
tab_indknr.txt	Industri på KNR-nivå
tab_kat.txt	Kategori av virksomhet
tab_katmkorr.txt	Kategori av virksomhet, inkl. korreksjonsposter
tab_omr.txt	Hovedområder (jfr. juli-publisering)
tab_naer22.txt	22-næringer
tab_naer24.txt	24-næringer
tab_naer25.txt	25-næringer
tab_naer26.txt	26-næringer
tab_kohformaal.txt	Konsum i husholdninger gruppert etter formål
tab_kohvarig.txt	Konsum i husholdninger gruppert etter varighet
tab_koialt.txt *	Hovedtall for konsum
tab_kostat.txt	Konsum i statsforvaltningen gruppert etter formål
tab_kokom.txt	Konsum i kommune forvaltningen gruppert etter formål
tab_invert.txt	Bruttoinvestering i fast realkapital etter art
tab_hoved.txt *	BNP etter anvendelse

* = spesialtabell hvor flere variabler inngår

2. Ferdige tabelloppsett med KNR-aggregering

Tabelltekstfiler etter KNR-aggregering ligger på området \$NASJREGN/knr/tabell/

Tabell 5-6 Tabelltekstfiler med KNR-aggregering

Tabellfil	Beskrivelse
tab_ps.txt	Produksjonsnæringer, detaljert (23xx-26xx)
tab_aggps.txt	Produksjonsnæringer, aggregert
tab_ps29.txt	Produksjonsnæringer inkl. 29-konti, detaljert (23xx-29xx)
tab_aggps29.txt	Produksjonsnæringer inkl. 29-konti, aggregert
tab_aggpin.txt	Ekstra tabell for aggregert produktinnsats
tab_js.txt	Investeringsnæringer, detaljert (83xx-86xx)
tab_aggjs.txt	Investeringsnæringer, aggregert
tab_jr.txt	Investeringsarter, detaljert (08xx)
tab_aggjsjr.txt	Investeringsart- og næringer, aggregert (08xx, 83xx-86xx)
tab_cp.txt	Konsumgrupper i husholdninger, detaljert (61xx)
tab_aggcp.txt	Konsumgrupper i husholdninger, aggregert
tab_aggkons.txt *	Konsum, hovedtall
tab_aggeks.txt	Eksport aggregert
tab_aggimp.txt	Import aggregert
tab_g.txt	Konsumgrupper i offentlig forvaltning, detaljert (64xx-65xx)
tab_produkt.txt	Produkter, detaljert
tab_makro.txt *	Makroøkonomiske hovedstørrelser
tab_anvendelse.txt *	BNP etter anvendelse
tab_jord.txt	Div. størrelser for jordbruksproduktene (011 og 012)
tab_fiske.txt	Div. størrelser for produkter fra fiske og fangst (051 og 052)
tab_driftsregn.txt *	Driftsregnskapet overfor utlandet

* = spesialtabell hvor flere variabler inngår

3. Brukerveiledning for uttak av analysetabeller

- Legg følgende inputfil på katalogen du ønsker å starte fra (eks. på ~hal/ekstern/analyseform.inp), evt. skriv det samme i dialogvinduet fra uansett område på UNIX:

analyseform.inp

```
analyseform.inp load
«$NASJREGN/tidsserier/tabell/analyseform»
analyseform
```

- Start FAME ved å skrive 'FAME'
- For uttak av en enkel analysetabell, skriv 'inp analyseform' i dialogvinduet, evt. skriv de kommandoer som filen over inneholder. Klikk på ønsket variabel, aggregeringsnivå og siste år i tabell med 3 år.
- For uttak av en pakke med analysetabeller, skriv «inp analyse», evt. 'inp analyse_t3' for tabellpakke til T3. Se avsnitt 6.3.3 for oversikt over eksisterende tabellpakker.
- Tabeller lagres som enkeltfiler på din hjemmekatalog med filnavn ut fra variabel, aggregeringsnivå og siste år i tabellen.
- Gå til din hjemmekatalog ved å skrive 'cd'
- List alle tabeller med UNIX-kommandoen 'ls *.ps'
- Kan se på en tabell ved å bruke ghostview i UNIX. Snu tabellen til landskap ved å trykke 'orientation- landscape'
- Skriv ut tabeller i 'porsjoner' med UNIX-kommandoen cat, eks.
cat prob*.ps | ps5b16
cat bnpb*.ps | ps5b16

4. Brukerveiledning for uttak av publiseringstabeller

- Legg følgende inputfil på katalogen du ønsker å starte fra (eks. på `~hal/ekstern/pubtabell.inp`), evt. skriv det samme i dialogvinduet fra uansett område på UNIX:

pubtabell.inp

```
close all
load
«$NASJREGN/tidsserier/tabell/tab
ellform»
$nrtableform
```

- Start FAME ved å skrive ‘ FAME ‘
- For uttak av en enkel tabell, skriv ‘ inp pubtabell ’ i dialogvinduet, evt. skriv de kommandoer som filen over inneholder. I vindusformen velger man om det gjelder NR- eller KNR-nivå, variabel, aggregeringsnivå, tabelltype, transformasjon, år, format, næringskoder eller ikke og norsk eller engelsk forspaltetekst. For nærmere beskrivelse av de forskjellige valg i tabellformen vises til «Dokumentasjon av tabellprogram for publisering av NR- og KNR-tabeller» av Cecilie Alnæs (710).
- For å lage en bunke med publiseringstabeller, skriv ‘ inp «\$NASJREGN/tidsserier/tabell/nrpub» ‘ i dialogvinduet. Evt. ‘ inp «\$NASJREGN/tidsserier/tabell/nrpub_eng» ’ for tabellpakke på KNR-nivå. Se avsnitt 6.4.2 for oversikt over tabellpakker.
- Tabeller lagres som enkeltfiler på den katalogen du startet FAME fra. Filnavnet lages ut fra variabel, tabelltekstfil, transformasjon og format. Eks. `prob_naer_vrp.rpt`, `koh_5_vlp.ps`.
- Når ønskede tabeller er laget, avslutt FAME ved å skrive ‘ exit ‘. List alle tabeller med UNIX-kommandoen ‘ `ls *.ps` ’ for Postscript-tabeller, eller ‘ `ls *.rpt` ’ for ASCII-tabeller.
- Kan se på en tabell ved å bruke `ghostview` i UNIX. Skriv ‘ `ghostview + navn på filen` ’. Kan skrive ut tabellen herfra ved å velge ‘print’ under fil-menyen.
- Skriv ut tabeller i ‘porsjoner’ med UNIX-kommandoen `cat`, eks.
`cat prob*.ps | ps5b16`
`cat bnpb*.ps | ps5b16`

5. Hvordan definere nye aggregater - et eksempel

- Utgangspunkt: Ønsker å definere følgende nye næringsaggregater:

Navn (maks 9 tegn)	Innhold (8x for invest.)	Norsk tekst (maks 45 tegn)	Eng. tekst (maks 45 tegn)
051N	Sum 22051, 23051	Fiske og fangst	Fishing
151 8N	Sum 23151-23158	Næringsmiddelindustri	Food products
26853 4	sum 26853, 26854	Sosial- og omsorgstjenester	Social work
25853 4	sum 25853, 25854	Sosial- og omsorgstjenester	Social work
24851 2	sum 24851, 24852	Helsetjenester	Health
45m	sum 2245_, 2345_	Bygge- og anleggsvirksomhet	Construction

Fremgangsmåte:

1. Bestemmer at aggregeringen foreløpig skal gjelde for variablene prob, prop, pin, bnpb, bnpp, bif, loko i NR-REA i løpende og faste priser.
2. Lager en FAME-inputfil i Emacs på området \$NASJREGN/tidsserier/meta/ hvor man kaller opp aggregeringsrutinen \$lag_formel for hvert enkelt aggregat. Husk at underliggende serier som defineres i aggregatet må finnes i nr01.db!

I eksemplet har jeg laget en fil *agg_tillegg.inp* (finnes på \$NASJREGN/tidsserier/meta).

agg_tillegg.inp

```
-- -- Initierer og åpner databaser --

glue dot
load «$NASJREGN/tidsserier/meta/lag_formel.pc»
close all
open <access shared> «$NASJREGN/tidsserier/NR/DB/nr02.db» as db
store nr02
----

$lag_formel {bnpb, bnpb, prob, prop, pin, loko}, «nr051n», {nr22051, nr23051}, {vr, fp}
$lag_formel {bif}, «nr051n», {nr82051, nr83051}, {vr, fp}

$lag_formel {bnpb, bnpb, prob, prop, pin, loko}, «nr45m», {nr2245_, nr2345_}, {vr, fp}
$lag_formel {bif}, «nr45m», {nr8245_, nr8345_}, {vr, fp}

etc..
```

3. Starter FAME fra \$NASJREGN/tidsserier/meta/. Oppretter formler ved å kjøre *agg_tillegg.inp*. Skriv *<inp_agg_tillegg>* i dialogvinduet. Liste med alle formler som er laget skrives ut i outputvinduet.
4. Legger inn norsk og engelsk beskrivelse i en tekstfil på området \$NASJREGN/tidsserier/meta/.

I eksemplet har jeg laget en ny fil, *agg_tillegg.txt*.

Teksten kan skrives inn i hvilken som helst editor, men må overføres til UNIX dersom man velger en PC-editor. Svakheten med Emacs er at den ikke håndterer kolonner, noe som er praktisk i dette tilfellet for å sikre at teksten ligger i de riktige posisjoner. Viktig at norsk og engelsk tekst ikke er over 45 karakterer, at det ikke er brukt tabulator og at det ikke ligger noen blanke linjer på slutten av filen.

agg_tillegg.txt

Navn (posisjon 1-9)	Norsk tekst (posisjon 10-55)	Eng. tekst (posisjon 56-102)
051N	Fiske og fangst	Fishing
151_8N	Næringsmiddelindustri	Food products
26853_4	Sosial- og omsorgstjenester	Social work
25853_4	Sosial- og omsorgstjenester	Social work
24851_2	Helsetjenester	Health
45m	Bygge- og anleggsvirksomhet	Construction

5. Dersom tekstfilen med beskrivelser er ny må den inkluderes i styringsrutinene *meta_nr02.inp*. Må også inkluderes i *meta_nrsyss.inp* dersom aggregeringen gjelder sysselsetting. Begge inputfiler ligger på \$NASJREGN/tidsserier/meta/.

I disse inputfilene ramser man opp hvilke beskrivelsesfiler som skal brukes. Veldig teknisk, men det er måten det gjøres på i FAME (&& betyr feks. bare at kommandoen går over flere linjer). Eksempel:

Utdrag av FAME-kode fra meta_nr02.inp

```
---
---
type "Starter med oppdatering av metadata i nr02.db " + now --

-/dim1 = system ("cat variabel.txt",case)

-/dim2 = system("cat agg_2sifret.txt agg_3sifret.txt agg_4sifret.txt agg_mellom.txt agg_naering.txt
agg_tillegg.txt", case)

-/dim4 = system("cat verdi.txt",case)

-- fortsetter videre og kaller opp metarutinen $meta_ledd3 -
--
```

6. Starter FAME fra \$NASJREGN/tidsserier/meta/, og skriver «*inp meta_nr02*» i dialogvinduet for å legge inn beskrivelser på alle serier og formler i nr02.db, også nye tekster i **agg_tillegg.txt**.

Meta-rutinen vil ta noe tid dersom man oppdaterer alle eksisterende aggregater. Resultatet vil være oppdaterte norske og engelske beskrivelser på alle objekter i nr02.db.

7. Ønsker å lage en ny tabell med kun de nye aggregatene og en total for næringer og offentlig forvaltningsvirksomhet.

Lager en tekstfil på området \$NASJREGN/tidsserier/tabell/ med navn av typen «*tab_*.txt*», i dette eksempel «*tab_eksempel.txt*». Definerer linje for linje 2. ledd i navnet på de serier som skal inngå i tabellen, også hvor man ønsker blanke linjer og innrykk. Det man må huske er at prefikset «NR» nå en del av serienavnet sammen med næringskoden, og må derfor skrives i tabelltekstfilen.

tab_eksempel.txt

0 nr01_95n
0 nr01_95ofn
1 nr051n2
1 nr151_8n
1 nr26853_4
2 nr25853_4
0 nr24851_2
0 nr45m

8. Tabelltekstfilen kan nå brukes både for analysetabeller og publiseringstabeller, i kombinasjon med de variabler aggregatene er laget for.

Eksempel på kall på analysetabeller for produksjon i basisverdi:

\$nr_analyse_tab «PROB», «tab_eksempel», 93, «Produksjon i basisverdi», 2

Eksempel på kall på publiseringstabeller for produktinnsats i faste priser, nivå, for årene 1988 til 1993, med næringskoder, i ASCII-format.:

*\$Tabell «NR-REA», «Poduktinnsats», "PIN", "tab_eksempel", "FP", 2
,»,»,»,»,»1988»,»1993», "level", "", «Dette er en test på nye aggregater», «arb», «rpt»*

De sist utgitte publikasjonene i serien Notater

- 99/31 T.P. Bø: Klassifisering av registrerte arbeidsledige og personer på tiltak i arbeidskraftundersøkelsen (AKU). 45s. merker knyttet til familie- og husholdningsstatistikken. 37s.
- 99/32 E. Birkeland (red.): Forskjeller i levekår: Hefte 1: Inntekt. 176s. 99/45 E. Rønning: Omnibusundersøkelsene 1998: Dokumentasjonsrapport. 123s.
- 99/33 E. Birkeland (red.): Forskjeller i levekår: Hefte 2: Levekår og helse. 107s. 99/46 C. Torp: Situasjonsuttak fra Bedrifts- og foretaksregisteret. 33s.
- 99/34 E. Birkeland (red.): Forskjeller i levekår: Hefte 3: Bruk av velferdsordninger. 126s. 99/47 T.N. Evensen: Utlendingers konsum i Norge: En vurdering av eksporttallene for reisetraffikk i nasjonalregnskapet på bakgrunn av statistikk som belyser forbruket til utenlandske turister i Norge. 28s.
- 99/35 E. Birkeland (red.): Forskjeller i levekår: Hefte 4: Regionale forskjeller. 118s. 99/48 H. Hartvedt (red.): Definisjonskatalog for grunnskoleopplæring for barn og voksne. 14s.
- 99/36 M. Stålnacke, J-A. Sigstad Lie og L. Solheim: En analyse av SSBs generelle utvalgsplan fra 1995 basert på næringsvise sysselsettingstall. 83s. 1999/49 K. Bjønnes og J. Johansen: FD - Trygd: Dokumentasjonsrapport. Attføringspenger, 1992-1997. 126s.
- 99/37 B.O. Lagerstrøm: Trivsels- og arbeidsmiljøundersøkelse blant intervjuere i Statistisk sentralbyrå. 155s. 1999/50 E. Høydahl: FoB2000: Rapport fra seminar 4. juni 1999 om kommuneprodukter fra Folke- og boligtellingsen 2000. 32s.
- 99/38 K.J. Einarsen: Evalueringsrapport for pilotforsøket for FylkesKOSTRA-utdanning. 55s. 1999/51 P.E. Tønjum: Teknisk dokumentasjon av beregningsopplegget for kvartalsvis nasjonalregnskap (KNR). 91s.
- 99/39 L. Rogstad: FoB2000: Adressesamsvar mellom folkeregister og adresseregister i GAB: – rapport fra Lysebu-seminar 8. og 9. desember 1998, – tiltaksplan for bedre adressesamsvar. 39s. 1999/52 F. Gundersen: Statistikk over etterforskede lovbrudd: Dokumentasjon. 46s.
- 99/40 D. Roll-Hansen: Samordnet levekårsundersøkelse 1998 – tverrsnittundersøkelsen: Dokumentasjonsrapport. 102s. 1999/53 N. Arnesen og Ø. Skullerud: Statistikk over emballasjeavfall: Beregningsresultater for 1997. 36s.
- 99/41 R. Johannessen: Kommunale gebyrer knyttet til bolig. Januar 1999. 30s. 1999/54 Ø. Kleven: Bruk av kreftundersøkelsen PSA blant menn i alderen 50 til 65 år. 19s.
- 99/42 M. Stålnacke, A.G. Hustoft og L. Solheim: Vurdering av kvalitet i statistikk: En oversettelse av notater fra Eurostat om kvalitetsrapportering. 77s. 1999/55 P.M. Holt og L. Wiker: Inntekts- og formuesundersøkelsen for aksjeselskaper 1996: Dokumentasjon. 30s.
- 99/43 E. Engelen, K. Myklebust, J.A. Paulsen og L. Rogstad: FoB2000: Stedfesting av bedrifter – forprosjekt. 40s. 1999/56 B.O. Lagerstrøm: Små og mellomstore bedrifters vurdering av kostnader ved lover og regelverk: Hovedresultater. 129s.
- 99/44 I. Hauge, C. Hendriks, Ø. Hokstad og A.G. Hustoft: Standard for begreper og kjennemerker knyttet til familie- og husholdningsstatistikken. 37s. 1999/57 L.H. Thingstad: Regnskapsstatistikk for varehandel 1996: Dokumentasjon av produksjonsrutiner. 36s.